



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MONITORING SERVERŮ

SERVER MONITORING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. IVAN STRAKA

VEDOUcí PRÁCE

SUPERVISOR

Ing. JAROSLAV DYTRYCH, Ph.D.

BRNO 2019

Zadání diplomové práce



21475

Student: **Straka Ivan, Bc.**
Program: Informační technologie Obor: Informační systémy
Název: **Monitoring serverů**
Server Monitoring
Kategorie: Web
Zadání:

1. Seznamte se s jazyky a prostředky pro tvorbu webových informačních systémů a s administrací operačního systému Linux.
2. Prostudujte informační systém Výzkumné skupiny znalostních technologií (KNOTIS) a jeho moduly pro monitoring serverů.
3. Navrhněte nové moduly pro monitoring serverů do systému KNOTIS určené zejména pro monitoring diskových úložišť a logů. Zaměřte se při tom na minimalizaci režii na monitorovaných serverech a přehlednou prezentaci uživateli umožňující včasné odhalení blížícího se selhání jednotlivých disků.
4. Implementujte navržené řešení.
5. Zhodnoťte dosažené výsledky a vytvořte stručný plakát prezentující výsledky práce.

Literatura:

- Dle doporučení vedoucího

Podrobné závazné pokyny pro vypracování práce viz <http://www.fit.vutbr.cz/info/szz/>

Vedoucí práce: **Dytrych Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2018

Datum odevzdání: 22. května 2019

Datum schválení: 5. listopadu 2018

Abstrakt

Práce řeší monitoring serverov so zameraním na úložiská a systémové záznamy formou modulov do informačného systému KNOTIS. Systém upozorňuje zodpovedné osoby na nezvyčajné aktivity na serveri, či zhoršujúci sa stav úložisk, ktorý môže viesť k zlyhaniu a následnej strate dát. Práca popisuje automatický zber dát, ich transformáciu a užívateľské rozhranie, ktoré je vytvorené vo webovom prostredí a umožňuje nastavenie rôznych parametrov sledovania serverov. Informácie o stave diskov poskytuje technológia SMART a práca berie ohľad na využitie diskových polí, či technológiu LVM. Riešenie monitoruje aj najdôležitejšie súbory systémových záznamov – auth.log, syslog a kern.log a záznamy aplikácie apache.

Abstract

The thesis deals with server monitoring, focusing on the server logs and storage devices in the form of modules into the KNOTIS information system. An administrator is warned of unusual activities or possible disk failures that may lead to data loss. It describes automatic data collection, data processing and user interface that is developed in the web environment and allows you to set different server monitoring parameters. SMART technology has been used to obtain the status of disk units. The thesis works with the use of disk arrays and LVM technology. It monitors also the most important server logs, such as auth.log, syslog, kern.log and apache's log files.

Klíčové slová

monitoring, monitoring serverov, SMART, systémové záznamy, monitoring diskov, monitoring úložišť

Keywords

monitoring, server monitoring, SMART, server logs, disk monitoring, storage monitoring

Citácia

STRAKA, Ivan. *Monitoring serverů*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Dytrych, Ph.D.

Monitoring serverů

Prehlásenie

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením pana Ing. Jaroslava Dytrycha, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....

Ivan Straka
16. mája 2019

Podakovanie

Rád by som vyjadril nesmiernu vďaku za rady, priateľské konzultácie, trpezlivosť a pripomienky vedúcemu práce pánovi Ing. Jaroslavovi Dytrychovi, Ph.D.

Obsah

1	Úvod	8
2	Špecifikácia požiadavkov	10
2.1	Úložiská	10
2.2	Systémové záznamy	12
2.3	Ethernet rozhrania	12
2.4	Integrácia do systému KNOTIS	13
3	Analýza súčasného stavu	14
3.1	Monitoring úložísk	15
3.2	Ethernet rozhrania	20
3.3	Systémové záznamy	20
3.4	Batérie	20
3.5	Databáza	23
4	Technológie	26
4.1	Webový server	26
4.2	Webový klient	27
4.3	Technológie pre monitorovanie serverov	28
5	Návrh riešenia	33
5.1	Užívateľské rozhranie	33
5.1.1	Úložiská	33
5.1.2	Systémové záznamy	38
5.2	Návrh databázy	39
5.3	Migrácia dát	47
5.4	Zber a spracovanie dát pre úložiská	53
5.4.1	Zber a transformácia dát pre úložiská	53
5.4.2	Zber a transformácia dát pre ethernet rozhrania	58
5.4.3	Spracovanie dát	59
5.5	Zber a spracovanie systémových záznamov	61
6	Implementácia modulu monitoringu úložísk	64
6.1	Migrácia dát	64
6.2	Zber a transformácia dát do formátu JSON	64
6.3	Spracovanie dát	65
6.4	Užívateľské rozhranie	68

7	Implementácia modulu monitoringu systémových záznamov	97
7.1	Migrácia dát	97
7.2	Analýza systémových záznamov	97
7.3	Užívateľské rozhranie	99
8	Testovanie a vyhodnotenie	105
9	Záver	109
	Literatúra	111
A	Sledované atribúty batérie a kondenzátora HW diskového poľa	114
B	Obrázky wireframe	116

Zoznam obrázkov

3.1	Detail servera	14
3.2	Zoznam SW diskových polí	15
3.3	Zoznam HW diskových polí	15
3.4	Zoznam diskov	15
3.5	Detail disku	16
3.6	Smart atribúty disku	17
3.7	História smart atribútu disku	18
3.8	Nastavenie disku	19
3.9	Problémové disky	19
3.10	Zoznam batérií	21
3.11	Detail batérie	21
3.12	História atribútu batérie	22
3.13	Nastavenie parametrov sledovania batérie	23
3.14	ER diagram súčasnej databázy	24
4.1	Cross-platform SMART monitoring	29
5.1	ER diagram úložísk	40
5.2	ER diagram pre stav disku	42
5.3	ER diagram pre stav batérie	44
5.4	ER diagram ethernet rozhraní	45
5.5	ER diagram systémových záznamov	46
6.1	ER diagram SMART skratiek	70
6.2	Pridanie/úprava SMART skratky	71
6.3	Zoznam SMART skratiek	71
6.4	Problémové disky	72
6.5	SMART agregácia	73
6.6	História SMART atribútu disku	74
6.7	Správa diskov	75
6.8	Správa diskov	76
6.9	Grafické zobrazenie histórie SMART atribútu disku	77
6.10	Grafické zobrazenie histórie všetkých SMART atribútov disku	78
6.11	Detail disku	79
6.12	Nastavenie disku	80
6.13	Zoznam diskov	81
6.14	Mazanie diskových polí	82
6.15	Detail HW diskového pola	83
6.16	Detail SW diskového pola	84

6.17	Zoznam HW diskových polí	85
6.18	Zoznam SW diskových polí	86
6.19	Správa batérií	87
6.20	História atribútu batérie	88
6.21	Grafické zobrazenie histórie atribútu batérie	89
6.22	Grafické zobrazenie histórie všetkých numerických atribútov batérie	90
6.23	Detail batérie	91
6.24	Nastavenie batérie	92
6.25	Zoznam batérií	93
6.26	Manažment ethernet rozhraní	94
6.27	Pridanie/úprava ethernet rozhrana	95
6.28	Pridanie/úprava IP adresy	95
6.29	Detail serveru	96
7.1	Detail serveru pre systémové záznamy	100
7.2	Zoznam súborov systémových záznamov	101
7.3	Pridanie/úprava definície pre súbor systémových záznamov	102
7.4	Pridanie/úprava typu súboru systémových záznamov	102
7.5	Zoznam pravidiel pre systémové záznamy	103
7.6	Pridanie/úprava pravidla pre systémových záznamov	104
7.7	Mazanie záznamov súborov systémových záznamov	104
8.1	Počet SMART záznamov – stará verzia	107
8.2	Počet SMART záznamov – nová verzia	107
8.3	Počet záznamov pre monitoring systémových záznamov – stará verzia	108
8.4	Počet záznamov pre monitoring systémových záznamov – nová verzia	108
B.1	Detail servera	117
B.2	Správa ethernet rozhraní servera	118
B.3	Pridanie/úprava ethernet rozhraní servera	119
B.4	Pridanie/úprava ethernet IP servera	120
B.5	Zoznam diskov	121
B.6	Zoznam problémových diskov	122
B.7	Agregácia SMART	123
B.8	Mazanie diskov	124
B.9	Detail disku	125
B.10	Nastavenie parametrov sledovania disku	126
B.11	Nastavenie východiskových hodnôt parametrov sledovania disku	127
B.12	Detail SMART atribútu disku	128
B.13	História self testov disku	129
B.14	Zoznam batérií a kondenzátorov	130
B.15	Detail batérie a kondenzátoru	131
B.16	Detail atribútu batérie a kondenzátoru	132
B.17	Nastavenie parametrov sledovania pre batérie a kondenzátory	133
B.18	Nastavenie východiskových hodnôt parametrov sledovania pre batérie a kondenzátory	134
B.19	Zoznam HW diskových polí	135
B.20	Detail HW diskového pola	136
B.21	Zoznam SW diskových polí	137

B.22 Detail SW diskového pola	138
B.23 Mazanie diskových polí	139
B.24 Zoznam pravidiel pre systémové záznamy	140
B.25 Vytvorenie/úprava pravidla pre systémové záznamy	141
B.26 Zoznam súborov systémových záznamov	142
B.27 Vytvorenie/úprava definície súbory systémových záznamov	143
B.28 Vytvorenie/úprava všeobecnej(východiskovej) definície súbory systémových záznamov	144
B.29 Mazanie záznamov o systémových záznamov	145
B.30 Detail súborov systémových záznamov na serveri	146

Zoznam tabuliek

5.1	Možnosti filtrovania zoznamu diskov	34
5.2	Možnosti filtrovania zoznamu HW a SW diskových polí	37
5.3	Možnosti filtrovania zoznamu batérií a kondenzátorov	38
5.4	Migrácia entity raid_sw	47
5.5	Migrácia entity raid_hw_unit	48
5.6	Migrácia entity raid_hw	48
5.7	Migrácia entity raid	48
5.8	Migrácia entity bbu	49
5.9	Migrácia entity bbu_konst	49
5.10	Migrácia entity bbu_akt	49
5.11	Migrácia entity bbu_default	49
5.12	Migrácia entity disk	50
5.13	Migrácia entity self_test	50
5.14	Migrácia entity smart_konst	51
5.15	Migrácia entity smart_akt	51
5.16	Migrácia entity smart_default	51
5.17	Migrácia entity mount_point	52
5.18	Migrácia entity server_logy	52
5.19	Migrácia entity server_eth	53
5.20	Migrácia entity server_eth	53
5.21	Objekt JSON disku	54
5.22	JSON 10-tica SMART atribútu	54
5.23	JSON šestica SELF testu	55
5.24	JSON päťica SMART chyby	55
5.25	Objekt JSON radiča	55
5.26	Objekt JSON HW diskového pola	55
5.27	Objekt JSON disku v diskovom poli	56
5.28	Objekt JSON batérie a kondenzátora na radiči RAID	56
5.29	JSON štvorica pre atribúty batérie a kondenzátoru	56
5.30	Objekt JSON pre SW RAID	57
5.31	Dvojica JSON pre jednotku v SW RAID	57
5.32	Objekt JSON reprezentujúci bod pripojenia	57
5.33	Objekt JSON reprezentujúci skupinu LVM	58
5.34	Objekt JSON reprezentujúci zväzok LVM	58
5.35	Štvorica pre ethernet rozhranie	58
5.36	Trojica pre ethernet adresu	59
5.37	Trojica pre bránu	59
5.38	11-tica pre súbor systémových záznamov	62

5.39	Trojica pre pravidlo pre súbor systémových záznamov	62
5.40	11-tica pre výsledok analýzy súboru systémových záznamov	63
6.1	Príklad triedy A, ktorá reprezentuje entitu A	67
6.2	Zoznam súborov, ktoré implementujú jednotlivé wireframe z návrhu	69
7.1	Zoznam súborov, ktoré implementujú jednotlivé wireframe návrhy	99
A.1	Zoznam sledovaných parametrov batérií a kondenzátorov	115

Kapitola 1

Úvod

V dnešnom svete sú informácie pre spoločnosti často jednou z najcennejších komodít, ktoré potrebujú patričnú ochranu na rôznych úrovniach. Jednou je uloženie dát a ochrana zariadení. Cenné informácie sa zväčša ukladajú do diskových polí, ktoré ponúkajú rôzne úrovne ochrany dát proti zlyhaniu jednotlivých diskových jednotiek. A ako každý dobrý hospodár, ktorý chce vedieť čo najviac o svojich statkoch – v akom sú stave, tak aj dobrý správca musí vedieť v akom stave sú zariadenia pod jeho správou. Najmä ak obsahujú drahé dáta. Druhou je ochrana serverov, ktoré k dátam majú prístup. Správca potrebuje vedieť, čo sa deje, a patrične reagovať na rôzne rizikové udalosti. Môže sa totiž jednať o cieľný útok ale aj chybu, či nezvyčajnú aktivitu užívateľa systému.

Bežný užívateľ je schopný starať sa o svoje zariadenia sám bez veľkej pomoci s voľne dostupnými nástrojmi. Avšak svet veľkých spoločností pracujúcich s obrovským množstvom dát uložených na mnohých diskoch funguje na iných princípoch.

- Je nemožné, aby jeden správca manuálne monitoroval veľké množstvo zariadení a zároveň je príliš drahé zamestnávať množstvo ľudí, ktorí by mali túto prácu vykonávať – dáta musia byť zbierané a spracovávané automaticky.
- Ak hrozí zlyhanie diskovej jednotky, ktoré môže zapríčiniť stratu cenných prostriedkov, je nutné, aby zodpovedná osoba mala možnosť rýchleho získania potrebných informácií pre zodpovedajúcu reakciu.
- Ak je na serveri nezvyčajne veľká aktivita, systém by ju mal zaznamenať a poskytnúť relevantné informácie správcovi.
- Systém musí poskytovať prehľadné užívateľské rozhranie, z ktorého administrátor vie efektívne získať požadované informácie a ktorého konfigurácia nie je príliš zložitá.

Súčasná technológia umožňuje vytvoriť informačný systém, ktorý tvorí abstraktnú vrstvu medzi správcami a monitorovanými servermi. Práca navrhuje moduly do informačného systému KNOTIS, ktoré zabezpečujú monitorovanie serverov, ich úložísk a systémových záznamov. Kapitola 2 sa venuje analýze požiadaviek a cieľom systému, aké informácie musia byť správcovi prístupné a aké akcie môže vykonávať. Práca nadväzuje na súčasné riešenie, ktoré je podrobené kritike v kapitole 3. Využitie technológií pre vytvorenie systému popisuje kapitola 4. Samotný návrh predkladá obsah kapitola 5. V kapitole je možné nájsť návrh užívateľského rozhrania, ktoré poskytuje správcovi informácie o aktuálnom stave úložísk a možnosť konfigurácie monitorovania a hlásenia udalostí na serveroch. Kapitola následne

popisuje ER diagram modelujúci informácie o serveroch, návrh automatického zberu, spracovania a vyhodnotenia dát.

Implementácia modulov pre monitorovanie úložísk a systémových záznamov je popísaná v kapitolách 6 a 7. Pre každý modul je zdokumentovaný zber dát (a prípadná analýza systémových záznamov) na strane klienta a ich spracovanie systémom KNOTIS. Ďalej je možné vidieť implementované užívateľské rozhranie, pričom sú predstavené rôzne zmeny oproti návrhu za účelom zefektívnenia práce so systémom.

Na záver je v kapitole 8 popísané testovanie modulov a dosiahnuté výsledky riešenia. Konkrétne úspora miesta v databáze, úspora v záťaži siete a rýchlejšie užívateľské rozhranie. V závere 9 sú zhrnuté dosiahnuté výsledky. Plagát je možné nájsť na priloženom nosiči.

Kapitola 2

Špecifikácia požiadavkov

Táto kapitola popisuje požiadavky pre monitoring serverov so zameraním sa na úložiská a systémové záznamy. Riešenie je určené pre správcu serverov, ktorý potrebuje efektívne spravovať disky a zabezpečovať ich výmeny v prípade výskytu poruchy. Systém má napomáhať pri detekcii vadných diskov a má adekvátne upozorňovať na zvýšenú aktivitu serverov.

Systém KNOTIS spravuje viac než 50 serverov, na väčšine ktorých sa nachádza niekoľko diskových jednotiek, z ktorých sú mnohé v diskovom poli RAID. Z dôvodu relatívne vysokého počtu zariadení je nutné, aby bola minimalizovaná potrebná réžia správcu a informačného, a s ním súvisiaceho databázového systému.

2.1 Úložiská

Ak v počítači zlyhá procesor, displej, základná doska alebo iná komponenta, tak síce porucha obmedzí, či dokonca znefunkční počítač, avšak po jej výmene, prípadne oprave je zariadenie opäť funkčné a môžeme pokračovať v predchádzajúcej práci, pre ktorú sú kritické práve dáta. Preto strata, či porucha pamäťových zariadení býva najcitlivejšia. Pri poruche operačnej pamäte strácame aktuálny stav, ale po výmene komponenty môžeme opäť naďalej pracovať. Na druhej strane, pri strate dát z disku je cena omnoho vyššia. Ak neexistuje záloha, tak informácie (veľakrát najhodnotnejší obsah počítača) môžeme navždy stratiť a počítač už nikdy nebude v predchádzajúcom stave. Ak je na disku uložený operačný systém, tak jednoduchá výmena komponenty neprichádza do úvahy.

Ak informácie uložené na diskoch majú vysokú hodnotu, tak je snaha, aby boli zodpovedajúco chránené. K tomu sa najčastejšie využívajú diskové polia, ktoré ponúkajú rôzne formy ochrany proti strate dát, čo sa označuje ako level diskového pola (RAID level) [22]. Diskové pole môže byť hardvérové alebo softvérové.

Hardvérové diskové pole je subsystém nezávislý od operačného systému, s ktorým komunikuje cez zbernicu. Správa patrí do réžie radiča diskového pola, ktorý zvykne poskytovať možnosť definovania viacerých virtuálnych jednotiek s rôznymi konfiguráciami, ktoré sú k dispozícii operačnému systému [22].

Softvérové diskové pole implementuje rôzne RAID úrovne a ponúka lacnejšiu formu využívania diskového pola, nakoľko nie je nutné kupovať iné hardvérové jednotky, ktoré by zabezpečili potrebnú podporu. Technológia pracuje so systémovými zariadeniami, ktoré operačný systém vidí [22].

Diskové polia priamo závisia na diskoch, ktorých zlyhanie prináša riziko straty dát. Diskové polia môžu byť imúnne voči konkrétnemu počtu zlyhaných zariadení, nikdy však nie voči počtu o jedna vyššiemu. Čím sú dáta cennejšie, tým je riziko vyššie a preto je nutné dávať veľký dôraz na prevenciu zlyhaní. I keď pole podporuje napríklad stratu jedného disku, môže nastať situácia, že zlyhajú dva. Tomuto prípadu sa dá predísť, ak máme informáciu o zhoršujúcich sa stavoch týchto dvoch diskov, čo umožňuje patričnú reakciu. Či správca vymení jeden a počká, ako sa situácia vyvinie, alebo sa rozhodne vymeniť obidva, musí jednotky fyzicky vymeniť za nové. Z toho dôvodu je nutné poznať architektúru jednotiek. Správca musí vedieť, ktorú jednotku z akého servera a z akého slotu vybrať a vymeniť. Navyše, v poliach sa častokrát používajú rovnaké zariadenia. Z týchto dôvodov je potrebné, aby systém poskytoval správcovi nasledujúce informácie o architektúre.

- Musí byť zreteľné, na akom serveri sa zariadenie nachádza.
- Musí byť dostupná informácia o bode pripojenia a názvu zariadenia v linuxovom prostredí.
- Ak sa pre zväzok využíva technológia LVM, pre bod pripojenia musí byť uvedený názov logického zväzku.
- O diskovom poli musí mať správca informácie:
 - jednoznačný identifikátor,
 - model, ak sa jedná o fyzickú jednotku.
 - Zoznam diskových jednotiek,
 - adresa v rámci RAID radiča, ak sa jedná o hardvérové pole.
- O diskovej jednotke musí byť známe:
 - sériové číslo a model zariadenia,
 - identifikátor diskového poľa, ak je súčasťou RAID jednotky,
 - jednoznačná identifikácia v rámci diskového poľa, ak je súčasťou nejakej RAID jednotky.

Pri dôraze na prevenciu zlyhaní diskových jednotiek za účelom ochrany cenných informácií je nutné sledovať a vyhodnocovať stav diskov pravidelne a automaticky. Pri veľkom počte diskových jednotiek, ktoré skupina KNOT vlastní (cez 500), je neudržateľné, aby boli jednotky kontrolované ručne.

O diskovom poli alebo disku musí teda správca vedieť, kedy bolo úložisko pridané a naposledy kontrolované. Kontroly prebiehajú periodicky a o každej je potrebné zaznamenávať čas vykonania. Zozbierané dáta majú byť uchovávané perzistentne a efektívne vzhľadom na zabrané miesto, keďže v súčasnom stave cluster obsahuje viac než 500 aktívnych diskov. Ukazovatele je nutné vyhodnocovať, či nedochádza k neočakávanej zmene, čo môže značiť opotrebovanie disku, ktorý má blízko ku koncu svojej životnosti. Ak dôjde k takejto situácii, správca musí byť adekvátne upovedomený.

Ak disk poskytuje možnosť testovania funkčnosti, sú to ďalšie cenné informácie vypovedajúce o stave disku, ktoré treba uchovávať a vyhodnocovať. Musí sa dať nastaviť perióda exekúcie testov a pre sledované atribúty minimálne a maximálne dosiahnuteľné hodnoty. Pri ich presiahnutí systém musí varovať správcu o tejto skutočnosti. Systém umožní nastavenia

systému varovania tak, aby nastavenie bolo efektívne a rýchle. Varovania sú uložené perzistentne, je možné ich prehľadne zobrazíť a manipulovať s nimi (označovať ako prečítané). O kritických udalostiach musí byť správca upovedomený e-mailom.

Diskové polia môžu obsahovať vyrovnávaciu pamäť pre zvýšenie rýchlosti, ktorá v prípade straty napájania môže prísť o dáta. Musí teda obsahovať záložnú batériu, ktorú je nutné sledovať pre elimináciu straty dát. Musia byť sledované jej atribúty, ktoré v súčasnosti systém sleduje.

2.2 Systémové záznamy

Pre skupinu KNOT pracuje niekoľko zamestnancov, ktorí pracujú na mnohých projektoch. Skupina ponúka možnosť seberealizácie aj študentom, čo zvyšuje počet užívateľov výpočtových strojov na nezanedbateľné číslo. Počas práce sa môžu vyskytnúť rôzne úmyselné či neúmyselné incidenty nie len zo strany pracovníkov skupiny, ale aj vonkajších užívateľov Internetu. Pod incidentom si je možné predstaviť útok na server za účelom získania kontroly (napr. bruteforce útok na služby SSH) alebo chyba v programe pracovníka, ktorá neúmerne vyťažuje sieť, či konkrétny zdroj Internetu, čo môže byť považované poskytovateľom ako DoS útok.

Na serveroch je nainštalovaný operačný systém Ubuntu, ktorý vedie systémové záznamy o rôznych udalostiach.

- auth.log – súbor obsahuje informácie týkajúce sa autentizácie
- kern.log – súbor obsahuje detailné záznamy týkajúce sa jadra operačného systému
- syslog – obsahuje väčšinu správ operačného systému

Skupina KNOT využíva niekoľko serverov Apache, ktorých záznamy majú byť tiež sledované:

- access.log – obsahuje záznamy o udalostiach o prístupoch, ktoré sa udiali na serveri
- error.log – obsahuje chybové záznamy servera
- ssl_access.log – obdoba access.log s rozdielom, že sa týka zabezpečeného SSL spojenia

Systém musí sledovať systémové záznamy a záznamy webového servera Apache a upozorňovať na nezvyčajnú aktivitu správcu. Systém musí umožniť filtrovanie záznamov, čo pomôže lepšej detekcii problémových incidentov. KNOTIS musí sledovať nárast veľkosti súborov záznamov a upozorňovať na neúmerne zvyšovanie ich veľkostí, čo vypovedá o zvýšenej aktivite, ktorá môže stať za preskúmanie.

2.3 Ethernet rozhrania

Servery spadajúce pod skupinu KNOT sú pripojené do internetovej siete cez ethernet linku, pričom správna konfigurácia je kritická pre správne fungovanie strojov. Preto monitorovanie ethernet rozhraní umožní rýchle overenie v prípade sieťového problému.

Internetová sieť na VUT FIT má množstvo používateľov a neraz sa stane pri vytváraní vlastných podsietí chybné ohlasovanie umiestnenia brány v sieti. Či už sa jedná o úmysel,

alebo nie, v prípade, ak je táto informácia propagovaná na server spadajúci pod skupinu KNOT, je nežiadúce, aby server túto bránu využíval. Preto je nutný monitoring brán pre servery.

2.4 Integrácia do systému KNOTIS

Monitoring má byť riešený formou modulov do informačného systému KNOTIS, ktoré musia dodržiavať zavedenú kultúru. To znamená využívanie vytvorených šablón a funkcií pre zobrazovanie informácií. Primárne má byť dodržiavaný GNU štandard kódovania [8].

Analýza súčasného stavu

Táto kapitola pojednáva o aktuálnych možnostiach monitoringu diskov, systémových záznamov a ethernet rozhrania, ktoré ponúka systém KNOTIS pre správcu. V jednotlivých podkapitolách sú popísané náhľady na zobrazenia a nastavenia monitoringu sprevádzané fotkami obrazoviek vytvorenými autorom práce. Systém je podrobený konštruktívnej kritike.

Systém pre každý server zobrazuje, aké úložiská, rozhrania a batérie pre zálohovacie jednotky obsahuje (obr. 3.1). Správca má prehľad o architektúre zapojenia diskov a o ich aktuálnom stave, či histórii jednotlivých ukazovateľov stavu. Prezentácia informácií je dostatočná, správca má k dispozícii všetky informácie, ktoré potrebuje. Problémom je najmä neprehľadnosť stavu disku a jeho histórie. Správca je nedostatočne upozornený o dôvode označenia disku ako problémového (zrelého na výmenu).

KNOTIS

Detail serveru

Přihlásit

- Osobní údaje
- Projevy
- Vlastní projekty
- Přehled projektů
- Poslední vyšetření
- Údaje
- Dávání výkazy
- Tiskování výkazů
- Plánovaná DPP
- Plánovaná školení

- Pracovní výkazy
- Plán práce
- Rukovně projektů
- Učební
- Očekávané e-mail

- Historie - napájení L
- Historie - napájení v t.t.
- Historie - napájení k sítě
- Historie - plán o hodu
- Historie - bodový systém

- Více úrovní
- Testy zátěže
- Vlastní chyby
- Rukovně
- Servery
- Uživatelské

Informace o serveru:

Název: athens2
Doména: tl.vultr.cz
Autentifikace: athens2_AlBAzCVVYmuhbmF-JwtoMWWnHAIATDmuGSPRhdKvntloWzUEBHfB
Datum výroby: 10.07.2013
Datum kontroly: 14.09.2016

Upravit informace

Zatít

ETH rozhraní

ETH	MAC	IP adresy	Břídina
eth2	00:25:90:c8:3ac6	147.229.8.39 2001.87c:1220:808:30:a6:027 fe80::225:90ff:fac8:3ac6 2001.87c:1220:808:30:a6:027 fe80::225:90ff:fac8:3ac6	147.229.8.1 2001.87c:1220:808:-1
eth1	00:25:90:98:30:b3		
eth0	00:25:90:98:30:b3		

Disky

Sériové číslo	Model	Adresa	Bod přípojiny	Kapacita	Obsazená kapacita	Obsazeno i užítí	Stav	Datum kontroly	Datum výroby
WD-WXSS1E48RW1DA	WDC WD1000CHTZ-0ALCPV1	ata1	/	100024889016 (1 TiB)	579231528 / 9322492654 (579.32 GiB / 932.27 GiB) 62.14%	480698 / 59211776 0.81%		14.09.2016	13.07.2016

RAID HW

Sériové číslo	Model	Zařízení	Bod přípojeiny	Raid level	Kapacita	Obsazená kapacita	Obsazeno i užítí	Stav	Datum kontroly	Datum výroby
EVS3210187	LSI MegaRAID SAS 9200-1644	sib	mraidata	RAID6	37655.17 [37.26 TiB]	16633001544 / 39062811968 (16.63 TiB / 39.06 TiB) 42.58%	2781700 / 3906449888 0.97%	Opt	14.09.2016	20.03.2014

RAID SW

UUID	Zařízení	Bod přípojeiny	Raid level	Kapacita	Obsazená kapacita	Obsazeno i užítí	Stav	Datum kontroly	Datum výroby
------	----------	----------------	------------	----------	-------------------	------------------	------	----------------	--------------

Baterie

Seriál	Typ	Raid	Název výrobce	Název zařízení	Kapacita	Napětí	Datum kontroly	Datum výroby	Datum výroby
651	dBL008	EVS3210187	LSI36691	bq27541	1500 mAh	4.1 V	13.07.2016	07.12.2015	13.06.2014

Zařízení serveru

Procesory

Obr. 3.1: Detail servera

Najväčšou slabinou je databáza, keďže model nereflektuje realitu, čo popisuje podkapitola 3.5. Skript pre zber dát posiela veľa neúčinných informácií na server pre ďalšie spracovanie, ktoré navyiac obsahuje veľa chýb.

3.1 Monitoring úložísk

Súčasný stav umožňuje prehľadný výpis softvérových a hardvérových diskových polí a ich najnutnejších atribútov ako RAID level, kapacita, dátum vloženia a iné. Dáta sú zobrazené v jednoduchéj tabuľke s možnosťou stránkovania a filtrovania podľa jednotlivých atribútov. Filtre nie je možné uložiť, pričom tlačítka pre uloženie sú prítomné. Náhlady obsahujú menšie chyby, napríklad kapacita v bitoch, odkazy na detail RAID jednotky sú umiestnené na modeloch radiča (radič môže obsahovať viac RAID jednotiek). Inak sú náhlady relatívne v poriadku. Náhľad na zoznam diskov je v rovnakom štýle s rovnakými nedostatkami (obr. 3.2 – 3.4).

KNOTIS **Seznam SW diskových polí**

Príhlásen:

[?](#) [Nápověda](#) [Osobní nastavení](#) [Odhlášení](#)

Nastavit filtry Vymazat filtry Uložit filtry +

Server A.Y.A.Y	Zařízení A.Y.A.Y	Bod připojení A.Y.A.Y	Raid level A.Y.A.Y	Kapacita A.Y.A.Y	Využití A.Y.A.Y	Stav A.Y.A.Y	Datum kontroly A.Y.A.Y	UUID A.Y.A.Y	Datum vložení A.Y.A.Y
knot01	md0	/mnt/data	raid5	17581166592 (17.58 TiB)	15749388432 (15.75 TiB)	clean	14.09.2016	6852a337-391414b-49ac3275-20ad0b30	19.12.2014
knot03	md2	/mnt/data	raid5	17581166592 (17.58 TiB)	15684389216 (15.68 TiB)	clean	14.09.2016	e1228678-aab9a927a-f9a7b3-00c20b6	20.12.2014
knot04	md0	/mnt/data	raid5	17581174080 (17.58 TiB)	15617549024 (15.62 TiB)	active	14.09.2016	66c25d53-a456b0a8-a9b91a8-5e2b92a8	08.10.2015
knot05	md0	/mnt/data	raid5	17581174080 (17.58 TiB)	15575451364 (15.58 TiB)	clean	14.09.2016	8a272789-54b8b65d-91572491-5d4dd4d4	10.10.2015
knot06	md3	/mnt/data	raid5	17581166592 (17.58 TiB)	15635513104 (15.64 TiB)	clean	14.09.2016	61b8b149-4dc299903-a7b802cf-8308b2a9	20.12.2014
knot07	md1	/mnt/data	raid5	17581166592 (17.58 TiB)	15550642408 (15.55 TiB)	clean	14.09.2016	ab7a0c43-39a42127-60bafdbd-02912d5d	13.01.2015
knot08	md0	/mnt/data	raid5	17581166592 (17.58 TiB)	15597315236 (15.6 TiB)	clean	14.09.2016	a7a14d1a-fab2080c-822d8f21-655a3219	13.01.2015
knot09	md0	/mnt/data	raid5	5880270720 (5.88 TiB)	2052348092 (2.05 TiB)	clean	14.09.2016	2c8e8773-c0a8938c-ab53c0cf-3b184ded	14.07.2013

Obr. 3.2: Zoznam SW diskových polí

KNOTIS **Seznam HW diskových polí**

Príhlásen:

[?](#) [Nápověda](#) [Osobní nastavení](#) [Odhlášení](#)

Nastavit filtry Vymazat filtry Uložit filtry +

Server A.Y.A.Y	Zařízení A.Y.A.Y	Bod připojení A.Y.A.Y	Raid level A.Y.A.Y	Kapacita A.Y.A.Y	Stav A.Y.A.Y	Datum kontroly A.Y.A.Y	Model A.Y.A.Y	Sériové číslo A.Y.A.Y	Datum vložení A.Y.A.Y
minerva1	sd0	/mnt/data	RAID-6	78230.9 (78.23 TiB)	OK	14.09.2016	9750-164a	SV13300210	11.04.2016
minerva1	sd0	/mnt/data-2	RAID-6	39115.4 (39.12 TiB)	OK	14.09.2016	9750-164a	SV13300210	15.07.2013
minerva1	sd0	/mnt/data-in	RAID-6	24214.3 (24.21 TiB)	OK	14.09.2016	9750-164a	SV10712505	18.07.2013
minerva1	sd0	/mnt/data-3	RAID-6	28076.9 (28.08 TiB)	OK	14.09.2016	9750-164a	SV10712505	15.07.2013
athena2			Cac0	223.06 (223.06 GiB)	Opti	14.09.2016	LSI MegaRAID SAS 9200-164a	SV32210187	20.03.2014

Obr. 3.3: Zoznam HW diskových polí

KNOTIS **Seznam disků**

Príhlásen:

[?](#) [Nápověda](#) [Osobní nastavení](#) [Odhlášení](#)

Nastavit filtry Vymazat filtry Uložit filtry +

Model A.Y.A.Y	Serial A.Y.A.Y	Kapacita A.Y.A.Y	Server A.Y.A.Y	RAID	Datum kontroly A.Y.A.Y	Stav A.Y.A.Y	Bod připojení A.Y.A.Y	Zařízení / Port A.Y.A.Y	Datum vložení A.Y.A.Y	Aktivní A.Y.A.Y
INTEL SSDRC2B480G04	8THC60210802480G04	480103981056 (480.1 GiB)	minerva1	-	14.09.2016	-	-	ssd1	06.06.2016	ano
KINGSTON SKC1908	248B60001321	238510098745 (239.51 GiB)	athena2	SV32210187245.13	14.09.2016	Online	-	-	20.03.2014	ano
QCY TRON150	284B6200K10LJ	480103981056 (480.1 GiB)	minerva2	-	14.09.2016	-	-	ssd1	06.06.2016	ano
Samsung SSD 850	S1SLRNWAF860323Z	255517268115 (255.52 GiB)	athena2	SV34340234245.15	14.09.2016	Online	-	-	16.01.2015	ano
SEAGATE ST3300065SS	30PTVCAH	299999999488 (300 GiB)	minerva3	L340865A9360158c0p17	13.06.2016	OK	/mnt/data-2	ssd	23.03.2016	ne
WDC WD10EALX-60BB0A0	WD-WCATR7388513	1008204886016 (1 TiB)	ssd	-	14.09.2016	-	/mnt/data	ssd1	19.07.2013	ano
WDC WD3600FF-YY-62WB1	WD-WCAY0845796	2000398934016 (2 TiB)	minerva3	L340865A9360158c0p23	24.03.2016	SMART-FAILURE	/mnt/data	ssd	23.03.2016	ne
WDC WD3600FF-YY-62WB1	WD-WMC111719692	3000592982016 (3 TiB)	minerva1	-	14.09.2016	-	/mnt	ssd1	15.07.2013	ano
WDC WD3600FF-YY-62WB1	WD-WMC111318013	3000592982016 (3 TiB)	athena2	md0 (athena2)	14.09.2016	active sync	/mnt/data	md0	15.07.2013	ano

Obr. 3.4: Zoznam diskov

Náhľad na detail disku ponúka tabuľku s bližšími informáciami o stave disku, kde sú zobrazené SMART atribúty, ktoré sú poskytované diskovými jednotkami [9]. Systém zo-

brazuje ich aktuálne raw hodnoty a hodnoty parametrov sledovania pre účel kontroly. Pod tabuľkou sú atribúty opäť vypísané, pričom je možné zobrazíť ich poslednú a celkovú históriu v novom okne. Nasledujú informácie o self testoch [24] nástroju smartmontools (viď obr. 3.5).

KNOTIS

- Osoby
 - Vložit osobu
- Projekty
 - Vložit projekt
 - Skupiny projektů
- Průhledy
 - Úkoly
 - Denní výkazy
 - Týdenní výkazy
 - Přehled DPP
 - Přehled Spondii
- Pracovní výkazy
- Plán práce
- Řešené projekty
- Údělba
- Odeslané e-maily
- Hříšníci - nepostali z.
 - Hříšníci - moc h. v 1 t.
 - Hříšníci - moc h. k odp.
 - Hříšníci - plán. 0 hodin
 - Hříšníci - bodový systém
- Wiki úkoly
- Texty zpráv
- Všechny osoby
- Řešitelé
- Severny
- Události

Detail disku

Informace o disku:

Model: WDC WD2003FYYS-02W0B1

Sériové číslo: WD-WCAY00457396

Kapacita: 2000398934016 [2 TiB]

Server: [minerva3](#)

RAID: L340605AS360156

Datum kontroly: 24.03.2016

Stav: SMART-FAILURE

Bod připojení: /mnt/data

Připojené zařízení: sdc

Adresa disku: /c2/p23

Datum vložení: 23.03.2016

[Upravit nastavení disku](#) [Zpět](#)

Aktuální hodnoty

Název hodnoty	Hodnota	RAW hodnota	Nejhorší hodnota	Prah	Max. změna	Max. hodnota	Max. změna nejhorší hodnoty	Selhalo	Čas kontroly
Raw Read Error Rate	182	509036	182	51	1	509036	1	-	2016-03-23 22:18:38

[Rozbalit vše](#) [Skrýt vše](#)

Raw Read Error Rate [historie](#)

Att_name	Thresh	Type	Update	Max_hodnota	Max_zmena
Raw_Read_Error_Rate	51	Pre-fail	Always	509036	1

Hodnoty Raw_Read_Error_Rate

RAW	Hodnota	Worst	When_fail	Kontrola
509036	182	182	-	2016-03-24 14:09:53
509036	182	182	-	2016-03-24 02:33:37
509036	182	182	-	2016-03-23 22:18:38

SELF Testy [historie](#)

Typ	Status	Stav hodin	První výskyt chyby	Čas kontroly
extended offline	ERROR - Completed: unknown failure	29569	-	24.03.2016

[Rozbalit vše](#) [Skrýt vše](#)

Obr. 3.5: Detail disku

V prípade viacerých atribútov sa náhľad stáva neprehľadný (obr. 3.6). Podobne je to aj pri prezeraní celkovej histórie atribútu, kde sú uložené všetky doposiaľ namerané záznamy. Pri mnohých opakujúcich sa hodnotách nastáva sťažená orientácia. Tento prístup je však aj technickým problémom. Nie len, že je vyžadovaná vysoká pamäťová náročnosť pri správe mnohých diskových jednotiek, ale aj výpočtová pri práci s dátami a časová náročnosť pri získavaní dát z databázy (viď obr. 3.7).

KNOTIS

- Osoby
 - Vložit osobu
- Projekty
 - Vložit projekt
- Skupiny projektů
- Prostředky
- Úkoly
 - Denní výkazy
 - Týdenní výkazy
 - Přehled DPP
 - Přehled Stipendií
- Pracovní výkazy
- Plán práce
- Řešené projekty
- Údržba
- Odeslané e-maily
- Hříšníci - neposlali z.
- Hříšníci - moc h. v 1 t.
- Hříšníci - moc h. k odp.
- Hříšníci - plán. 0 hodin
- Hříšníci - bodový systém
- Wiki úkoly
- Texty zpráv
- Všechny osoby
- Řešitelé
- Servery
- Události

Count										22:18:39
Multi Zone Error Rate	1	75684	1	0	1	75684	1	-		2016-03-23 22:18:39

Rozbalit vše

Skrýt vše

Raw Read Error Rate

historie

Spin Up Time

historie

Att_name	Thresh	Type	Update	Max_hodnota	Max_zmena
Spin_Up_Time	21	Pre-fail	Always	10000	1

Hodnoty Spin_Up_Time

RAW	Hodnota	Worst	When_fail	Kontrola
8525	253	253	-	2016-03-24 14:09:53
8525	253	253	-	2016-03-24 02:33:37
8525	253	253	-	2016-03-23 22:18:38

Start Stop Count

historie

Reallocated Sector Ct

historie

Seek Error Rate

historie

Power On Hours

historie

Spin Retry Count

historie

Calibration Retry Count

historie

Power Cycle Count

historie

Power-Off Retract Count

historie

Load Cycle Count

historie

Temperature Celsius

historie

Reallocated Event Count

historie

Current Pending Sector

historie

Offline Uncorrectable

historie

Obr. 3.6: Smart atribúty disku

KNOTIS <ul style="list-style-type: none"> Osoby <ul style="list-style-type: none"> Všetř osobu Projekty <ul style="list-style-type: none"> Všetř projekt Skupiny projektu Prostředky <ul style="list-style-type: none"> Úlohy Denní výkazy Týdenní výkazy Přehled DPP Přehled Spenzů Pracovní výkazy <ul style="list-style-type: none"> Plan práce Resumé projektu Údělba Odstávané e-mail História - nepostlali z. História - moc h. v. 1.1. História - moc h. k. odp. História - plan. 0 hodin História - bodový systém Vše úlohy Texty zpráv Všechny osoby Resumé Servery Události 	25	100	100	-	2016-07-09 02.33.21
	25	100	100	-	2016-07-08 02.33.28
	25	100	100	-	2016-07-07 02.33.27
	25	100	100	-	2016-07-06 02.33.29
	25	100	100	-	2016-07-05 02.33.27
	25	100	100	-	2016-07-04 02.33.27
	25	100	100	-	2016-07-03 02.33.29
	25	100	100	-	2016-07-02 02.33.28
	25	100	100	-	2016-07-01 02.33.29
	25	100	100	-	2016-06-30 02.33.28
	25	100	100	-	2016-06-29 02.33.28
	25	100	100	-	2016-06-28 02.33.29
	25	100	100	-	2016-06-27 02.33.31
	25	100	100	-	2016-06-26 02.33.32
	25	100	100	-	2016-06-25 02.33.27
	25	100	100	-	2016-06-24 02.33.25
	25	100	100	-	2016-06-23 02.33.28
	25	100	100	-	2016-06-22 02.33.27
	25	100	100	-	2016-06-21 02.33.28
	25	100	100	-	2016-06-20 02.33.27
	25	100	100	-	2016-06-19 02.33.28
	25	100	100	-	2016-06-18 02.33.27
	25	100	100	-	2016-06-17 02.33.28
	25	100	100	-	2016-06-16 02.33.29
	25	100	100	-	2016-06-15 02.33.21
	25	100	100	-	2016-06-14 02.33.21
	25	100	100	-	2016-06-13 02.33.24
	25	100	100	-	2016-06-12 02.33.23
	25	100	100	-	2016-06-11 02.33.22
	25	100	100	-	2016-06-10 02.33.20
	25	100	100	-	2016-06-09 02.33.21
	25	100	100	-	2016-06-08 02.33.23
	25	100	100	-	2016-06-07 02.33.23
	Předchozí 1 2 3 4 5 6 7 Další				

Obr. 3.7: História smart atribútu disku

Pre disk je možnosť nastavenia parametrov sledovania jednotlivých atribútov. Je možné nastaviť maximálnu hodnotu, čo sa aplikuje na raw hodnotu atribútu. Maximálna zmena worst hodnoty značí, o koľko sa môže hodnota worst atribútu zmeniť. Keď sú tieto hodnoty dosiahnuté, systém vygeneruje udalosť a zašle e-mail správcovi, pričom je možné vypnúť sledovanie parametrom analýza. Je možné nastaviť periódu vykonávania self testov nástroju smartmontools. Správca má možnosť nastaviť východiskové parametre sledovania pre SMART atribúty, avšak, tie sa využívajú iba pri detekovaní nového disku. Na tieto hodnoty sú nastavené parametre sledovania disku. Tiež si musí pamätať aktuálne raw hodnoty

atribútov, ak ich potrebuje zohľadniť (viď obr. 3.8). Východiskové hodnoty sú nastaviteľné rovnakým spôsobom v inej časti systému.

KNOTIS

- Osoby
 - Vložit osobu
- Projekty
 - Vložit projekt
- Skupiny projektů
- Prostředky
- Úkoly
- Denní výkazy
- Týdenní výkazy
- Přehled DPP
- Přehled Stipendií

- Pracovní výkazy
- Plán práce
- Řešené projekty

- Údržba

- Odeslané e-maily

- Hříšníci - neposlali z.

Úprava disku

Nastavení pro správu tohoto disku:

Nastavení hodnot pro self testy:

Self short: dnů hodin *

Self long: dnů hodin *

Nastavení hodnot pro SMART:

Nastavit výchozí hodnoty

max_zmena
max_hodnota
max_zmena_worst
analýza

Raw Read Error Rate ☒

Throughput Performance ☒

Uložit
Zpět

Obr. 3.8: Nastavenie disku

Systém poskytuje zoznam problémových diskov, ktorý zobrazuje rovnaké hodnoty ako zoznam diskov s pridaním popisu problému. Disk je označený ako problémový, ak SMART atribút klesne pod threshold hodnotu alebo jeho raw hodnota je nad povoleným maximom. Výpis však danú hodnotu nezobrazuje a správca musí sám zanalyzovať SMART hodnoty disku a zistiť, prečo je disk označený ako problémový. Správca nemá žiadnu možnosť úpravy nastavenia sledovania daného atribútu na mieste. Náhľad kopíruje chyby nachádzajúce sa v detaile disku (viď obr. 3.9).

KNOTIS

- Osoby
 - Vložit osobu
- Projekty
 - Vložit projekt
- Skupiny projektů
- Prostředky
- Úkoly
- Denní výkazy
- Týdenní výkazy
- Přehled DPP
- Přehled Stipendií

- Pracovní výkazy
- Plán práce
- Řešené projekty

- Údržba

- Odeslané e-maily

- Hříšníci - neposlali z.

Problémové disky

Nápověda
Ochrana nastavení
Odrážení

Model A.T.S.V	Serial A.T.S.V	Kapacita A.T.S.V	Server A.T.S.V	RAID	Datum kontroly A.T.S.V	Stav A.T.S.V	Bod připojení A.T.S.V	Zařízení / Port A.T.S.V	Datum vložení A.T.S.V	Chybné atributy
INTEL SSDSC2BK480G4	BTHC8012087C480MGN	480103981056 (480.1 GiB)	minerva3	-	14.09.2016	/	/	sd1	15.06.2016	Program_Fail_Count_Chip
INTEL SSDSC2BK480G4	BTHC8021014P480MGN	480103981056 (480.1 GiB)	nov02	-	14.09.2016	/	/	sd1	13.07.2016	Unknown_Attribute
INTEL SSDSC2BK480G4	BTHC8021014U480MGN	480103981056 (480.1 GiB)	nov02	-	14.09.2016	/	/	sd1	06.06.2016	Unknown_Attribute
INTEL SSDSC2BK480G4	BTHC80210219480MGN	480103981056 (480.1 GiB)	nov02	-	14.09.2016	/	/	sd1	06.06.2016	Unknown_Attribute
KINGSTON SKC100S	244B60001321	238510998745 (238.51 GiB)	minerva2	SV32210187245.13	14.09.2016	Online	/	sd1	20.03.2014	Load_Cycle_Count
OCZ ARD100	A20L2061519001220	480103981056 (480.1 GiB)	nov02	-	14.09.2016	/	/	sd1	02.06.2016	Unknown_Attribute
ST1000DM001-1NS162	Z4Y9N4N75	1008204886016 (1 TiB)	-	-	02.09.2016	/	/	sd1	18.07.2016	Raw_Read_Err_Rate Start_Stop_Count Seek_Error_Rate
ST3000DM001-1GH166	W1F2ZC86	300859282016 (3 TiB)	athens1	-	14.09.2016	/	mm00cweeb09	sd1	16.01.2014	Power_On_Hours

Obr. 3.9: Problémové disky

Zber a analýza informácií

Zber hodnôt vypovedajúcich o stave disku je vykonávaný automaticky pomocou cron úlohy raz denne. Skript zozbiera informácie o diskových poliach a o priradených diskových jednotkách. Skript získa hodnoty SMART atribútov všetkých diskových jednotiek na serveri. Výstup posiela informačnému systému KNOTIS využitím protokolu HTTP. KNOTIS následne spracuje vstup a ukladá dáta do databázy. Výstup zo skriptu obsahuje veľa neužitočných informácií z aplikácií získavajúcich informácie o poliach a diskoch v poliach. Aj výstup zo

smartmontools aplikácie obsahuje množstvo neužitočných informácií a tak dochádza k zbytočnému nadužívaniu kapacity siete. Najmä ak sa zoberie do úvahy fakt, že dáta odosielané na server majú veľkosť v rozmedzí 100 – 400KB, pričom cluster v súčasnosti obsahuje viac než 508 aktívnych diskových jednotiek.

Skript je relatívne dobre napísaný, i keď občas až moc zložito, pričom je príliš prispôsobený spôsobu používania technológií. To vyúsťuje do niekoľkých vážnych chýb. Prvou je, že skript nepredpokladá rozdelenie RAID jednotky na viac logických zväzkov technológiou LVM. V druhom prípade skript predpokladá, že LVM zväzok nebude šifrovaný. Skript mapuje disky zapojené v poli rodiny MegaRaid na jednotlivé zariadenia (sda) podľa kapacity diskového pola a kapacity známych bodov pripojenia. Metóda nemusí vrátiť správny výsledok pre podobne veľké diskové polia a tiež zlyháva, ak RAID má viac partícií a teda viac bodov pripojenia.

PHP skript pre spracovanie dát má viaceré nedostatky. Najzávažnejšou chybou je časté hľadanie špecifických reťazcov, ktoré sú produkované iba určitými modelmi diskov a konkrétnymi verziami programov, čo má za následok chybné spracovanie dát a následné uloženie do databázy. Skript je neprehľadný a zmeny sa doň zavádzajú veľmi ťažko. Navyše v súčasnosti nie je schopný spracovať stav, keď sa na jednom serveri nachádza RAID rodiny MegaRaid a ďalší rodiny LSI 3ware.

3.2 Ethernet rozhrania

Ethernet rozhrania sú zobrazené jednoducho a prehľadne v náhľade na detail serveru. Systém chybné zobrazuje adresu všesmerového vysielania ako adresu brány a vyžaduje tradičné pomenovanie rozhraní (eth0, eth1, ..., ktoré je nepredvídateľné [18], pričom moderné operačné systémy od neho upúšťajú. Konkrétne, táto metóda pomenovania je až posledná možná [5], [11]. Viď obr. 3.1.

3.3 Systémové záznamy

Monitoring serverových záznamov zahŕňa spracovávanie dôležitých súborov – auth.log, server.log, ... Pre každý súbor záznamov na každom serveri je možné nastaviť maximálny možný nárast veľkosti, po ktorého prekročení nastáva vygenerovanie udalosti a zaslanie e-mailu správcovi. Tento prístup vyúsťuje do enormného posielania e-mailov vždy, keď sa zvýši či už dátová prevádzka na serveri, alebo nejaký užívateľ testuje svoj program, ktorý generuje záznamy, alebo iné. Dôsledkom je veľká neprehľadnosť a možné nepostrehnutie dôležitej informácie.

3.4 Batérie

Niektoré hardvérové polia využívajú vyrovnávaciu pamäť, ktorá musí mať k dispozícii batériu pre uchovanie informácií počas výpadku napájania kvôli zachovaniu integrity dát. Systém ponúka zobrazenie zoznamu všetkých batérií, o ktorých zobrazuje základné informácie ako sériové číslo, typ, umiestnenie, názov, kapacitu, aktuálne napätie a iné (viď obr. 3.10). Detail batérie (obr. 3.11), história hodnôt atribútu (obr. 3.12) a nastavovanie parametrov sledovania atribútov (obr. 3.13) sú v podobnom štýle ako detail disku s rovnakými chybami – neprehľadnosť, strácanie sa v informáciách.

Niektoré vyrovnávacie pamäte diskových polí využívajú miesto klasickej batérie vysokokapacitný kondenzátor, ktorý sa líši nie len jednotkou kapacity, ale aj atribútmi vypo-vedajúcimi o stave kondenzátora. Systém s takouto možnosťou nepočíta a ani neponúka podporu pre iný typ zariadenia pre uchovávanie energie.

KNOTIS **Seznam baterií** Přihlášen:

? Nápověda Osobní nastavení Odhlášení

Nastaví filtr
Vymaže filtr
Uloží filtr

Serial ▲▼▲▼	Typ ▲▼▲▼	Server ▲▼▲▼	Raid ▲▼▲▼	Název výrobce ▲▼▲▼	Název zařízení ▲▼▲▼	Kapacita ▲▼▲▼	Napětí ▲▼▲▼	Datum kontroly ▲▼▲▼	Datum vložení ▲▼▲▼	Datum výroby ▲▼▲▼	Aktivní ▲▼▲▼
8808	IBBU08	athens2	SV43402534	LS36691	bq27541	1500 mAh	4.1 V	14.09.2016	07.12.2015	13.10.2014	ano
551	IBBU08	athens2	SV32210187	LS36691	bq27541	1500 mAh	4.1 V	13.07.2016	07.12.2015	13.09.2014	ne

Obr. 3.10: Zoznam batérií

KNOTIS **Detail baterie** Přihlášen:

? Nápověda Osobní nastavení Odhlášení

Informace o baterii:

Serial: 8808

Název zařízení: bq27541

Název od výrobce: LS36691

Typ: IBBU08

Server: athens2

Raid: SV43402534

Napětí: 4.1 V

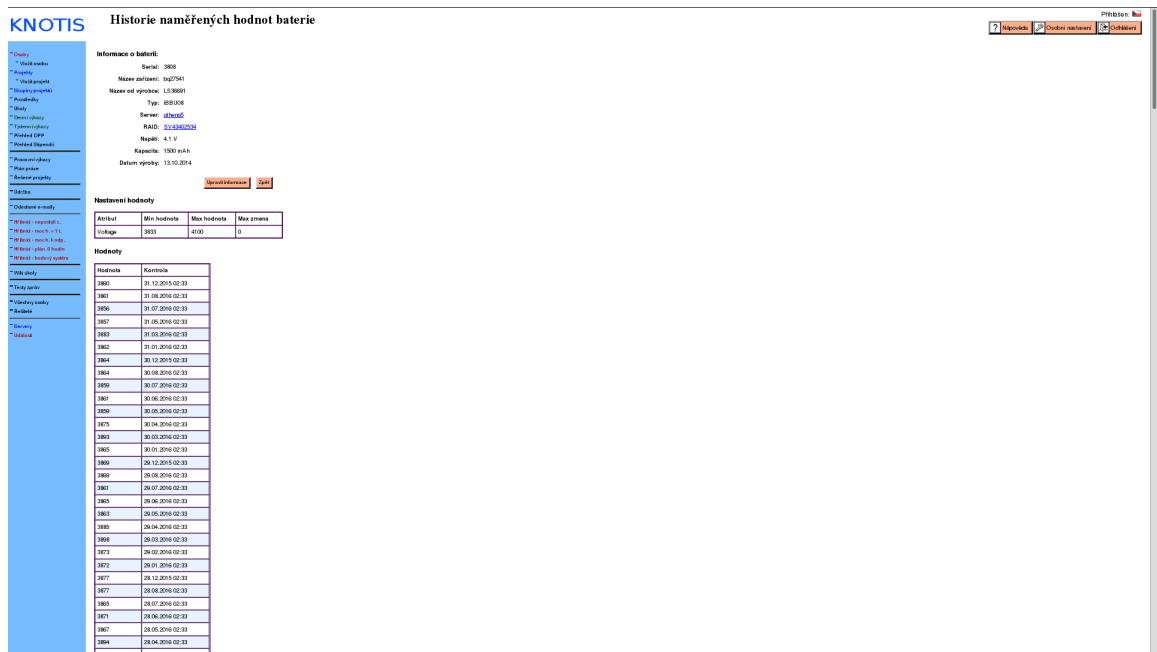
Kapacita: 1500 mAh

Datum výroby: 13.10.2014

Aktuální hodnoty

Název hodnoty	Hodnota	Min. hodnoty	Max. hodnoty	Čas hodnoty
Absolute State of charge	99	30	100	2015-12-07 20:03:06
Cycle Count	2	0	0	2015-12-07 20:03:06
Full Charge Capacity	1347	1300	2000	2015-12-07 20:03:06
Temperature	OK	0	0	2015-12-07 20:03:06
Voltage	OK	0	0	2015-12-07 20:03:06
Fully Charged	No	0	0	2015-12-07 20:03:06
Fully Discharged	No	0	0	2015-12-07 20:03:06
Max Error	0	0	0	2015-12-07 20:03:06
Over Charged	No	0	0	2015-12-07 20:03:06
Over Temperature	No	0	0	2015-12-07 20:03:06
Battery State	Optimal	0	0	2015-12-07 20:03:06
Current	0	0	0	2015-12-07 20:03:06
Temperature	21	0	25	2015-12-07 20:03:06
Voltage	2853	2823	4100	2015-12-07 20:03:06
Auto Learn Period	2960000	0	2960000	2015-12-07 20:03:06
Next Learn Time	2015-12-18 16:46:10	0	0	2015-12-07 20:03:06

Obr. 3.11: Detail batérie



Obr. 3.12: História atribútu batérie

Osoby

Vložit osobu

Projekty

Vložit projekt

Skupiny projektů

Prostředky

Úkoly

Denní výkazy

Týdenní výkazy

Přehled DPP

Přehled Stipendií

Pracovní výkazy

Plán práce

Řešené projekty

Údržba

Odeslané e-maily

Hřěnki - nepodali z.

Hřěnki - moch. v 1 t.

Hřěnki - moch. k odp.

Hřěnki - plán. 0 hodin

Hřěnki - bodový systém

Wiki úkoly

Texty zpráv

Všechny osoby

Řešitelé

Servery

Události

Nastavení pro správu této baterie:

Nastavení hodnot pro Baterii:

Nastavit výchozí hodnoty

	max_zmena	max_hodnota	min_hodnota	analýza
Voltage	<input type="text" value="0"/>	<input type="text" value="4100"/>	<input type="text" value="3833"/>	<input checked="" type="checkbox"/>
Current	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Temperature	<input type="text" value="0"/>	<input type="text" value="21"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Battery State	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Voltage	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Temperature	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Learn Cycle Requested	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Learn Cycle Active	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Learn Cycle Status	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
I2C Errors Detected	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Replacement required	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Remaining Capacity Low	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
Periodic Learn Required	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>
No space to cache offload	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input checked="" type="checkbox"/>

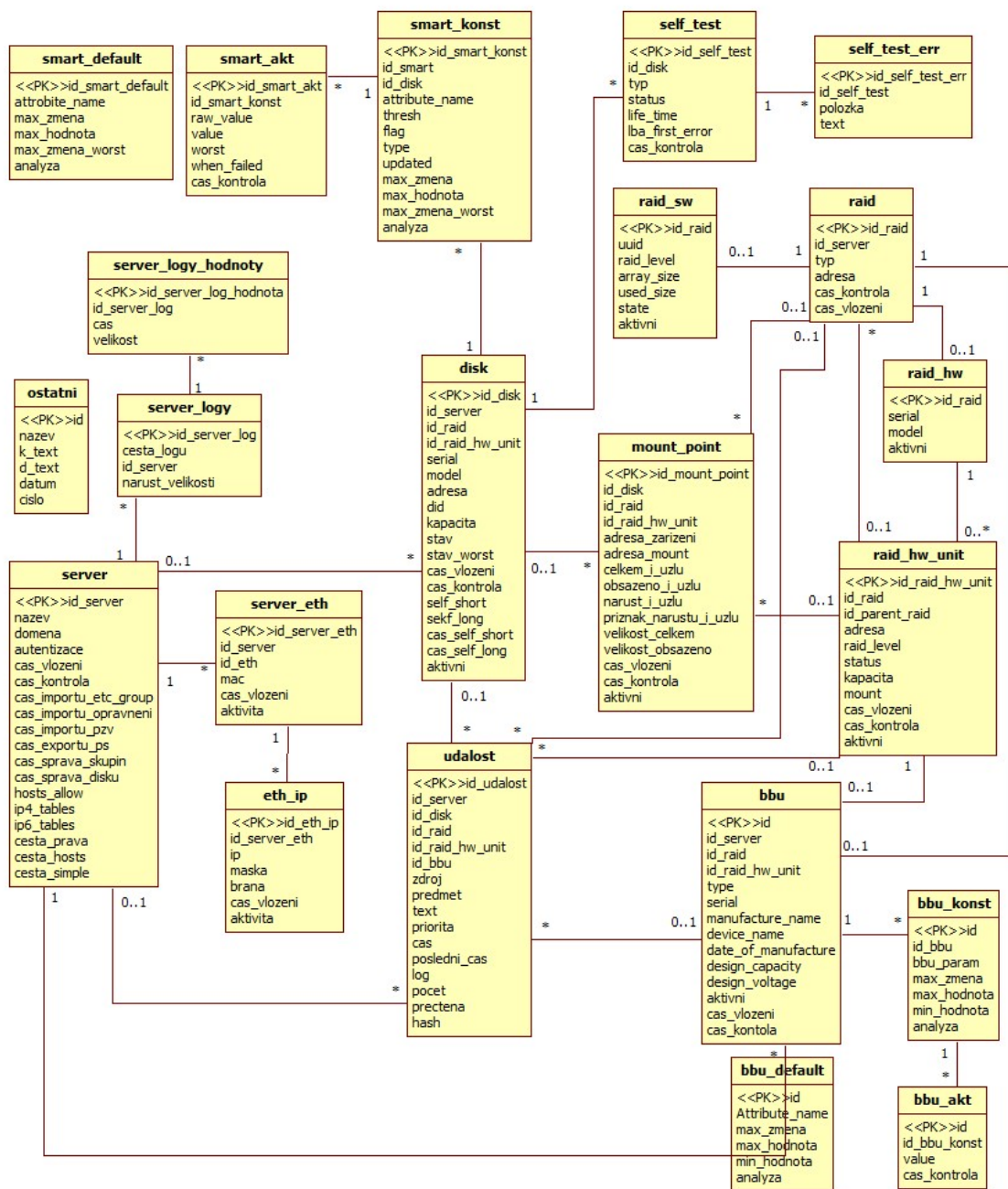
Uložit

Zpět

Obr. 3.13: Nastavenie parametrov sledovania batérie

3.5 Databáza

Na obrázku 3.14 je možné vidieť ER diagram entít, s ktorými pracuje súčasný modul pre monitorovanie serverov a diskov. Model neodráža realitu z niekoľkých dôvodov.



Obr. 3.14: ER diagram súčasnej databázy

Entita **raid** generalizuje entity **raid_sw** a **raid_hw**, pričom **raid_sw** reprezentuje softvérové diskové pole a **raid_hw** RAID radič. HW diskové pole reprezentuje entita **raid_hw_unit**, pričom jej vzťah s entitou **raid_hw** je nepoužívaný systémom.

Fakt, že HW RAID jednotka (**raid_hw_unit**) patrí do radiča (**raid_hw**) je modelovaný pomocou entity **raid**, na ktorú musia mať odkaz.

Entita **bbu_konst** podobne ako entita **smart_konst** modelujú objekty, ktoré zaobalujú nastavenia konkrétneho atribútu. Ich záznamy reprezentujú entity **bbu_akt** a **smart_akt**. Pre nastavenia existujú entity východiskového nastavenia **bbu_default** a **smart_default**,

s ktorými však entity nemajú žiadny vzťah. Väzby je teda nutné modelovať v informačnom systéme porovnávaním názvov atribútov, čo najmä pri SMART technológii môže byť veľmi zradné – SMART technológia je implementovaná výrobcom a názvy sú síce podobné, ale nie rovnaké.

Ako bolo už spomenuté, záznamy z kontrol pre batériu záložnej jednotky a SMART atribútov sú ukladané všetky, čo spôsobuje využívanie neúmerne veľkého priestoru. Tento stav je o to nešťastnejší, že namerané hodnoty sa veľakrát nelíšia od predchádzajúcich.

Kapitola 4

Technológie

Táto kapitola sa venuje technológiám využívaným v práci a dôvodom ich voľby. V jednotlivých podkapitolách nižšie sú zdôvodnené využívané technológie pre webový server a koncové zariadenie, nástroje pre získavanie informácií o úložiskách – diskových a RAID jednotkách. Posledná podkapitola informuje o monitorovaných systémových záznamoch serverov.

4.1 Webový server

Server je výpočtový stroj, na ktorom beží celá aplikačná logika a je zodpovedný za integritu a konzistenciu dát, spracovanie vstupov od užívateľa a vygenerovanie výstupov informačného systému. Server odpovedá na dotazy klienta [12]. V podkapitole sú vymenované technológie, ktoré sú využité na strane servera.

PHP

PHP Hypertext Preprocessor (PHP) je skriptovací jazyk určený pre serverovú časť navrhnutý pre vývoj predovšetkým dynamických webových aplikácií, ktorého začiatky sa datujú do roku 1995. Jeho autorom bol Rasmus Lerdorf a od počiatku tento jazyk prešiel veľkým vývojom. Dnes je to plnohodnotná otvorená platforma pre univerzálne programovanie. Teda ho je možné použiť nie len pre vytváranie webových aplikácií, ale aj pre vývoj desktopových, či konzolových programov. Skript je možné spustiť interpretom alebo webovým serverom a užívateľovi je poslaný až výstup zo skriptu [10].

Hlavným dôvodom výberu jazyka je fakt, že informačný systém KNOTIS je vyvíjaný práve v tomto jazyku. Medzi jeho ďalšie výhody patrí možné spojenie priamo s dokumentom HTML a obľúbenosť – až 82% [25]. Vďaka tomu má jazyk širokú podporu komunity a z toho vyplývajúce veľké množstvo dostupnej funkcionality. Pre vývoj aplikácie je použitá verzia PHP7.

MySQL

MySQL je relačný databázový systém vydaný v roku 1995 spoločnosťou TCX DataKonsult AB, dnes vlastnený spoločnosťou Oracle Corporation už od roku 2009, pod ktorej správou je projekt otvorený komunite. Spoločnosť však ponúka platenú verziu s pridanou funkcionalitou. Projekt si našiel veľkú podporu užívateľov a stal sa jedným z najobľúbenejších. Na svete existuje viac než sto miliónov aktívnych inštalácií a používajú ho aj veľké korporácie ako NASA či Google [10].

MySQL podporuje mnohé operačné systémy [17], ponúka možnosť v rámci jednej databázy pracovať s rôznymi ukladacími mechanizmami (InnoDB, MERGE, MEMORY, ...), z ktorých každý je vhodný na iný typ úloh [10].

MySQL ponúka možnosť fulltextového vyhľadávania a spolu s využitím pamäte cache, do ktorej ukladá dotazy s výsledkami, MySQL ponúka vysoký výkon [10] [17].

Výhody vyplývajúce z odsekov vyššie tvoria len časť argumentov pre zvolenie MySQL ako databázového systému pre prácu. Informačný systém KNOTIS dlhodobo používa MySQL vo verzii 5.5.58-0ubuntu0.14.04.1 a tento systém využíva aj riešenie práce.

4.2 Webový klient

Klient alebo koncové zariadenie slúži užívateľovi ako prístupový bod do informačného systému. Klient získava vstupné dáta od užívateľa, komunikuje so serverom a zobrazuje jeho výstup či odpoveď. Na klientskom zariadení môže byť spustený algoritmus pre manipuláciu s dátami, avšak to by malo byť len k účelu zlepšenia užívateľského rozhrania. Teda zodpovednosť za dáta ostáva stále na strane servera, ktorý sa nemôže spoliehať na klienta, keďže nieje pod jeho kontrolou [12]. Podkapitola predstavuje technológie, ktoré sú použité na strane klienta.

HTML

Jazyk Hyper Text Markup Language, skrátene HTML, vznikol v roku 1990 ako aplikácia jazyka SGML a od tohto počiatku prešiel značným vývojom až k verzii 5.2, pričom momentálne sa pracuje na verzii 5.3 [7]. HTML je štandardizovaný značkovací jazyk určený pre tvorbu webových dokumentov a v práci je využívaná verzia 4.1 najmä kvôli faktu, že systém KNOTIS je implementovaný v HTML 4.01 Transitional.

CSS

Jazyk Cascading Style Sheets (CSS) od roku 1996 prináša možnosť oddelenia mechanizmu formátovania vzhľadu dokumentu buď do samostatnej sekcie, alebo do samostatného dokumentu [4]. V spojení s HTML nám jazyk umožňuje oddelenie štruktúry od vzhľadu, čím sa získava jednoducho čitateľný kód. V práci je použitá verzia CSS 3.

JavaScript

Jazyk označovaný ako klientský JavaScript je objektovo orientovaný a najviac využívaný vo webovom prostredí. Je to skriptovací, slabo typovaný, na prototypoch založený, vysoko úrovňový jazyk, ktorý je spúšťaný najmä vo webovom prehliadači. Je podmnožinou jazyka JavaScript a prináša niektoré rozšírenia, najmä možnosť práce s modelom DOM¹ a prvkami užívateľského rozhrania [26].

JavaScript, pôvodne vyvíjaný spoločnosťou Netscape, je určený pre webové aplikácie. Norma jazyka je stanovená európskou asociáciou výrobcov počítačov ECMA v podobe ECMAScripts, ktorú implementuje väčšina prehliadačov. Posledná stabilná verzia je ECMA-262² [21]. V práci je využívaná ECMAScript 6, ktorá je z väčšiny implementovaná vo všetkých moderných prehliadačoch (Chrome, Firefox, Edge)³.

¹<https://www.w3.org/DOM/>

²<https://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf>

³<http://kangax.github.io/compat-table/es6/>

jQuery

JavaScript knižnica jQuery vznikla v roku 2006 vylepšením knižnice Behavior Johnom Resigom. Dnes je najnovšou verziou jQuery 3.2.0.

Knižnica poskytuje viacúčelovú abstraktnú vrstvu pre bežné webové programovanie, umožňuje efektívnejšiu prácu s modelom DOM, vyhľadávanie v dokumente jazyka HTML, efektívnejšie upravovanie vzhľadu a obsahu modelu DOM, ľahko reagovať na akcie užívateľa a mnoho ďalšieho [3]. V práci je využitá verzia 1.9.2 z dôvodu, že ju využíva a podporuje systém KNOTIS.

Dygraph

Pomocou knižnice Dygraph⁴ je možné vytvárať interaktívne grafy na webovej stránke. Pre riešenie knižnica využíva jazyk JavaScript. Pre grafy je možné definovať viac datasetov, definovať viac osí, interaktívne približovať (vertikálne aj horizontálne) a oddalovať. Hlavným dôvodom zvolenia tejto knižnice je však schopnosť rýchlo a efektívne pracovať s veľkým množstvom dát⁵.

4.3 Technológie pre monitorovanie serverov

Monitorovanie úložísk je možné rôznymi metódami. Jednou je priamy zápis a čítanie sektorov a overovanie, či je hodnota správne zapísaná a či ju je možné prečítať. Toto je veľmi spoľahlivá metóda pre odhalenie chybného disku, pretože nám povie presný počet vadných sektorov. Avšak spadá viac do kategórie testovania ako priebežného monitorovania, pretože metóda zataží disk do takej miery, že ho je nemožno používať v produkčnom prostredí.

Ďalšou možnosťou je sledovanie záťaže, teda počtu zapísaných a prečítaných dát. Aby sme mohli vyvodzovať závery o stave disku, museli by byť k dispozícii mnohé dáta rôznych modelov pre koreláciu či strojové učenie. Tieto dáta však výskumná skupina KNOT k dispozícii nemá. Navyše tieto dáta nevypovedajú nič o aktuálnom stave disku.

Technológie S.M.A.R.T. poskytuje prehľad informácií, ktoré vypovedajú o stave disku. Informácie poskytuje samotný disk a implementácia je v režii výrobcu. Bližšie informácie o technológii poskytuje podkapitola nižšie.

Zabbix

V súčasnosti na trhu existuje niekoľko zaužívaných produktov, či opensource projektov zameraných na monitorovanie serverov, ktoré umožňujú vytvárať, prípadne rovno poskytujú moduly a plugíny pre monitorovanie úložísk. Jedným z nich je Zabbix⁶. Projekt poskytuje užívateľovi moduly, ktoré autonómne monitorujú disky – S.M.A.R.T. monitoring with smartmontools⁷, či Cross-platform SMART monitoring⁸, ktorý umožňuje sledovanie diskov v RAID jednotkách. Problémom je, že mnohé moduly zamerané na S.M.A.R.T. poskytujú iba náhľad na aktuálne hodnoty, prípadne upozornia na problémové atribúty, pričom nie je možná akákoľvek konfigurácia upozorňovania. Avšak väčším problémom je monitorovanie

⁴<http://dygraphs.com>

⁵<http://blog.dygraphs.com/2012/08/html5-charting-tool-performance.html>

⁶<https://www.zabbix.com/>

⁷<https://share.zabbix.com/storage-devices/smartmontools/s-m-a-r-t-monitoring-with-smartmontools>

⁸<https://share.zabbix.com/storage-devices/smartmontools/zabbix-smartmontools>

diskov nachádzajúcich sa v diskovom poli. Skript nedetekuje diskové polia automaticky a je nutné manuálne pridať refazec pre každý disk do skriptu v jazyku python, ktorý zbiera údaje⁹. A tento skript sa musí nachádzať na každom serveri. Údržba týchto serverov by bola nesmierne náročná. Existujú aj iné moduly, žiadny žiaľ nie je schopný automaticky získať SMART údaje o všetkých diskoch, konkrétne o diskoch v diskovom poli 3ware. Tieto disky moduly nie sú schopné na serveri identifikovať.

Obr. 4.1: Cross-platform SMART monitoring. Zdroj: <https://share.zabbix.com/storage-devices/smartmontools/zabbix-smartmontools>

Solarwinds

Produkt tiež ponúka monitorovanie systémových záznamov a mnohé pridružené funkcie. Podobne ako Zabbix, aj Solarwinds agreguje, kategorizuje a prehľadne zobrazuje udalosti na serveroch. Je síce možná filtrácia a ignorovanie záznamov, tu však systém ignoruje záznamy úplne a správca nemá informáciu o skutočnom nefiltrovanom náraste. Pri zle

nastavenom filtri to môže byť osudné. Napríklad, ak filter obsahuje chybu a filtruje záznamy, ktoré by nemal, správca nemá žiadnu možnosť to zistiť.

Nagios

Produkt Nagios je už pomerne dlho na trhu. Ponúka obrovské množstvo rozšírení a jednoduchý systém pre vytváranie vlastných rozšírení. Najväčšou nevýhodou však je predpokladanie statického prostredia. Nie je schopný automaticky odhaliť či rozpoznať nový pripojený server do infraštruktúry.

Vyhodnotenie

Podkapitola prezentuje tri možné riešenia a ukazuje najmä ich nedostatky a dôvody, prečo ich nemožno využiť v systéme KNOTIS. Medzi hlavné nedostatky patrí nemožnosť objaviť všetky disky, nerozpoznávanie nových serverov a nejasné zobrazenie architektúry zapojenia úložísk.

Skupina KNOT je špecifická v tom, že spravuje veľa serverov a pracujú pre ňu nie len študenti postgraduálneho štúdia, ale bežne aj študenti magisterského a bakalárskeho štúdia. Zamestnanci a študenti vykonávajú často výpočtovo náročné experimenty, ktoré môžu generovať veľa udalostí, ktoré sú servermi zaznamenávané. Len pre ilustráciu si môžeme predstaviť experimentovanie a vyhodnocovanie slovníku určeného pre slovníkový útok na službu SSH. Pre každý neúspešný pokus o pripojenie bude v súbore `/var/logs/auth.log` hneď niekoľko nových záznamov. S takýmto scenárom žiadne z uvedených riešení nepočíta a nie je možné odlíšiť nárast experimentálnej a podozrivej aktivity na serveroch.

Z vyššie uvedených dôvodov práca prináša vlastnú implementáciu modulov systému KNOTIS, ktoré zabezpečia monitorovanie úložísk a systémových záznamov.

S.M.A.R.T.

Self-Monitoring, Analysis and Reporting Technology, skrátene SMART, má pôvod v špecifikácii SFF-8035i [6] od skupiny výrobcov diskov (Seagate, IBM, Quantum Western Digital, ...). Technológia v súčasnosti zahŕňa sériu atribútov vypovedajúcich o stave a výkonnosti disku, ktoré sú špecifické pre konkrétny model disku [9], [23]. Väčšina veľkých výrobcov však poskytuje štandardné informácie¹¹.

SMART technológia je zavedená pre rozhrania ATA aj SCSI, avšak oznamovanie informácií sa líši a rozhrania nepodporujú plne túto technológiu [9]. I keď mnoho SMART atribútov bolo štandardizovaných, niektorí výrobcovia diskov nepodporujú všetky a pridávajú vlastné [23].

Eduardo Pinheiro spolu so spoluautormi [20] uvádzajú korelácie medzi SMART atribútmi a diskovými zlyhaniami na veľkom počte zariadení v množine úložísk v spoločnosti Google. K podobným záverom prišiel aj Hughes [13], ktorý koreláciu potvrdil.

Z praktického hľadiska SMART technológia pozostáva zo signalizačného protokolu medzi počítačom a elektromechanickými senzormi disku, ktoré poskytujú informácie, z ktorých je možné vyvodiť stav disku [23]. Ako nástroj pre získavanie výsledkov SMART technológie je v Linuxovom prostredí zaužívaný a veľmi obľúbený program smartmontools. V práci je použitá verzia 6.5.

Aplikácia poskytuje základné informácie o disku, avšak najvýhodnejšou je práve pre možnosť získania hodnôt SMART atribútov, self testov a záznamov o chybách, ak disk

¹¹http://www.drivehealth.com/supported_drives.html

tieto informácie poskytuje. Najdôležitejšie informácie o SMART atribútoch, ktoré aplikácia poskytuje, sú normalizovaná hodnota, threshold hodnota, ktorá ak je dosiahnutá, tak disk už zlyháva, a raw hodnota, čo je nenormalizovaná hodnota atribútu. Hodnoty sú aktualizované stále alebo iba offline testom, ktorý zozbiera informácie a atribúty aktualizuje. Tento zber môže byť automatický, ak ho disk podporuje. Aplikácia tiež poskytuje zoznam posledných chýb, ktoré disk zaznamenal, a informácie o nich, ako typ chyby, stav registrov, príkaz, pri ktorom chyba nastala, a iné. Poslednou dôležitou časťou výstupu sú informácie o self testoch. Najčastejšie využívané sú short test a long test [1]. Testy testujú elektrické a mechanické vlastnosti a kontrolu správnosti čítania¹². Rozdiel medzi nimi je limit doby vykonávania short testu – 2 minúty. Long test nie je obmedzovaný a kontrola správnosti čítania sa robí na väčšom rozsahu adries oproti short testu [1].

mdadm

Mdadm je nástroj pre manipuláciu so softvérovým RAID v linuxovom prostredí. Umožňuje vytváranie, úpravu, rušenie softvérových RAID jednotiek, výpis ich stavov a mnoho iného [16]. V práci je použitá verzia 3.3.

StorCLI

StorCLI je konzolová aplikácia pre manažment RAID radičov rodiny MegaRAID. Poskytuje manažment radiča, batérie zálohovacej jednotky, diskových jednotiek a logických jednotiek [15]. V práci je použitá verzia 1.21.06.

3ware CLI

3ware CLI je konzolová aplikácia pre manažment 3ware RAID radičov rodiny LSI 3ware¹³. Poskytuje manažment radiča, batérie zálohovacej jednotky, diskových jednotiek a logických jednotiek [2]. V práci je použitá verzia 2.00.11.022.

LVM

LVM je otvorený softvér vyvíjaný spoločnosťou Red Hat, ktorý vytvára abstraktnú vrstvu nad fyzickými zväzkami (physical volume). Základ abstrakcie tvorí skupina zväzkov (volume group), do ktorej patria rôzne fyzické zväzky. S LVM Group následne môžeme pracovať a vytvárať z nej logické zväzky, ktoré môžeme následne využívať v operačnom systéme. Technológia umožňuje jednoduchú a rýchlu správu logických zväzkov (logical volume), meniť ich názvy, veľkosti, zrkadliť ich, či vytvárať tzv. snapshot-y [14]. V práci je použitá verzia 2.02.98.

Nástroj ip

Nástroj je súčasťou balíka `net-tools`, ktorý obsahuje konzolové aplikácie pre prácu so sieťovými zariadeniami. Práca využíva nástroj `ip` pre získanie informácií o sieťových rozhraniach a bránach. V práci je použitá verzia 1.60¹⁴.

¹²https://www.thomas-krenn.com/en/wiki/SMART_tests_with_smartctl

¹³<https://www.broadcom.com/support/knowledgebase/1211161499498/operating-systems-supported-for-the-lsi-3ware-and-megaraid-contr>

¹⁴<https://wiki.linuxfoundation.org/networking/net-tools>

Nástroj lsblk

Práca využíva konzolovú aplikáciu lsblk, ktorá je súčasťou balíka `util-linux`. Poskytuje informácie o mapovaní systémových zariadení na body pripojenia a informácie o štruktúre fyzických a logických zväzkov, ak je využitá technológia LVM. V práci je použitá verzia 1.31¹⁵.

Logwatch

Nástroj Logwatch slúži pre získanie unifikovanej správy o tom, čo sa deje na serveri. Pre to využíva rôzne systémové záznamy, ktoré je možné nakonfigurovať a aj poskytnúť skript pre ich spracovanie. Mnohé štandardné však nástroj podporuje. V práci je využitá verzia 7.4.0¹⁶.

Python

Python je objektovo orientovaným skriptovacím jazykom, ktorého interpret je voľne dostupný. V práci je využívaný jeho interpret vo verzii 3.4.2¹⁷ pre konzolovú aplikáciu pre transformáciu dát do formátu JSON.

¹⁵<https://www.kernel.org/pub/linux/utils/util-linux/v2.31/>

¹⁶<https://sourceforge.net/projects/logwatch/files/>

¹⁷<https://www.python.org/downloads/>

Kapitola 5

Návrh riešenia

Kapitola sa venuje návrhu riešenia modulov monitoringu serverov so zameraním na systémové záznamy a úložiská. Keďže zmena informačného systému nie je jednoduchým procesom nie len pre vývojárov, ale aj pre samotných užívateľov, keďže či už vo väčšej, alebo menšej miere sa musia prispôbiť zavádzaným zmenám. Preto by mali ostať zachované dobré zaužívané princípy aktuálneho riešenia a ak je užívateľské rozhranie dobré, malo by byť podobné, keďže užívatelia sú naučení na istý spôsob práce so systémom.

5.1 Užívateľské rozhranie

Práca navrhuje užívateľské rozhranie pre správu a vizualizáciu monitorovaných dát formou wireframe-ov nachádzajúcich sa v prílohách. Práca sa drží zaužívaného konceptu zobrazovania v systéme KNOTIS a preberá hierarchiu stránok, ich štruktúru a rozloženie zo súčasného stavu. To najmä z dôvodu, že zmena užívateľského rozhrania nemusí byť vždy prijímaná dobre a užívatelia si musia zvykať na nové prostredie. Práca prináša novú funkcionálnosť pre správcu pre urýchlenie a uľahčenie práce so systémom a dátami. V jednotlivých podkapitolách je návrh zmeny užívateľského rozhrania oproti súčasnému stavu.

5.1.1 Úložiská

Správcovi sú poskytnuté náhľady na zoznamy jednotlivých typov úložísk a batérií (a kondenzátorov), má možnosť mazať diskové jednotky a polia. Vidí detaily úložísk a batérií, a môže spravovať parametre sledovania.

Pre každý server je viditeľné, aké úložiská sa na ňom nachádzajú, pričom identifikátory úložísk slúžia ako odkaz na detail daného úložiska. Informácie poskytované systémom sú oproti súčasnému riešeniu doplnené o LVM logické zväzky pre body pripojenia, ak je technológia použitá, a o zoznam bodov pripojenia úložísk (viď obr. B.1). Pre server je možné spravovať ethernet rozhrania v samostatnom náhlade (obr. B.2). Obrázky B.3 a B.4 navrhujú formuláre pre pridanie (a zároveň úpravu existujúceho) ethernet rozhrania a príslušnej IP adresy.

Disky

Systém ponúka náhľad zoznamu diskov, viď obr. B.5. Pohľad ponúka možnosti filtrovania a zoradovania podľa jednotlivých stĺpcov a tiež možnosť ukladať a načítavať filtre, ktoré sú často využívané. Tabuľka 5.1 zobrazuje možnosti filtrovania v zozname.

stĺpec	metóda filtrovania
model	model disku obsahuje zadaný podreťazec
sériové číslo	s. číslo disku obsahuje zadaný podreťazec
kapacita	pri zadanej hodnote v GB sa selektujú hodnoty s istou toleranciou, kvôli hodnotám na nižších rádoch, na druhej strane, je možné pred hodnotu zadať znamienko < a > so známym významom
server	podľa vybraného servera z dostupných
raid	podľa vybraného diskového poľa z dostupných
dátum kontroly	filtrovanie diskov, ktoré boli kontrolované v danom dni, v prípade zadania < a > sú zobrazené disky, ktoré boli kontrolované pred, či po dátume
stav	stav obsahuje zadaný podreťazec
bod pripojenia	bod pripojenia existuje a obsahuje zadaný podreťazec
zariadenie	názov zariadenia obsahuje zadaný podreťazec
dátum vloženia	podobne ako dátum kontroly
aktívny	podľa výberu: áno alebo nie
chybný	podľa výberu: áno alebo nie

Tabuľka 5.1: Možnosti filtrovania zoznamu diskov

Detail disku (obr. B.9) zobrazuje základné údaje o disku ako model, sériové číslo, kapacita v ľudske čitateľnej podobe a iné. Ponúka možnosť označiť disk ako chybný, čo uľahčuje evidenciu diskov a nezobrazuje ho naďalej v zozname problémových diskov. Správca vidí aktuálne hodnoty SMART atribútov a nastavenia ich parametrov sledovania, pričom tlačítkom '+' si môže zobrazíť nedávnú históriu zmien hodnôt. Názov atribútu odkazuje na detail histórie, kde je v textovej a grafovej podobe zobrazená história hodnôt. Viď obr. B.12.

Správca nižšie vidí históriu posledných 20 self testov s odkazom na celú históriu (viď obr. B.13). Následne je zobrazená história zaznamenaných SMART chýb.

V nastavení (obr. B.10) má správca popri možnosti nastavenia periódy exekúcie self testov možnosť nastaviť prioritu (zobrazená v udalostiach) a parametre sledovania pre jednotlivé SMART atribúty, a to konkrétne:

- maximálna raw hodnota,
- minimálna raw hodnota,
- maximálna zmena raw hodnoty,
- maximálna relatívna zmena raw hodnoty,
- maximálna zmena worst hodnoty,
- zmena raw hodnoty, po ktorej dosiahnutí je uložený záznam.

Formu sledovania môže správca nastaviť nasledujúcu:

- nesledovať,
- generovať len udalosť v systéme KNOTIS,
- zobrazovať len v zozname problémových diskov,

- generovať udalosť v systéme KNOTIS a zobrazovať v zozname problémových diskoch,
- odosielať e-mail správcovi, generovať udalosť a zobrazovať v problémových diskoch.

Správca má možnosť nastaviť používanie východiskových hodnôt. Po zaškrtnutí daného zaškrťavacieho políčka správca hneď vidí hodnotu, ktorá bude používaná. K dispozícii sú tiež tlačítka pre používanie východiskových hodnôt pre všetky nastavenia daného SMART atribútu a tlačítka pre používanie východiskových hodnôt pre všetky možné nastavenia.

Nastavenie východiskových hodnôt je možné na samostatnej stránke, ktorej štruktúra je veľmi podobná nastavovaniu parametrov sledovania SMART atribútov popísaných v odseku vyššie (obr. B.11). Tu je k dispozícii možnosť, ktorá pre konkrétny atribút nastaví všetkým diskom, aby využívali východiskovú hodnotu. Správca teda nie je nútený to nastavovať u každého disku zvlášť.

Nový koncept východiskových hodnôt umožňuje správcovi nastaviť všeobecné hodnoty platné pre väčšinu diskov a následne korigovať nastavenia pre vybrané (podozrivé) disky.

Obrázok B.8 zobrazuje výpis diskov pre účel zmazania. Systém vymaže disk iba ak nebol kontrolovaný po istú dobu podľa nastavení systému (v súčasnosti 365 dní). Systém tiež vymaže SMART dáta a body pripojenia.

Problémové disky

Systém ponúka zoznam problémových diskov (viď obr. B.6), kde je možné problémové disky filtrovať ako v zozname diskov. Popis chyby prináša detailné informácie, prečo je disk označený za problémový. Označený môže byť z nasledujúcich dôvodov:

- raw hodnota aspoň jedného SMART atribútu je väčšia ako nastavená `max raw` hodnota,
- raw hodnota aspoň jedného SMART atribútu je menšia ako nastavená `min raw` hodnota,
- value hodnota (normalizovaná) aspoň jedného SMART atribútu je menšia ako hodnota prahu (`threshold`),
- `when_failed` aspoň jedného SMART atribútu má akúkoľvek hodnotu ,
- posledný short test je chybný,
- posledný long test je chybný.

V prípade chybného SMART atribútu musí byť úroveň sledovania daného atribútu taká, aby umožňovala zobrazenie v zozname problémových diskov. Pre atribút s chybou je v stĺpci akcia možné vykonať nasledujúce úkony s parametrami sledovania atribútu:

- nastaviť dosiahnutý parameter sledovania na aktuálnu hodnotu, čím správca dosiahne to, že ak sa hodnota zmení, systém ho upozorní,
- nastaviť dosiahnutý parameter sledovania na hodnotu, ktorá je o 10 % vzdialená od aktuálnej hodnoty (max hodnota sa navýši na aktuálnu + 10 %),
- zrušiť sledovanie SMART atribútu,
- generovať len udalosť v budúcnosti,

- nastaviť hodnotu parametru sledovania na NULL (napr. `max raw` a teda hodnota môže rásť do nekonečna),
- nastaviť hodnotu sledovania parametru na východiskovú hodnotu.

Agregácia SMART hodnôt

Práca navrhuje analytický nástroj, ktorý agreguje SMART hodnoty pre vybrané disky (viď obr. B.7). Správca má možnosť si vybrať z dvoch metód agregovania: priemer a medián. Agregácia je možná len pre jeden vybraný SMART atribút, a to z dôvodu rýchlejšieho získania a zobrazenia výsledkov. Správca má opäť možnosť nastavenia agregácie uložiť. Nástroj práca navrhuje z dôvodu poskytnutia odpovedí na nasledujúce otázky:

- Ktorý model má najviac chýb čítania?
- Rastú chyby s vekom jednotlivých modelov?
- Sú väčšie disky rovnakých modelov chybovejšie?

Po analýze otázok sú navrhnuté 3 možnosti, podľa ktorých sú dáta agregované:

- model: dáta sú agregované pre také disky, ktoré majú rovnaký model,
- model-kapacita: dáta sú agregované pre také disky, ktoré majú rovnaký model a rovnakú ľudsky čitateľnú kapacitu (napr. 6T),
- model-rok vloženia: dáta sú agregované pre také disky, ktoré majú rovnaký model a boli pridané v rovnakom roku.

Diskové polia

Návrh náhľadov na zoznam HW a SW diskových polí zobrazujú obr. B.19 a B.21. Správca má možnosť filtrovať a zoradovať výstup podľa rôznych stĺpcov a rôznymi metódami. Viac popisuje tabuľka 5.2. Filtre je možné uložiť, či vybrať z uložených, ktoré sú často využívané. Tabuľka obsahuje odkazy na server a detail poľa.

stĺpec	metóda filtrovania
server	podľa vybraného servera z dostupných
zariadenie	názov zariadenia obsahuje zadaný podreťazec
bod pripojenia	bod pripojenia existuje a obsahuje zadaný podreťazec
level	podľa vybraného z dostupných RAID levelov
kapacita	pri zadanej hodnote v GB sa selektujú hodnoty s istou toleranciou, kvôli hodnotám na nižších rádoch, na druhej strane je možné pred hodnotu zadať znamienko < a > so známym významom
stav	stav obsahuje zadaný podreťazec
dátum kontroly	filtrovanie polí, ktoré boli kontrolované v danom dni, v prípade zadania < a > sú zobrazené polia, ktoré boli kontrolované pred, či po dátume
adresa jednotky/UUID SW poľa	adresa/UUID obsahuje zadaný podreťazec
model (HW RAID)	model obsahuje zadaný podreťazec
dátum vloženia	podobne ako dátum kontroly
aktívny	podľa výberu: áno alebo nie

Tabuľka 5.2: Možnosti filtrovania zoznamu HW a SW diskových polí

Obrázky [B.20](#) a [B.22](#) navrhujú pohľady pre detail SW a HW diskového poľa. Pohľady zobrazujú základné informácie ako adresa či UUID, RAID level, body pripojenia, kapacity v ľudske čitateľnej podobe a iné. Pohľady ponúkajú zoznam diskov nachádzajúcich sa v poli, či v prípade SW poľa aj zoznam HW polí, ak je z takýchto jednotiek SW RAID zložený.

Obrázok [B.23](#) zobrazuje výpis SW a HW diskových polí pre účel zmazania. Systém vymaže z databázy dané pole iba v prípade, že neobsahuje žiadny disk, a nebolo kontrolované po určitú dobu (v súčasnosti je nastavenie systému 182 dní). Ak sa jedná o HW RAID, vymaže aj radič, pokiaľ je neaktívny. V danom prípade tiež zmaže záznamy o kontrole a batériu. Body pripojenia sú tiež vymazané.

Batérie a kondenzátory

Keďže niektoré hardvérové diskové polia v serveroch skupiny KNOT využívajú vyrovnávaciu pamäť a s ňou batériu alebo kondenzátor, systém poskytuje informácie týchto záložných zdrojov energie. Obrázok [B.14](#) navrhuje pohľad zoznamu batérii a kondenzátorov a ich niektorých informácií v prehľadnej tabuľke. Správca má možnosť zoradovať údaje podľa stĺpcov a filtrovať tabuľku. Filtre je možné ukladať a neskôr znovu načítavať. Tabuľka [5.3](#) informuje o možnostiach filtrovania. Príloha [A](#) zobrazuje, ktoré atribúty sú sledované pre batérie a ktoré pre kondenzátory. Hodnoty niektorých atribútov majú textovú formu, napr, či sa batéria nabíja, hodnota môže byť **yes** alebo **no**. V takýchto prípadoch systém upozorňuje, ak sa reťazec zmenil. V týchto prípadoch nemá zmysel nastavovať parametre sledovania.

stĺpec	metóda filtrovania
sériové číslo	s. číslo zariadenia obsahuje zadaný podreťazec
typ	typ zariadenia obsahuje zadaný podreťazec
server	podľa vybraného servera z dostupných
radič	podľa vybraného HW RAID radiča z dostupných
výrobný názov	názov zariadenia obsahuje zadaný podreťazec
názov zariadenia	názov zariadenia obsahuje zadaný podreťazec
dátum kontroly	filtrovanie zariadení, ktoré boli kontrolované v danom dni, v prípade zadania < a > sú zobrazené také, ktoré boli kontrolované pred, či po dátume
dátum vloženia	podobne ako dátum kontroly
technológia	technológia zariadenia obsahuje zadaný podreťazec
aktívny	podľa výberu: áno alebo nie

Tabuľka 5.3: Možnosti filtrovania zoznamu batérií a kondenzátorov

Náhľad na detail batérie (obr. B.15) je v rovnakom koncepte ako detail disku – správca má rovnako zobrazené atribúty batérie s možnosťou rýchleho zobrazenia poslednej či celkovej histórie (obr. B.16). Pre atribúty má správca možnosť nastaviť podobné parametre sledovania (obr. B.17):

- maximálna hodnota,
- minimálna hodnota,
- maximálna zmena hodnoty,
- maximálna relatívna zmena hodnoty,
- priorita – zobrazené v udalostiach,
- zmena, po ktorej dosiahnutí je uložený záznam o kontrole.

Správca môže nastaviť formu upozornenia a nastaviť, aby konkrétne atribúty sledovania boli rovnaké ako východiskové hodnoty. Tie je možné nastaviť formulárom, ktorý zobrazuje obr. B.18. Správca tu môže vynútiť všetkým zariadeniam, aby využívali východiskové nastavenia pre konkrétne atribúty sledovania.

5.1.2 Systémové záznamy

Systém umožňuje definovať súbory systémových záznamov, ktoré majú byť monitorované. Takýto súbor obsahuje na jednom riadku jeden záznam, či udalosť, ktorého súčasťou býva čas. Skúsenosti so servermi vo výskumnej skupine KNOT ukázali, že do súboru môžu zapisovať rôzne aplikácie pod rôznymi formátmi. Doposiaľ to boli maximálne 2 rôzne časové formáty. Obrázok B.27 navrhuje rozhranie pre pridanie, či úpravu existujúcej definície súboru na serveri. Pre daný súbor je teda možné definovať:

- typ,
- cestu k súboru,

- maximálny nárast veľkosti po filtrovaní,
- regulárny výraz pre získanie reťazca obsahujúceho čas a formát času pre python¹,
- 2. regulárny výraz pre získanie reťazca obsahujúceho čas a 2. formát času pre python.

Je možné využívať východiskové hodnoty daného typu, čo sa hodí napr. pri ceste, reg. výrazoch a formátoch času. Obrázok B.28 navrhuje rozhranie pre pridanie, či úpravu typu súboru systémových záznamov a jeho východiskových hodnôt.

Obrázok B.26 zobrazuje návrh výpisu súborov systémových záznamov podľa zadaných kritérií. Pre každý súbor je možné graficky zobrazit celkovú veľkosť a nárast v čase. Je možné vybrať, pre ktoré servery a ktoré typy súborov systémových záznamov má systém zobrazovať. Správca má k dispozícii rôzne odkazy (nastavenia daného súboru, mazanie, pridanie definície, ...) pre rýchly prístup. K dispozícii je tiež prehľad súborov a ich grafy pre samotný server (obr. B.30).

Pravidlá pre systémové záznamy

Správca môže definovať súbor pravidiel platných v konkrétnom čase, či neobmedzene pre dané typy súborov systémových záznamov a pre konkrétne servery (obr. B.25 - vizuálne takmer totožný formulár je aj pre úpravu pravidla). Práca navrhuje dve pravidlá.

Prvým je ignorovanie podľa regulárneho výrazu. To znamená, že riadok (záznam) je ignorovaný, ak vyhovuje reg. výrazu. Ignorovanie znamená, že dátová veľkosť riadku nebude započítaná do nárastu, ktorý keď prekročí stanovenú hodnotu, systém upozorní správcu.

Druhým je hlásenie udalostí podľa regulárneho výrazu. Teda ak riadok vyhovuje výrazu, systém na to upozorní. V upozornení sa nachádza, pre koľko riadkov bolo pravidlo platné s konkrétnymi príkladmi.

5.2 Návrh databázy

Podkapitola popisuje návrh databázy formou ER diagramov v niekoľkých častiach, nakoľko jeden ER diagram by bol príliš veľký a neprehľadný. Samotný obrázok so všetkými entitami a vzťahmi sa nachádza na priloženom nosiči.

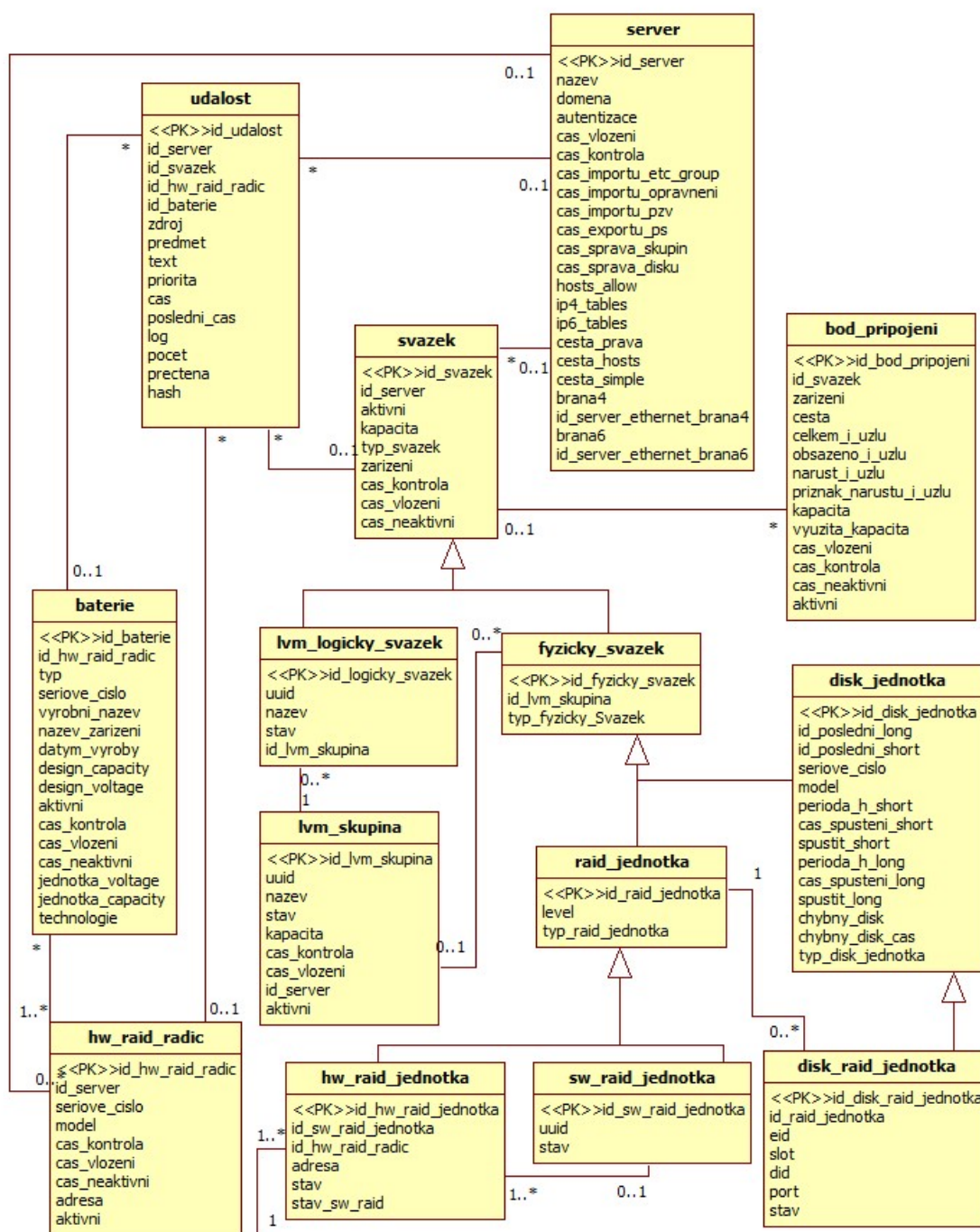
Entita server je prebraná zo súčasného systému a sú do nej pridané atribúty pre IPv4 a IPv6 bránu a referencie na ethernet rozhrania.

Generalizácia je implementovaná tak, že primárny kľúč generalizujúcej a generalizovanej entity je rovnaký a unikátny. Napr. entita **svazek** má nejaký unikátny primárny kľúč a entity, ktoré zovšeobecňuje, majú rovnaký primárny kľúč. Tento vzťah je tranzitívny na ďalšie entity.

Zväzky

Pre správne uloženie informácií o úložiskách je potrebné mať model, ktorý čo najlepšie reprezentuje realitu a spôsob zapojenia jednotlivých fyzických a logických zväzkov. V podkapitole je popísaný abstraktný model (obr. 5.1) popisujúci reálne zapojené diskové polia a diskové jednotky, nad ktorými je možné použiť technológiu LVM, ktorá dovoľuje abstrahovať logický zväzok od fyzického zariadenia. Diagram modeluje možné vytvorenia diskových polia a väzby na entitu **udalost**, ktorá reprezentuje udalosť v systéme.

¹<https://docs.python.org/3/library/datetime.html#strftime-strptime-behavior>



Obr. 5.1: ER diagram úložisk

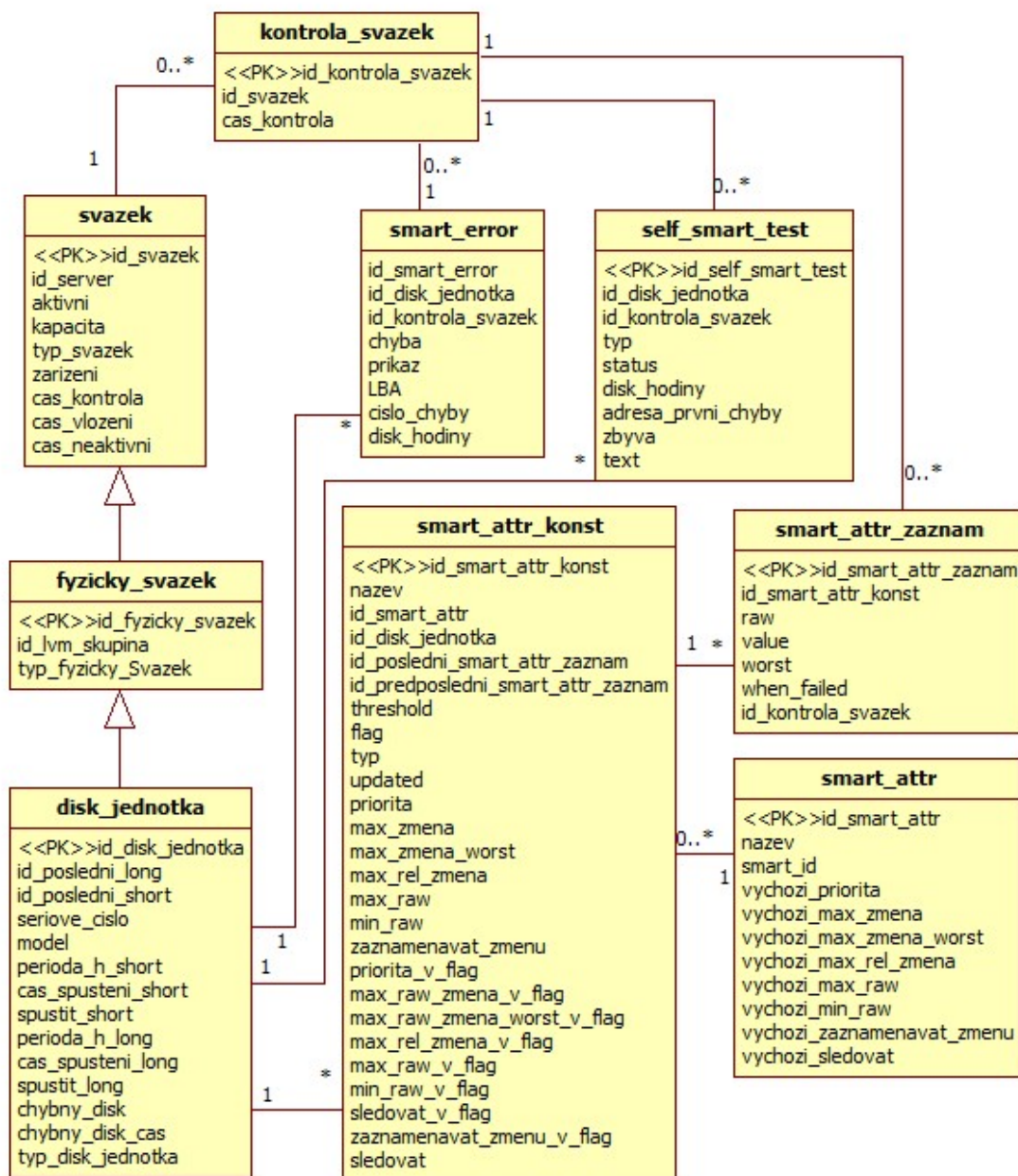
Entita **svazek** reprezentuje úložisko na najvyššej abstraktnej úrovni, tak, ako ho vidí operačný systém. Je o ňom známa kapacita, systémové zariadenie (sda) a čas vloženia. Úložisko sa zväčša nachádza na serveri a môže mať rôzne body pripojenia, o ktorých sa evidujú údaje o i-uzloch, systémovom zariadení, kapacite, ...

Entita `svazek` generalizuje entity `lvm_logicky_svazek` a `fyzicky_svazek`. Fyzický zväzok, čo môže byť diskové pole alebo disk, môže patriť do zväzkovej skupiny technológie LVM (`lvm_skupina`), z ktorej môže byť vytvorených niekoľko logických zväzkov.

Stav, kedy je disk súčasťou diskového poľa, modeluje vzťah medzi `raid_jednotka` a `disk_raid_jednotka` rozširujúcou entitu `disk_jednotka`. `disk_raid_jednotka` rozširuje svoju nadradenú entitu práve o informácie týkajúce sa umiestnenia a stavu jednotky v diskovom poli.

Entita `raid_jednotka` generalizuje entity `sw_raid_jednotka` a `hw_raid_jednotka` reprezentujúce diskové polia na serveri. Keďže softvérové diskové pole môže byť zložené z hardvérových poli, existuje vzťah medzi reprezentujúcimi entitami.

Stav disku



Obr. 5.2: ER diagram pre stav disku

Entity na obr. 5.2 modelujú dáta získateľné aplikáciou smartmontools (SMART technológia). Entity sú navrhnuté tak, aby nie len reprezentovali realitu, ale aj umožnili efektívnu prácu s informáciami.

Disk je pravidelne kontrolovaný, čo je zaznamenávané v entite *kontrola_svazek*. Toto je z dôvodu zachovania informácií o kontrolách, aj keď žiadne SMART dáta nebudú uložené. To môže nastať napríklad pri chybe v procese zberu, spracovania a uloženia dát alebo ak žiadne dáta jednoducho uložiť netreba (napr. sa nič nezmenilo).

SMART technológia ponúka niekoľko typov informácií o stave disku. Názvy entít sa skladajú z anglických slov a ich skratiek z dôvodu ich zaužívanosti a kvôli faktu, že ich tak definuje SMART technológia.

Hodnoty z jednotlivých kontrol sú ukladané do tabuľky `smart_attr_zaznam`. Nastavenia parametrov sledovania jednotlivých SMART atribútov modeluje entita `smart_attr_konst`. Entita tiež zaobahuje statické informácie (typ, flag, ...) o SMART atribúte disku. Východiskové nastavenia sledovania SMART atribútov pre všetky disky modeluje entita `smart_attr`. Atribúty `*_v_flag` sú príznaky, v ktorých je uložená informácia, či pre daný atribút sledovania má byť použitá východisková hodnota, teda ak je nastavený príznak (`max_raw_v_flag`), daný SMART atribút bude vyhodnocovaný proti zodpovedajúcemu `vychozi_max_raw`.

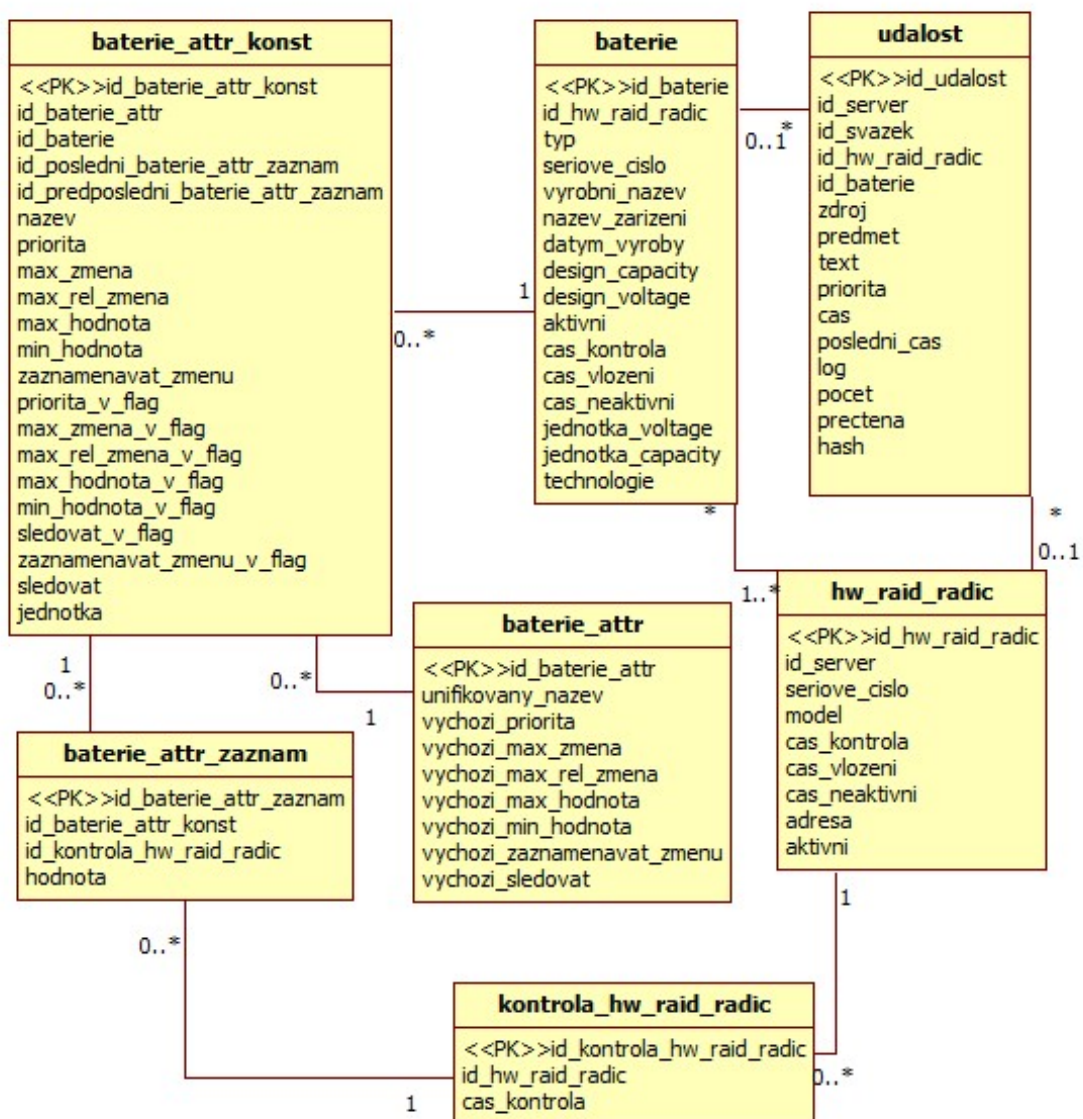
Záznamy o SMART self testoch reprezentuje entita `smart_self_test`.

Disky podporujúce technológiu SMART môžu poskytovať záznam o SMART chybách, ktoré je treba zaznamenávať, keďže môžu indikovať koniec životnosti disku a smartmontools uchováva iba niekoľko posledných záznamov chýb. Entita `smart_error` modeluje záznam o SMART chybe.

Z dôvodov optimalizácie entita `disk` obsahuje odkazy na posledne dva SMART testy a entita `smart_attr_konst` odkazy na posledné dva záznamy. Je to z dôvodu, aby pri implementácii zoznamu problémových diskov bolo možné vytvoriť jednoduchý a rýchlejší SQL príkaz.

Stav batérie

Entity pre zaznamenávanie stavu batérie sú podobné ako pre monitorovanie SMART atribútov disku. Entita `baterie` reprezentuje batériu alebo kondenzátor pre vyrovňavaciu pamäť RAID radiča, ktorý je pravidelne kontrolovaný – kontroly sú zaznamenávané entitou `kontrola_hw_raid_radic` a záznamy entitou `baterie_attr_zaznam` (viď obr. 5.3).

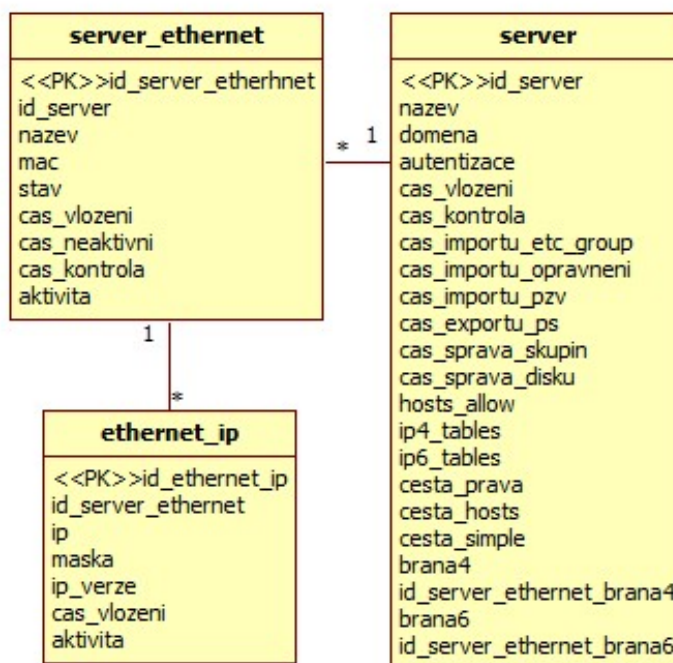


Obr. 5.3: ER diagram pre stav batérie

Záznam hodnoty konkrétneho atribútu reprezentuje entita **baterie_attr_zaznam**. Entita **baterie_attr_konst** má atribúty pre nastavenie sledovania ako je to pri diskoch.

Ethernet rozhrania

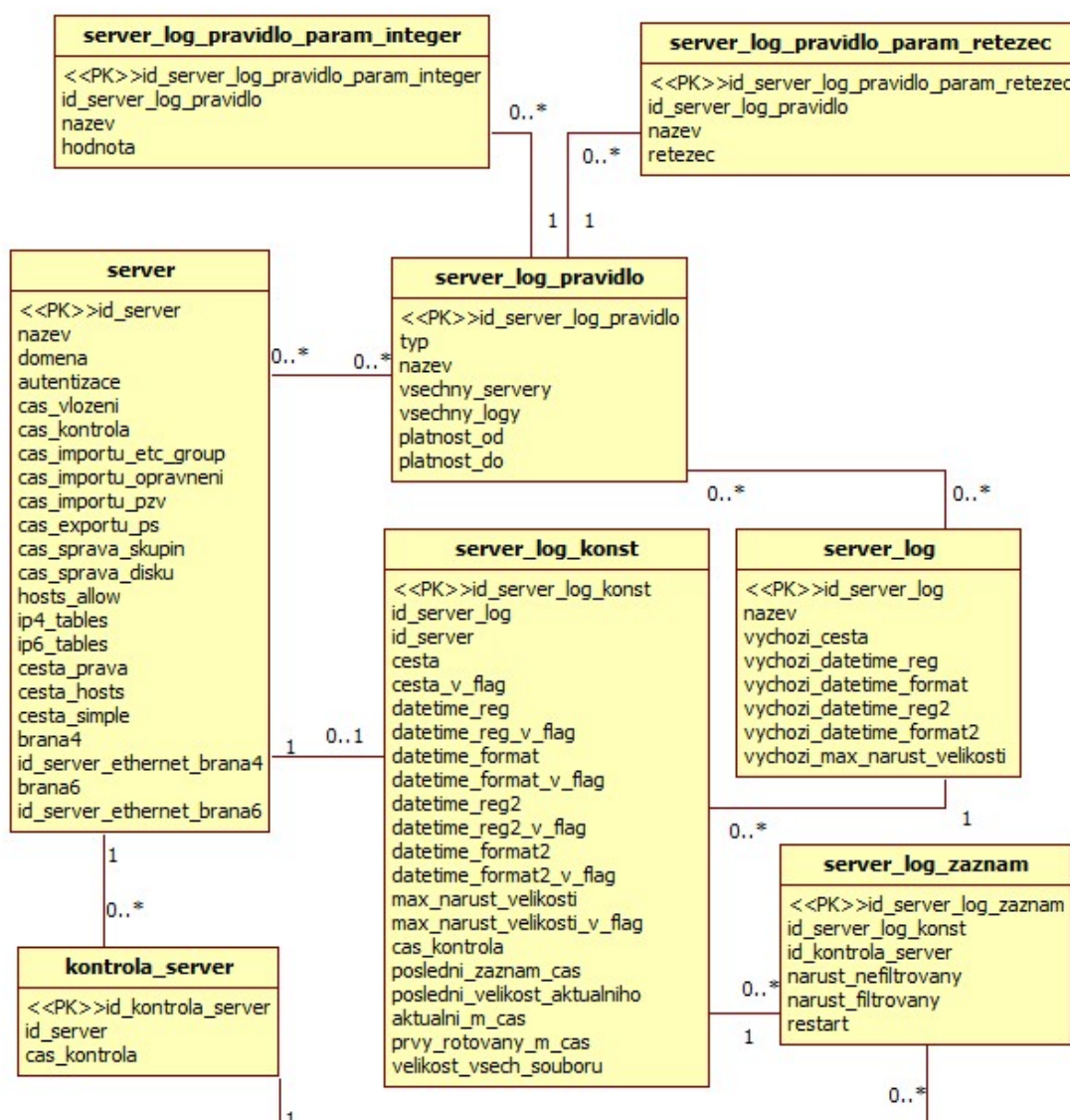
Keďže je potrebné, aby systém mal informácie o ethernet rozhraniach serverov, obr. 5.4 zobrazuje ER diagram entít pre ethernet rozhrania. Server môže mať nakonfigurovaných viac ethernet rozhraní (**server_ethernet**) a každé rozhranie môže mať nakonfigurovaných viac IP adries (**ethernet_ip**).



Obr. 5.4: ER diagram ethernet rozhraní

Systémové záznamy serverov

ER diagram na obr. 5.5 modeluje entity využívané k monitorovaniu systémových záznamov. Entita **server_log_konst** reprezentuje nastavenia konkrétneho súboru záznamov. Servery sú rozdielne a pre každý môže byť typická iná činnosť, ktorá ja na ňom vykonávaná. Vzhľadom k tomu je pre každý server typické množstvo generovaných záznamov. Záznamy o zmenách veľkosti je potrebné zbierať kvôli štatistikám. K tomu účelu je vytvorená entita **server_log_hodnota**. Typ súboru modeluje entita **server_log**, v ktorej sú uložené východiskové hodnoty.



Obr. 5.5: ER diagram systémových záznamov

Na mnohých serveroch skupiny KNOT je zapnuté rotovanie súborov systémových záznamov aby sa uchovávali aj staršie informácie a aby súbory nerástli na veľkosti do nekonečna. Entita **server_log_konst** obsahuje atribúty **aktualni_m_cas** a **prvy_rotovany_m_cas** uchovávajúce čas modifikácie aktuálneho a predchádzajúceho súboru kvôli detekcii rotácie súborov. Tiež je uchovávaný čas posledného záznamu a veľkosť a to z dôvodu, aby sa nekontrolovali už skontrolované záznamy.

Záznamy sú ukladané v entite **server_log_zaznam**, pričom jednotlivé záznamy neobsahujú časovú známku ale odkaz na **kontrola_server**, čím sa šetrí miesto.

Správca v istých okolnostiach potrebuje sledovať konkrétne udalosti a byť na ne upozorňovaný, prípadne ich ignorovať. K sledovaniu slúži entita **server_log_pravidlo**. Entity **server_log_pravidlo_retezec** a **server_log_pravidlo_hodnota** modelujú parametre pravidiel – reťazec a číselná hodnota. Riešenie je škálovateľné a implementované výlučne

na aplikačnej úrovni, na ktorej dovoľuje definovať akékoľvek množstvo pravidiel, pričom v databáze budú odlišené atribútom `typ`. Pravidlá môžu mať akékoľvek množstvo atribútov odlišných názvom.

Pravidlá môžu byť definované pre konkrétne servery a typy súborov systémových záznamov, prípadne pre všetky, na čo slúžia atribúty `vsechny_logy` a `vsechny_servery`. Výhoda takéhoto riešenia je, že ak je v budúcnosti pridaný nový server alebo nová definícia súboru so záznamami, pravidlá automaticky platia aj pre takéto súbory.

5.3 Migrácia dát

Pri prechode na iný informačný systém alebo pri zmene zavedeného je nutné brať ohľad na doposiaľ uložené dáta. Ak je definovaná nová štruktúra, staré dáta musia byť transformované do novej podoby. Keďže práca navrhuje novú schému uloženia dát, táto podkapitola popisuje, ako budú staré dáta prevedené do nových entít.

Pri prenášaní informácie o čase kontroly sa vytvorí väzba na aktuálne existujúci objekt v entite `kontrola_svazek` a `kontrola_hw_raid_radic`, prípadne sa vytvorí nový.

Migrácia diskových polí

Entity `raid_sw` a `raid_hw_unit`, ktoré sú generalizované entitou `raid` sú migrované do entít `hw_raid_jednotka` a `sw_raid_jednotka` a ich generalizujúcich entít v niekoľkých krokoch.

Najprv sa skontroluje, či existuje záznam s rovnakým sériovým číslom (HW RAID) alebo UUID (SW RAID) a ak áno, pokračuje sa ďalším objektom. Ak záznam neexistuje, riadok je pridaný do tabuľky `hw_raid_jednotka` (alebo `sw_raid_jednotka`) a do príslušných entít, ktoré ich generalizujú (poradie je od top-level entity), a nastaví sa atribúty `typ_*`. Atribúty sú prenesené nasledujúco.

Pre entitu `sw_raid` nie je migrovaný atribút `used_size`, keďže využívaná kapacita sa viaže k bodom pripojenia a k jednotlivým partíciám. Popis migrácie ostatných atribútov je v tabuľke 5.4.

atribút entity <code>raid_sw</code>	do entity	do atribútu
<code>uuid</code>	<code>sw_raid_jednotka</code>	<code>uuid</code>
<code>raid_level</code>	<code>raid_jednotka</code>	<code>level</code>
<code>array_size</code>	<code>svazek</code>	<code>kapacita</code>
<code>state</code>	<code>sw_raid_jednotka</code>	<code>stav</code>
<code>aktivni</code>	<code>svazek</code>	<code>aktivni</code>

Tabuľka 5.4: Migrácia entity `raid_sw`

Entita `raid_hw_unit` reprezentuje HW RAID jednotku a spôsob migrovania sa nachádza v tabuľke 5.5.

atribút entity raid_hw_unit	do entity	do atribútu
adresa	hw RAID jednotka	adresa
raid_level	raid_jednotka	level
status	hw RAID jednotka	stav
kapacita	svazek	kapacita
mount	svazek	zarizeni
cas_vlozeni	svazek	cas_vlozeni
cas_kontrola	kontrola_svazek	cas_kontrola
aktivni	svazek	aktivni

Tabuľka 5.5: Migrácia entity **raid_hw_unit**

Entita **raid_hw** modeluje radič HW diskového poľa a migrácia je popísaná v tabuľke 5.6.

atribút entity raid_hw	do entity	do atribútu
serial	hw RAID radič	seriove_cislo
model	hw RAID radič	model
aktivni	hw RAID radič	aktivni

Tabuľka 5.6: Migrácia entity **raid_hw**

Entita **raid** generalizuje entity **raid_sw** (SW diskové pole) a **raid_hw** (HW radič diskového poľa) (viac v tabuľke 5.7).

atribút entity raid	ak generalizuje	do entity	do atribútu
adresa	raid_sw	svazek	zarizeni
	raid_hw	hw RAID radič	adresa
cas_kontrola	raid_sw	kontrola_svazek	cas_kontrola
	raid_hw	kontrola_hw RAID radič	cas_kontrola
cas_vlozeni	raid_sw	svazek	cas_vlozeni
	raid_hw	hw RAID radič	cas_vlozeni

Tabuľka 5.7: Migrácia entity **raid**

Migrácia batérií

Práca navrhuje entitu **baterie**, ktorá reprezentuje aj iný typ úložiska energie – kondenzátor. Súčasný stav nie je úplne tomuto scenáru prispôsobený. Ak atribút **type** pôvodnej entity je CV, tak **jednotka_capacity** je nastavený na J a **jednotka_voltage** je prázdny. V opačnom prípade je **jednotka_capacity** nastavený na mAh a **jednotka_voltage** na mV. Migráciu entity **bbu** do entity **baterie** popisuje tabuľka 5.8.

atribút entity bbu	do atribútu entity baterie
type	typ
serial	seriove_cislo
manufacture_name	vyrobni_nazev
device_name	nazev_zarizeni
date_of_manufacture	datum_vyroby
design_capacity	design_capacity
design_voltage	design_voltage
aktivni	aktivni
cas_vlozeni	cas_vlozeni
cas_kontrola	cas_kontrola (entita kontrola_hw_raid_adapter)

Tabuľka 5.8: Migrácia entity bbu

Tabuľky 5.9 a 5.10 popisujú migráciu entít **bbu_konst** a **bbu_akt** do entít v novej schéme **baterie_attr_konst** a **baterie_attr_zaznam**. Nový záznam sa vytvorí iba vtedy, ak sa atribút hodnota líši od časovo predchádzajúceho záznamu.

atribút entity bbu_konst	do atribútu entity baterie_attr_konst
bbu_param	–
max_zmena	max_zmena
max_hodnota	max_hodnota
min_hodnota	min_hodnota
analiza	sledovat (0->1 1->10)

Tabuľka 5.9: Migrácia entity bbu_konst

atribút entity bbu_akt	do atribútu entity baterie_attr_zaznam
value	hodnota

Tabuľka 5.10: Migrácia entity bbu_akt

Tabuľka 5.11 popisuje spôsob migrovanie entity **bbu_default** do **baterie_attr**.

atribút entity bbu_default	do atribútu entity baterie_attr
id	nazev_unifikovany
attribute_name	nazev
max_zmena	default_max_zmena
max_hodnota	default_max_hodnota
min_hodnota	default_min_hodnota
analiza	sledovat (0->1 1->10)

Tabuľka 5.11: Migrácia entity bbu_default

Migrácia diskov

V starej schéme je disk reprezentovaný entitou **disk** nezávisle od toho, či sa nachádza v diskovom poli alebo nie. Práca navrhuje entitu **disk_raid_jednotka**, ktorá rozširuje entitu **disk_jednotka**, ak sa nachádza v diskovom poli. Atribút **stav_worst** systém nepoužíva a nie je migrovaný. Jednoduchú migráciu niektorých atribútov popisuje tabuľka 5.12.

atribút entity disk	do entity	do atribútu
serial	disk_jednotka	seriove_cislo
model	disk_jednotka	model
did	disk_raid_jednotka	did
kapacita	svazek	kapacita
stav	disk_raid_jednotka	stav
cas_vlozeni	svazek	cas_vlozeni
cas_kontrola	svazek	cas_kontrola
self_short	disk_jednotka	perioda_h_short
self_long	disk_jednotka	perioda_h_long
cas_self_short	disk_jednotka	cas_spusteni_short
cas_self_long	disk_jednotka	cas_spusteni_long
aktivni	hw_raid_radic	aktivni

Tabuľka 5.12: Migrácia entity **disk**

Ak sa disk nachádza v diskovom poli (cudzí kľúč `id_raid` má nejakú hodnotu), tak sa vytvorí patričná väzba na entitu **raid_jednotka**. Atribút **adresa** obsahuje refazec, ktorým je možné identifikovať disk v rámci diskového pola. Migrácia je nasledujúca:

- ak je adresa v tvare **X:Y** (X a Y sú čísla), hodnota X je prenesená do **eid** a hodnota Y do **slot**,
- ak je adresa v tvare **/cX/pY** (X a Y sú čísla), hodnota Y je prenesená do **port**.

Entita **self_test** reprezentuje záznamy self testov a entita **self_test_err** chybové hlášky testov (atribút **polozka** = 2) a chyby spracovania výsledkov (atribút **polozka** = 1). K chybám spracovania však dochádza ak test nemá jeden z očakávaných názvov. Migráciu entít do entity **smart_self_test** popisuje tabuľka 5.13.

atribút entity self_test	do atribútu entity smart_self_test
typ	typ
status	status (ak pôvodná hodnota je 1, tak 1, v opačnom prípade 4)
lba_first_error	adresa_prvni_chyby
life_time	disk_hodiny
cas_kontrola	cas_kontrola (entita kontrola_svazek)
text (entita self_test_err)	text

Tabuľka 5.13: Migrácia entity **self_test**

Informácie o SMART atribútoch sú uložené v entitách **smart_konst**, ktorá obsahuje statické informácie o atribútoch diskov, **smart_akt**, ktorá obsahuje záznamy atribútov a **smart_default**, ktorá obsahuje štandardné nastavenia pre atribúty. Jednoduchú migráciu popisujú tabuľky 5.14 – 5.16.

atribút entity smart_konst	do atribútu entity smart_attr_konst
tresh	threshold
flag	flag
type	typ
updated	updated
max_zmena	max_zmena
max_hodnota	max_hodnota
max_zmena_worst	max_zmena_worst
analiza	sledovat (0->0, 1->10)

Tabuľka 5.14: Migrácia entity smart_konst

Novým atribútom sledovania ako max_rel_zmena sa nastaví hodnoty na NULL.

Tabuľka 5.15 popisuje migráciu atribútov entity smart_akt. Záznam je migrovaný iba ak sa hodnota jedného z atribútov raw_value, value, worst a when_failed líši od časovo predchádzajúcej hodnoty.

atribút entity smart_akt	do atribútu entity smart_attr_zaznam
raw_value	raw
value	hodnota
worst	worst
when_failed	kdy_selhalo
cas_kontrola	cas_kontrola (entita kontrola_svazek)

Tabuľka 5.15: Migrácia entity smart_akt

Počas migrovania entity smart_konst je priebežne migrovaná entita smart_default s príslušným attribute_name. Nová schéma však predpokladá unikátne všeobecné nastavenia pre daný SMART atribút o konkrétnom SMART ID. V pôvodnom riešení mohla nastať situácia, keď existovalo viac východiskových nastavení pre rovnaké SMART ID. V práci bude zmigrované prvé nastavenie a ostatné s rovnakým SMART ID už nie. Viac v tabuľke 5.16.

atribút entity smart_default	do atribútu entity smart_attr
attribute_name	nazev
max_zmena	vychozi_max_zmena
max_hodnota	vychozi_max_hodnota
max_zmena_worst	vychozi_max_zmena_worst
analiza	vychozi_sledovat (0->0, 1->10)

Tabuľka 5.16: Migrácia entity smart_default

Migrácia bodov pripojenia

Migrácia entity mount_point musí prebehnúť, ak sú zmigrované ostatné entity reprezentujúce nejaké úložisko, aby bolo možné vkladať správne hodnoty do cudzích kľúčov – väzba na entitu svazek.

Entita bod_pripojeni je veľmi podobná entite v starej schéme a prakticky sa líši názvami atribútov, čo detailne popisuje tabuľka 5.17.

atribút entity mount_point	atribútu entity bod_pripojeni
adresa_zarizeni	zarizeni
adresa_mount	cesta
celkem_i_uzlu	celkem_i_uzlu
obsazeno_i_uzlu	obsazeno_i_uzlu
narust_i_uzlu	narust_i_uzlu
priznak_narustu_i_uzlu	priznak_narustu_i_uzlu
velikost_celkem	kapacita
velikost_obsazeno	vyuzita_kapacita
cas_vlozeni	cas_vlozeni
cas_kontrola	cas_kontrola
aktivni	aktivni

Tabuľka 5.17: Migrácia entity mount_point

Migrácia systémových záznamov

Novej entite `server_log_konst` pribudlo iba zopár atribútov oproti entite v súčasnom riešení. Migráciu aktuálnych dát popisuje tabuľka 5.18. Entita `server_log_zaznam` obsahujúca záznamy nárastov migrovaná nie je z toho dôvodu, že práca prináša nový koncept práce s veľkosťami súborov systémových záznamov.

atribút entity server_logy	atribútu entity server_log_konst
cesta_logu	cesta
narust_velikosti	max_narust_velikost (vynásobené č. 1000: KB -> B)

Tabuľka 5.18: Migrácia entity server_logy

Regulárne výrazy pre získanie reťazca obsahujúceho čas záznamu a časový formát sú vyhodnotené podľa nasledujúceho kľúča:

- ak sa v `cesta` nachádza `auth.log`, `kern.log` alebo `syslog`, tak regulárny výraz je `\w{3}\s+\d{1,2}\s+\d{2}:\d{2}:\d{2}` a formát času `%b %d %H:%M:%S`
- ak sa v `cesta` nachádza `apache2/access.log`, tak regulárny výraz je `\d{2}/\w{3}/\d{4}:\d{2}:\d{2}:\d{2}` a formát času `%b %d %H:%M:%S.%f %Y`,
- ak sa v `cesta` nachádza `apache2/error.log`, tak prvý regulárny výraz je `\w{3}\s+\d{2}\s+\d{2}:\d{2}:\d{2}\d+\s+\w{4}` a prvý formát času `%b %d %H:%M:%S.%f %Y`, druhý regulárny výraz je `\d{4}-\d{2}-\d{2}\s+\d{2}:\d{2}:\d{2}\d+` a druhý formát času je `%Y-%m-%d %H:%M:%S.%f`.

Počas migrácie sú priebežne vytvárané záznamy entity `server_log`, a to tak, že napr. pre nastavenia `/var/log/auth.log` je vytvorený riadok z danej hodnoty `narust_velikosti`, `cesta` a vyššie spomínaných regulárnych výrazov a formátov časov.

Migrácia ethernet rozhraní

Entity pre ethernet rozhrania `server_ethernet` a `ethernet_ip` neprinášajú veľké zmeny okrem možnosti uloženia dátumu, kedy rozhranie či adresa IP prestali byť aktívne na ser-

veri, verzie internetového protokolu adres a povolenie textových názvov rozhraní. Migráciu popisujú tabuľky 5.19 a 5.20.

atribút entity server_eth	atribútu entity server_ethernet
id_eth	nazev
mac	mac
cas_vlozeni	cas_vlozeni
aktivita	aktivni

Tabuľka 5.19: Migrácia entity **server_eth**

atribút entity eth_ip	atribútu entity ethernet_ip
ip	ip
maska	maska
brana	brana
cas_vlozeni	cas_vlozeni
aktivita	aktivni

Tabuľka 5.20: Migrácia entity **server_eth**

Atribút **id_eth** obsahuje index rozhrania pomenovaného ethX (X je index). Migrovaný je tak, že sa číslo pripojí k reťazcu **eth**. Počas migrácie sa tiež nastaví atribút **ip_verze** na hodnotu 4, ak sa v adrese IP nachádza znak bodky, a na hodnotu 6, ak sa v adrese nachádza znak dvojbodky.

5.4 Zber a spracovanie dát pre úložiská

Podkapitola popisuje návrh získania dát pre monitorovanie úložísk, ich transformáciu do formátu JSON a následné spracovanie na serveri. Keďže je možné využiť informácie o rozhraniach pre overenie požiadavku, dáta pre monitoring úložísk sú zbierané rovnakým skriptom a spracovávané naraz na serveri. Práca navrhuje skript, ktorý bude vykonávaný automaticky pomocou cron úlohy raz za deň a jednu konzolovú aplikáciu pre transformáciu dát, ktorá extrahuje potrebné informácie zo vstupu a dáta vypíše vo formáte JSON. Jej úlohou je tiež odfiltrovanie nepotrebných dát od potrebných a tým zmenšenie záťaže na sieťovú infraštruktúru. Skript získa a odošle informácie o úložiskách a ethernet rozhraniach na server. Skript komunikuje so systémom KNOTIS, ktorému odošle zozbierané dáta protokolom HTTP a prijme odpoveď, čo sú rôzne príkazy na vykonanie.

Aplikácia pre transformáciu dát obsahuje modul s definovanými regulárnymi výrazmi a reťazcami, ktoré jej umožňujú extrahovať informácie zo vstupu. Všetky regulárne výrazy sú case insensitive. Aplikácii je definovaný súbor, z ktorého číta doposiaľ uložené dáta vo formáte JSON. Keďže aplikácia bude volaná viackrát, tak súbor vždy najprv načíta, spracuje vstup a všetky dáta zapíše do definovaného súboru. Jednotlivé podkapitoly popisujú, aké a ako budú informácie uložené v JSON.

5.4.1 Zber a transformácia dát pre úložiská

Skript najprv získa hostname a doménu aplikáciou **hostname** a aplikácia pre transformáciu dát informácie uloží do **/server/hostname**, **/server/domain** a autentizačný reťazec, ktorý je uložený v premennej, do **/server/auth**.

Stavové informácie diskov

V prostredí výskumnej skupiny KNOT sú disky a HW diskové polia vedené pod systémovými zariadeniami na adresách `/dev/sd[a-z]`. Pre disky, ktoré sa nachádzajú v diskovom poli, aplikácia umožňuje získať dáta prepínačom `-d` pre polia rodín MegaRaid a 3ware. Získanie a spracovanie SMART informácií prebieha v niekoľkých krokoch, pričom pri každom bude výstup zo `smartmontools` predaný aplikácií pre transformáciu dát (ďalej len transformačná aplikácia) spustenej s parametrami: systémové zariadenie a reťazec, ktorý špecifikuje zariadenie pre `smartmontools` (v prípade disku v RAID napr. `-d megaraid,31`). Transformačná aplikácia spracuje vstup regulárnymi výrazmi definovanými ako konštanty.

Skript získa všetky informácie nástrojom `smartmontools` a výstup predá transformačnej aplikácii, ktorá v niekoľkých krokoch transformuje dáta. Tabuľka 5.21 popisuje objekt, ktorú aplikácia vytvorí a uloží do pola `/disks`. Dáta sú ukladané pod krátkymi kľúčmi, prípadne ako N-tice pre úsporu dát.

klúč	popis
0	názov zariadenia (sda)
1	model
2	sériové číslo
3	ID logickej jednotky (v prípade, že sa jedná o RAID)
4	kapacita
5	informácia o podpore SMART
6	status posledného SELF testu
7	podpora záznamov SELF testov
99	reťazec identifikujúci zariadenie pre aplikáciu tab. <code>smartmontools</code>
t	pole SELF testov, viac tab. 5.23
e	pole SMART chýb, viac tab. 5.24
a	pole atribútov, viac tab. 5.22

Tabuľka 5.21: Objekt JSON disku

N	popis
0	SMART ID
1	názov atribútu
2	flag
3	normalizovaná hodnota (VALUE)
4	najhoršia hodnota (WORST)
5	prah (THRESH)
6	typ atribútu (TYPE)
7	kedy je aktualizovaný (UPDATED)
8	kedy zlyhal (WHEN_FAILED)
9	RAW hodnota

Tabuľka 5.22: JSON 10-tica SMART atribútu

N	popis
0	číslo testu v zozname testov
1	názov testu
2	status
3	koľko % ostáva
4	hodiny disku (lifetime)
5	adresa prvej chyby (LBA)

Tabuľka 5.23: JSON šestica SELF testu

N	popis
0	číslo chyby
1	hodiny disku (lifetime)
2	chyba
3	adresa (LBA)
4	príkaz, pri kt. chyba vznikla (Command)

Tabuľka 5.24: JSON päťica SMART chyby

HW diskové polia

Informácie týkajúce sa hardvérového diskového pola sú získané aplikáciou `storcli`. Skript najprv zistí informácie o všetkých radičoch a ich RAID jednotkách. Transformačná aplikácia vytvorí objekt JSON 5.25 a uloží ho do pola `/hwRaids` v JSON. Následne pre jednotlivé RAID jednotky extrahuje a uloží informácie popísané v tabuľke 5.26.

klúč	popis
serial	sériové číslo RAID radiča
addr	adresa (c0)
units	pole RAID jednotiek pod daným radičom, viac tab. 5.26
model	model RAID radiča
bbu	batéria alebo kondenzátor

Tabuľka 5.25: Objekt JSON radiča

klúč	popis
level	RAID level
state	stav jednotky
capacity	kapacita jednotky
serial	SCSI NAA Id potrebné pre určenie zariadenia (sda,sdb, ...)
capacityUnit	jednotka kapacity
addr	adresa jednotky
disks	pole diskov tvoriace diskové pole, viac tab. 5.27

Tabuľka 5.26: Objekt JSON HW diskového pola

Ak sa jedná o radič rodiny MegaRaid, tak transformačná aplikácia pre jednotlivé disky ukladá `SLOT`, `EID`, `DID`, `STATE` (stav disku) a sériové číslo pre spárovanie so SMART informáciami. Ak sa jedná o `3WARE`, ukladá sa sériové číslo a port. Viď tab. 5.27.

klúč	popis
e	EID
sl	SLOT
d	DID
s	stav (STATE)
se	sériové číslo
p	port

Tabuľka 5.27: Objekt JSON disku v diskovom poli

Skupina KNOT využíva cache jednotky iba v diskových poliach MegaRaid. Informácie o batérii sa získajú s využitím príkazu `/c[X]/bbu show all` a o kondenzátore `/c[X]/cv show all`, kde X je index diskového radiča. Výstupy sú si podobné – líšia sa jednotkami a inými maličkostami. Transformačná aplikácia extrahuje jednotlivé informácie a uloží ich do objektu JSON do `bbu` daného RAID radiča. Viď tabuľka 5.28. Ostatné atribúty definované podľa A sú uložené ako štvorice, ktoré popisuje tabuľka 5.29.

klúč	zariadenie	popis
designCapacity	oboje	kapacita
capacityUnit	oboje	jednotka kapacity
designVoltage	batéria	napätie
voltageUnit	batéria	jednotka napätia
manufacturedDate	oboje	dátum výroby
serial	oboje	sériové číslo
manufacturedName	oboje	výrobný názov
deviceName	oboje	názov zariadenia
technology	batéria	technológia
properties	oboje	poľe atribútov zariadenia, viac tab. 5.29
type	oboje	type: cv (kondenzátor) alebo bbu (batéria)

Tabuľka 5.28: Objekt JSON batérie a kondenzátora na radiči RAID

N	popis
0	unifikovaný názov – jedinečný názov atribútu, ktorý sa skladá zo samotného názvu a názvu sekcie
1	názov
2	hodnota
3	jednotka

Tabuľka 5.29: JSON štvorica pre atribúty batérie a kondenzátoru

Pre HW diskové polia je potrebné identifikovať názov zariadenia, pod ktorým v rámci operačného systému existuje. K tomuto účelu poslúžia informácie, ktoré operačný systém ukladá do `/dev/disk/by-id`, kde sa nachádzajú zariadenia a reťazce začínajúce `wwn-0x`. Takéto zariadenia môžu byť práve diskové polia. Transformačná aplikácia vytvorí dvojicu číslo za `wwn-0x` a názov zariadenia a uloží ich do poľa `/disksById`.

SW diskové polia

Skupina KNOT na serveroch využíva pre spravovanie softvérových diskových polí aplikáciu `mdadm`. Operačný systém pozná tieto úložiská pod zariadeniami `md` a `md[0-9]+`. Aplikáciou `mdadm` je možné zistiť informácie o jednotlivých poliach. Tieto dáta sú predané transformačnej aplikácii, ktorá vytvorí objekt popísaný v tabuľke 5.30. Takéto objekty ukladá do poľa `/swRaids`.

klúč	popis
<code>state</code>	stav poľa (<code>state</code>)
<code>uuid</code>	UUID – jedinečný identifikátor
<code>device</code>	názov zariadenia (<code>md0</code>)
<code>capacity</code>	kapacita v KiB
<code>level</code>	RAID level
<code>failedDevices</code>	počet vadných zariadení
<code>units</code>	pole pre jednotky, viď tab. 5.31

Tabuľka 5.30: Objekt JSON pre SW RAID

N	popis
0	názov zariadenia (<code>sda</code>)
1	stav jednotky

Tabuľka 5.31: Dvojica JSON pre jednotku v SW RAID

Body pripojenia

Aplikáciou `df` skript získa informácie o bodoch pripojenia a transformačná aplikácia extrahuje informácie o systémových zariadeniach, vytvorí objekt JSON (tab. 5.32) a uloží ho do poľa `/mountPoints`.

klúč	popis
<code>usedInodes</code>	počet využitých i-uzlov
<code>inodes</code>	celkový počet i-uzlov
<code>capacity</code>	celková kapacita
<code>capacityUnit</code>	jednotka celkovej kapacity
<code>usedCapacity</code>	využitá kapacita
<code>usedCapacityUnit</code>	jednotka využitej kapacity
<code>device</code>	názov zariadenia
<code>path</code>	cesta bodu pripojenia

Tabuľka 5.32: Objekt JSON reprezentujúci bod pripojenia

LVM

Aplikácia `lsblk` poskytuje výstup, z ktorého je možné zistiť, z akých zariadení je LVM zväzok vytvorený. Výstup je užitočný aj pre body pripojenia, pretože môžeme opäť zistiť zariadenie, z ktorého je partícia vytvorená. Aplikácia `lsblk` poskytuje výstup vo formáte JSON, ktorý je uložený pod klúč `/lsblk` transformačnou aplikáciou.

Skript zistí informácie o vytvorených skupinách LVM aplikáciou `vgdisplay` a transformačná aplikácia pre každú skupinu vytvorí objekt JSON (tab. 5.33) a uloží do poľa `/lvmGroups`. Dáta o zväzkoch LVM skript získa aplikáciou `lvdisplay`, transformačná aplikácia pre každý vytvorí objekt JSON ako definuje tabuľka 5.34 a uloží do poľa JSON `/lvmLogicalVolumes`.

klúč	popis
state	stav skupiny
size	veľkosť (spolu s jednotkami)
uuid	identifikátor skupiny
name	názov skupiny

Tabuľka 5.33: Objekt JSON reprezentujúci skupinu LVM

klúč	popis
state	stav zväzku
size	veľkosť (spolu s jednotkami)
uuid	identifikátor skupiny
name	názov skupiny
path	cesta
group	názov skupiny

Tabuľka 5.34: Objekt JSON reprezentujúci zväzok LVM

Aplikácia `lvdisplay` poskytuje cestu k jednotlivým zväzkom, tá však nie je cesta zariadenia, pod ktorým operačný systém vidí zväzky. To je možné zistiť aplikáciou `dmsetup`, ktorá pre jednotlivé cesty zväzkov LVM a zariadení poskytuje UUID. Zväzok LVM má toto UUID rovnaké ako zariadenie, pod ktorým operačný systém vidí daný zväzok. Transformačná aplikácia vytvorí dvojicu cesta (zariadenia alebo cesta zväzku LVM) a UUID. Dvojicu pre zväzok LVM pridá do poľa `/dmLV` a dvojicu pre zariadenie do poľa `/dmMappers`.

5.4.2 Zber a transformácia dát pre ethernet rozhrania

Konzolová aplikácia `ip` poskytuje informácie o rozhraniach. Transformačná aplikácia získa informácie a pre každé rozhranie vytvorí štvoricu a uloží ju do poľa `/ethernet/interfaces` tak ako popisuje tabuľka 5.35. Pre bránu vytvorí trojicu a uloží ju do poľa `/ethernet/gates`.

N	popis
0	názov rozhrania
1	stav
2	adresa MAC
3	pole IP adries, viac tab. 5.36

Tabuľka 5.35: Štvorica pre ethernet rozhranie

N	popis
0	adresa
1	verzia
2	maska

Tabuľka 5.36: Trojica pre ethernet adresu

N	popis
0	adresa
1	verzia adresy
2	názov rozhrania, cez ktoré je brána dosiahnuteľná

Tabuľka 5.37: Trojica pre bránu

5.4.3 Spracovanie dát

Spracovanie dát je v práci poňaté získaním vstupu s dátami, uložením, vyhodnotením a poslaním odpovedi klientovi. Incidentsy ako snaha o aktualizovanie neaktívnej entity, či akákoľvek iná chyba je zaznamenávaná a hlásená.

Autentizácia

Prvým krokom je autentizácia serveru, čo spočíva v overení správnosti autentizačného reťazca. Ten sa porovná s reťazcom pre daný server získaným z databázy podľa hostname a domény. Pre server tiež musí existovať adresa IP v entite `ethernet_ip`, ktorá sa zhoduje s adresou IP odosielateľa požiadavky. Ak autentizácia zlyhá, incident sa zaznamená.

Systém umožňuje režim automatického pridávania serverov, kedy požiadavky nie sú autentizované spôsobom popísaným v predchádzajúcom odseku. Režim by mal byť zapnutý iba nevyhnutne dlhú dobu a následne by mal byť vypnutý. Ak je pridávanie zapnuté v prípade nového servera, je automaticky pridaný pod daným hostname, doménou a autentizačným reťazcom. Tiež je znížený stupeň bezpečnosti autentizácie a autentizuje sa len podľa hostname, domény a autentizačného reťazca.

Ethernet rozhrania a brány

Po autentizácii sa aktualizujú rozhrania a príslušné adresy IP pre daný server. To znamená – pridajú sa nové a ak sa v databáze nachádzajú rozhrania či adresy IP, ktoré vo vstupnom súbore nie sú uvedené, ich atribút `aktivni` sa nastaví na hodnotu 0 a zaznamená sa aktuálny čas do atribútu `cas_neaktivni`. O udalosti je informovaný správca.

Následne sa pre server skontroluje aktuálnosť údajov o IPv4 a IPv6 defaultnej bráne.

Disky

Ďalší krok spočíva v aktualizovaní dát diskov a v pridaní nových diskov. Disk je vyhľadaný v databáze podľa sériového čísla a ak záznam neexistuje, pridajú sa do databázy údaje o disku, záznamy pre SMART atribúty, self testy a SMART chyby. V prípade, ak sa disk v databáze nachádza, aktualizujú sa údaje o disku.

Pre každý disk sa skontroluje stav posledného self testu, či nedošlo k zmene aktivity, systémového zariadenia, či kapacity. Prípadné zmeny sú hlásené.

Pri ukladaní záznamov SMART atribútov sa najprv získa posledná RAW hodnota daného atribútu a hodnota, o ktorú sa musí líšiť nová hodnota, aby bola zaznamenaná. Hodnota je uložená v atribúte `zaznamenavat_zmenu` entity `smart_attr_konst`. Veľkosť zmeny je zaznamenaná v atribúte `raw_zmena`. Nový záznam je uložený aj v prípade, že došlo k zmene hodnoty value či worst, či v prípade hlásenia nejakej chyby počas kontrolovania aktuálneho stavu oproti nastaveným parametrom sledovania, ako je `max_raw`, ... Forma hlásenia zodpovedá nastaveniu daného SMART atribútu správcom (e-mail, udalosť, oboje alebo žiadne). Ak nejaký parameter sledovania nie je nastavený (hodnota NULL), tak proti tomuto parametru stav nie je vyhodnocovaný. Počas spracovávania SMART atribútov sú kontrolované aj samotné definície SMART atribútov, ako napr. názov. Tie sa môžu zmeniť po aktualizácii systému.

Nasleduje uloženie záznamov o SMART chybach a self testoch, ak disk poskytuje zoznam výsledkov. Short test má hodnotu typ 1, long test 2, selective 3, vendor 4 a ostatné typ 5. Ukladajú sa len nové záznamy, čo sa určuje podľa hodín disku a dátumu kontroly, keďže niektoré disky po reštarte servera reštartujú svoje hodiny a teda viac self testov môže byť vykonaných v rovnakom čase diskových hodín. Nové SMART chyby a chybné testy sú hlásené udalosťou a e-mailom.

Zároveň sa skontroluje čas posledného vykonania long a short self testov. V atribútoch `perioda_h_short` a `perioda_h_long` entity `disk_jednotka` je uložená hodnota, ktorá určuje, s akou hodinovou periódou majú byť vykonávané self testy. Ak podľa času má byť test vykonaný, alebo je nastavený príznak `spustit_long` alebo `spustit_short`, tak v odpovedi na požiadavok (HTTP) sa nachádzajú shell príkazy pre spustenie testov. V danom čase je spustený iba jeden test, pričom prednosť má test long.

Diskové polia

V ďalšom kroku sa aktualizujú a pridávajú dáta pre diskové polia, RAID radiče a disky v diskových poliach. Spárovanie so záznamami v databáze je nasledujúce.

- softvérový RAID – podľa UUID
- hardvérový RAID – podľa sériového čísla radiča a adresy jednotky
- disk – podľa sériového čísla v entite `disk_jednotka`, ktorá generalizuje hľadanú entitu `disk_raid_jednotka`

Systémové zariadenie pre virtuálnu jednotku HW diskového poľa je zistené pomocou SCSI NAA Id jednotky a informácií uložených vo vstupnom súbore JSON v `/disksById`.

Systém kontroluje, či nedošlo k zmenám v architektúre zapojenia, či už diskov alebo samotných polí. Tiež je kontrolovaná kapacita, stav polí a diskov, systémové zariadenia, ... Na všetko systém upozorňuje.

LVM

Proces pokračuje aktualizovaním údajov o naviazaní logických zväzkov na fyzické prostredníctvom LVM skupiny. Aplikáciou `lsblk`, ktorej výstup je uložený vo vstupnom súbore v `/lsblk`, je možné zistiť, ktoré logické zväzky sú vytvorené z ktorých fyzických zväzkov. Následne sa z informácií v `/lvmLogicalVolumes` zistí LVM skupina konkrétnych logických zväzkov, ktorá je aj skupinou konkrétnych fyzických zväzkov. Následne sa aktualizujú údaje o stave logických zväzkov a LVM skupín, prípadne sa vytvoria nové záznamy, ak neexistujú. Systém opäť upozorňuje na zmeny v architektúre, stave, kapacite a podobne.

Body pripojenia

Posledným krokom je aktualizácia dát bodov pripojenia. Vo vstupnom súbore v `/lsblk` sú uložené informácie, aké logické a fyzické zväzky majú aké body pripojenia. S využitím týchto informácií sa získajú entity **svazek** pre dané body pripojenia. Ak je odlišné systémové zariadenia, bod pripojenia je deaktivovaný a vytvorený nový – je vytvorený z iného zariadenia a pripojený na rovnakú cestu. Systém skontroluje body oproti nastaveniam v entite **ostatni**, a to:

- `d_i_uzly_max_zmena` – max zmena využitých i-uzlov,
- `d_i_uzly_max_obsazenost` – max využitost (%) dostupných i-uzlov,
- `d_i_uzly_max_rel_zmena` – max relatívna zmena využitých i-uzlov,
- `d_i_uzly_max_rel_zmena_vsech` – max relatívna zmena dostupných i-uzlov,
- `d_mount_kapacita_max_obsazenost` – max povolená využitost kapacity,
- `d_mount_kapacita_max_rel_zmena` – počet percent, o maximálne ktorý sa môže zmeniť využitost kapacity.

5.5 Zber a spracovanie systémových záznamov

Dokument PCI DSS² [19], ktorý sa zaoberá bezpečnosťou platobných kariet, hovorí, že procedúra pre monitorovanie bezpečnostných záznamov by mala byť vykonávaná na dennej báze, teda aspoň raz za 24 hodín. Takáto frekvencia však neumožňuje včasné reagovanie na práve prebiehajúci útok. Ak však monitorujeme záznamy každú hodinu, získame možnosť včasného reagovania napríklad na word-list, či DDOS útoky, ktoré sa typicky prejavujú zvýšeným počtom záznamov, teda aj prudším nárastom veľkosti.

Získanie definícií monitorovaných súborov s pravidlami

Práca navrhuje skript, ktorý je spúšťaný cron úlohou každú hodinu. V prvom kroku získa údaje o serveri – hostname, doménu a autentizačný refazec a spolu s informáciami o ethernet rozhraniach a defaultných bránach odošle požiadavok pre získanie definícií monitorovaných súborov spolu so serializovanými pravidlami vo formáte JSON. Do tohto formátu dáta prevedie Python aplikácia, ktorá má za úlohy vytvárať výstup vo formáte JSON a analyzovať súbory systémových záznamov (ďalej len analyzátor). Požiadavok je vo formáte JSON a údaje majú rovnaký formát ako v module pre monitorovanie úložísk.

Systém KNOTIS autentizuje požiadavok, pre daný server získa z databázy definície monitorovaných súborov a pre ne platné pravidlá. Na výstup vypíše pole JSON, ktorého jednotlivé položky sú 11-tice, ktoré definuje tabuľka 5.38.

²Payment Card Industry (PCI) Data Security Standard (DSS)

N	popis
0	cesta k súboru
1	formát času
2	regulárny výraz pre získanie reťazca času
3	formát času 2
4	regulárny výraz 2 pre získanie reťazca času
5	posledná veľkosť súboru
6	časová známka posledného kontrolovaného záznamu
7	čas modifikácie súboru so záznamami
8	čas modifikácie 1. rotovaného súboru so záznamami
9	maximálny povolený nárast
10	pole pravidiel pre daný súbor, viac tab. 5.39

Tabuľka 5.38: 11-tica pre súbor systémových záznamov

N	popis
0	názov
1	typ
2	parametre – objekt JSON pre jednotlivé parametre pravidla, kde kľúče sú názvy parametrov a hodnoty sú hodnoty alebo reťazce pravidiel

Tabuľka 5.39: Trojica pre pravidlo pre súbor systémových záznamov

Skript získané dáta spolu s autentizačnými reťazcami a ethernet dátami predá analyzátoru.

Analýza súborov

Analyzátor v prvom kroku opäť spracuje autentizačné reťazce a údaje o ethernet rozhraniach a bránach. Následne vďaka prítomnosti dát o súboroch vykoná ich analýzu. Aktuálny súbor so záznamami je analyzovaný ak sa zmenil čas jeho modifikovania. Podobne aj prvý rotovaný súbor. Komprimované rotované súbory nie sú analyzované. Pre každý takýto správcom definovaný súbor analyzátor vytvorí 11-ticu popísanú tabuľkou 5.40. Tieto N-tice sú uložené v poli JSON `/logs`.

N	popis
0	ceta k súboru
1	celkový nárast
2	nárast po filtrácii
3	true alebo false označujúce, či došlo ku zmazaniu súborov – to sa musí prejaviť v grafe celkovej veľkosti
4	veľkosť súboru
5	časová známka posledného záznamu
6	veľkosť všetkých súborov (aj rotovaných)
7	čas modifikovania súboru
8	čas modifikovania prvého rotovaného súboru
9	text obsahujúci informácie o pravidlách, ktoré majú vyústiť do hlásenia
10	text obsahujúci informácie o problémoch získavania času jednotlivých záznamov

Tabuľka 5.40: 11-tica pre výsledok analýzy súboru systémových záznamov

Či došlo k zmazaniu súborov je detekované tak, že bol spracovaný prvý záznam v súbore a neexistujú rotované súbory.

Ak nefiltrovaný nárast prekročil maximálny povolený, analyzátor získa výstup **logwatch** a uloží ho do výstupného súboru JSON pod kľúč **/logwatch**.

Spracovanie a vyhodnotenie výsledkov

Systém KNOTIS získa vstupný súbor (súbor vo formáte JSON) a autentizuje rovnakou metódou ako v 5.4.3. Následne skontroluje, či sa aktuálne veľkosti súborov nelíšia od posledných hodnôt o viac, ako je definované, a ak áno, systém uloží nový záznam o náraste. V tomto prípade a ak nejaké pravidlo má pozitívny nález, systém vygeneruje udalosť a pošle e-mail správcovi obsahujúci informácie o náraste, pravidlách, ktoré generujú udalosť s príkladmi záznamov a výstup z **logwatch** nástroja.

Kapitola 6

Implementácia modulu monitoringu úložísk

V tejto kapitole je popísaná implementácia modulu pre monitorovanie úložísk. To pozostáva zo zberu dát, spracovania a užívateľského rozhrania.

Modul je nasadený v produkčnom prostredí a implementácia môže mať isté odlišnosti od návrhu, keďže reálna skúsenosť v niektorých prípadoch ukázala potrebu menšej či väčšej úpravy. Zmeny sa týkajú primárne užívateľského rozhrania.

6.1 Migrácia dát

Skript `knotis/ostatni_skripty/monitoring_ulozisk_migrace_dat.php`, ktorý zabezpečuje migráciu dát, je postavený tak, aby mohol byť spúšťaný viackrát, teda na začiatku zmaže (ak existujú) a vytvorí všetky potrebné tabuľky pre modul monitoringu úložísk. Následne podľa návrhu pokračuje migráciou ethernet rozhraní, SW a HW diskovými polí, HW RAID radičov, diskov, ...

Skript zároveň opravuje chybné dáta súčasného monitoringu úložísk, ktorý zle spracovával vstup a vytváral duplicitné záznamy pre disky. Pre duplicitné disky skript prenesie SMART záznamy a ich nastavenia pôvodnej entity daného disku k novej. Tento krok je vykonaný pred migráciou diskov.

Skript pri migrovaní SMART záznamov a záznamov atribútov pre batérie nemigruje opakujúce sa hodnoty, ale iba ak nastane zmena v jednej z hodnôt.

Na konci nastavuje čas neaktivity neaktívnym diskom (starý modul čas neaktivity neevidoval). Ten je nastavený tak, že sa nájde posledný SMART záznam (v pôvodnom monitoringu) a čas neaktivity sa nastaví o deň neskôr oproti času vzniku daného záznamu.

6.2 Zber a transformácia dát do formátu JSON

V zložke pre inštaláciu serverov sa nachádzajú štyri skripty zabezpečujúce zber a transformáciu dát:

- `storage-monitoring.sh` – skript pre zber a odosielanie dát systému KNOTIS
- `storage-monitoring-parser.py` – skript pre transformáciu dát do formátu JSON
- `storage_monitoring_config.py` – modul pre transformačnú aplikáciu obsahujúci regulárne výrazy pre získanie rôznych sekcií a dát

- `storage_monitoring_update_crontabs.sh` – skript, ktorý aktualizuje úlohy spúšťané aplikáciou CRON

Skript `storagemonitoring.sh` je prebraný z pôvodného riešenia kvôli zachovaniu podobnosti pre správcu systému. Je upravený pre potreby súčasného riešenia, a to zberom nových dát (napr. LVM, `dmsetup`, ...) a vynechaním zbytočných príkazov. Základná štruktúra však zostala podobná. Skript po odoslaní dát vo formáte JSON systému KNOTIS odošle odpoveď, ktorá môže obsahovať príkazy pre spustenie na danom počítači. V súčasnosti očakáva iba jeden typ príkazu a to spustenie `self` testov. Takéto príkazy začínajú reťazcom `SMART:.` Tie následne spustí.

Skript `storage-monitoring-parser.py` ponúka nápovedu obsahujúcu informácie o jednotlivých parametroch, ktoré určujú, čo má skript na štandardnom vstupe očakávať a čo má teda transformovať do JSON. Výstup buď vypíše na štandardný výstup alebo do súboru, ktorý je špecifikovaný parametrom `-json`. V prípade využitia tohto parametra transformačná aplikácia tento súbor najprv načíta a do tejto štruktúry uloží transformované a filtrované dáta a následne súbor prepíše aktualizovanými dátami vo formáte JSON.

Skript `storage_monitoring_update_crontabs.sh` funguje v dvoch režimoch. Ak je použitý prepínač `-c`, skript aktualizuje súčasné nastavenie CRON úloh. Druhou možnosťou je predať skriptu názov súboru (jediný parameter), ktorý obsahuje CRON úlohy a ktorý má byť aktualizovaný. Skript zálohuje aktuálne nastavenia v dočasnej zložke (o umiestnení informuje) a pridá úlohu spúšťajúcu `storage-monitoring.sh`. Čas úlohy je určený následovne:

- ak sa v pôvodnom nastavení vyskytuje úloha pre `analyza.sh`, čas bude navýšený o 2 minúty oproti tejto úlohe,
- v opačnom prípade čas bude nastavený na náhodnú 1.–4. hodinu ráno a 7.–50. minútu, a to z dôvodu, že o celej hodine môže byť spustená úloha pre zálohovanie databázy, či iná náročná úloha.

6.3 Spracovanie dát

Skript `monitoring-ulozisk-spracovani.php` zabezpečuje na strane systému KNOTIS spracovanie dát pre monitoring úložísk. Pre skript je možné nastaviť prioritu a formu upozornenia pre mnohé udalosti (mimo atribútov batérií a SMART) v globálnom poli, ktoré je možné nájsť v súbore `knotis/knotis_monitoring3/tridy/priority.php`. Pre danú kategóriu a podkategóriu je možné buď priradiť prioritu (e-mail odosielaný nie je) alebo pole s kľúčmi `priorita` a `email`, kde hodnota pre `email` (1/0) určuje, či má byť e-mail poslaný. Trieda `knotis/knotis_monitoring3/tridy/StorageMonitoringLogger.php` poskytuje rôzne metódy, ktoré reprezentujú jednotlivé hlášky. Trieda zabezpečuje podporu pre ich hlásenie žiadanou formou.

Prvým krokom je získanie vstupu a kontrola, či neobsahuje nepovolené znaky, čo zabezpečuje funkcia `kontrolaRetezce`. Následne sú spracované ethernet rozhrania, teda sú aktualizované ich atribúty a ich IP adresy. Pridávajú sa nové alebo deaktivujú neaktívne.

Disky a SMART dáta

Druhým krokom je spracovanie SMART dát diskov. Tieto dáta obsahujú aj základné informácie o disku ako model, kapacita apod. Podľa sériového čísla sa zistí, či disk existuje,

a aktualizujú sa jeho údaje (spolu so zapojením apod.), prípadne je v systéme vytvorený nový disk. Ďalej sú spracované SMART atribúty – vytvárajú sa nové záznamy, pokiaľ RAW hodnota sa zmenila o nastavenú hodnotu a vyhodnocujú sa podľa návrhu parametre sledovania. Počas ukladania nových záznamov sú nastavované indexy na najnovší a 2. najnovší záznam pre daný SMART atribút.

SMART chyby sú ukladané v prípade, ak ich číslo je väčšie od čísla poslednej uloženej – teda iba novšie záznamy. U self testov je to zložitejšie, nakoľko niektoré disky po vypnutí môžu reštartovať svoje hodiny. Z databázy je získaných posledných 21 záznamov (toto číslo sa ukázalo ako najbežnejší počet uchovávaných záznamov v SMART log-u). Postupne sa prejde SMART záznam testov a ak sa v spomínaných 21 záznamoch nachádza test s daným časom disku (lifetime), tak nie je uložený. Táto metóda nie je „stopercentná“ a pri častom reštartovaní diskových hodín môže byť chybová. Počítače skupiny KNOT však bežia pomerne dlho a táto situácia sa ich netýka. Počas ukladania testov je opäť nastavený odkaz na najnovší záznam short a long testu.

Nakoniec sú pre jednotlivé disky spúšťané self testy (short a long). Podľa návrhu sú spúšťané vtedy, ak to vynúti správca alebo ubehne doba, po ktorej majú byť spustené znovu. To sa kontroluje jednak tým, že pri spustení testu si systém uchová informáciu o čase spustenie, alebo tým, že ubehla daná doba od naposledy uloženého testu zodpovedajúceho typu (short/long). Táto metóda bola pridaná po tom, ako sa objavil incident, keď systém vyhodnotil, že má byť test spustený, avšak v skutočnosti spustený nebol kvôli chybe. A je neprípustné, aby sa čakalo na ubehnutie ďalšej celej periódy. Test má byť spustený znovu pri kontrole o ďalší deň, kedy aj prípadný long test už skončí. Ak neskončí, systém síce vydá príkaz pre spustenie, avšak disk ho odmietne spustiť, kvôli už bežiacemu testu. Eventuálne, keď skončí, systém uloží nový long test a ďalší už spúšťať nebude.

Počas spracovania SMART údajov, kedy skript načíta všetky disky na danom serveri, je overené, či sa disky nachádzajú stále v diskovom poli, a prípadne sú upravené aj tieto informácie.

Diskové polia

Tretím krokom je spracovanie HW diskových polí. Podľa návrhu sú aktualizované hodnoty pre radiče, ich virtuálne jednotky, ich disky a prípadné batérie či kondenzátory. Systém upozorňuje na zmenu architektúry zapojenia a zmenu atribútov, napr. kapacity a podobne. Batérie sú kontrolované obdobným spôsobom ako SMART atribúty. Pre disky a jednotky je správca systémom upozorňovaný na nežiadúce stavy.

Virtuálne jednotky sú mapované na záznamy v databáze podľa sériového čísla radiča a adresy na ňom. Môže sa teda stať, že správca zmaže starú jednotku a vytvorí novú na rovnakej adrese, v databáze sa však len aktualizujú dáta.

Názov zariadenia je získaný nasledovne: Pre niektoré RAID jednotky na konkrétnych modeloch radičov SMART aplikácia poskytuje základné informácie (a to pre konkrétne systémove zariadenie, napr. sda). V tomto výstupe je tzv. UNIT ID alebo sériové číslo. Refazec SCSI NAA ID z údajov o HW RAID jednotke je buď rovnaký ako UNIT ID alebo podrefazcom sériového čísla v SMART dátach. Pre ostatné RAID jednotky, pre ktoré takéto SMART dáta nie sú k dispozícii, je možné zistiť zariadenie pomocou informácií v `/dev/disk/by-id`. SCSI NAA ID a identifikátor pre dané zariadenie majú spoločný podrefazec začínajúci na 9. znaku o dĺžke 11 znakov.

Štvrtým krokom je spracovanie údajov o SW RAID jednotkách a ich diskoch. Systém podľa návrhu skontroluje architektúru zapojenia a jednotlivé parametre a stavy jednotiek

alebo diskov. Správca má možnosť definovania stavov, medzi ktorými keď virtuálna jednotka prepína, systém zmenu nehlási ale iba uloží. Toto je možné definovať v globálnej premennej `ignorovneVsePrepinanieMedziStavmiSWRaidu` a `ignorovneMailyPrepinanieMedziStavmiSWRaidu`. Druhá premenná na rozdiel od prvej potlačí len e-mail, nie udalosti.

LVM a body pripojenia

Predposledným krokom je aktualizácia informácií o LVM štruktúre. Vytvárajú a kontrolujú sa LVM skupiny a hlásia sa prípadné zmeny, nastavujú sa LVM skupiny fyzickým zväzkom (ktoré generalizujú diskové polia a disky). Následne sa pridávajú a kontrolujú logické zväzky.

Posledným krokom je kontrola bodov pripojenia. Podľa návrhu je skontrolovaná zmena kapacity a miera jej využitia a podobne aj i-uzle. V prípade, ak neexistuje zväzok s názvom zariadenia, ako má bod pripojenia, pomocou výstupu aplikácie `lsblk` je možné určiť systémové zariadenie, z ktorého je vytvorený (bod pripojenia pripája partíciu). V prípade LVM je názov zariadenia totožný.

Triedy pre podporu práce s databázou

Z dôvodu väčšej prehľadnosti a jednoduchosti skriptu pre spracovanie dát práca prináša triedy zabezpečujúce podporu pre SQL príkazy nad danými entitami. Primárny požiadavok je poskytnúť podporu pre prácu s databázou a tabuľkou tak, že pre danú tabuľku sa vytvorí trieda, ktorá obsahuje konštanty definujúce názov tabuľky, primárny kľúč a názvy stĺpcov.

Podporu pre toto zabezpečuje abstraktná trieda `DBObject`, ktorá poskytuje základné metódy pre získanie konkrétneho riadku, a to formou inštancie daného objektu. Ten poskytuje statické metódy `GET_BY_ID`, `GET_ALL`, `GET_BY_CONDITION`, ... Pre ilustráciu predstavme tabuľku `A`, ktorá obsahuje stĺpce `id` a hodnota.

Trieda predpokladá, že užívateľ pracuje s triedami reprezentujúcimi jednotlivé entity a tieto triedy rozširujú triedu `DBObject`. Ďalej `DBObject` očakáva, že podtrieda obsahuje konštanty `TABLE_NAME` a `ID`, ktoré obsahujú reťazce pre názov tabuľky a názov primárneho kľúča. Viacnásobný primárny kľúč nie je podporovaný. Ďalej sa očakáva, že konštanty reprezentujúce jednotlivé stĺpce tabuľky (spolu s primárnym kľúčom) sú anotované `@DBObjectField`. Entita `A` teda bude reprezentovaná triedou `A` (rozširuje triedu `DBObject`), ktorá obsahuje konštanty podľa tab. 6.1.

názov konštanty	anotácia	hodnota
<code>TABLE_NAME</code>	–	<code>A</code>
<code>ID</code>	<code>@DBObjectField</code>	<code>id</code>
<code>VALUE</code>	<code>@DBObjectField</code>	<code>val</code>

Tabuľka 6.1: Príklad triedy `A`, ktorá reprezentuje entitu `A`

Je zabezpečená podpora pre ukladanie nového riadku (`INSERT`) alebo aktualizovanie existujúceho (`UPDATE`) metódou `persist`. Koncept je taký, že ak užívateľ vytvára nový riadok, vytvorí novú inštanciu triedy `A` (`new A()`), prípadne ak chce získať existujúci, tak použije napr. `A::GET_BY_ID($myID)`.

K hodnotám stĺpcov riadku je možné pristúpiť využitím dvoch metód, konkrétne: `$aRow->getDBProperty(A::VALUE)` a `$aRow->setDBProperty(A::VALUE, $value)`.

Keďže databáza je navrhnutá tak, že využíva koncept abstrakcie (entita **svazek** rozširuje entity **fyzicky_svazek**), DBObject myslí aj na toto. V prípade, ak máme triedu B (tabuľka B), ktorá je podtriedou triedy A, DBObject pri ukladaní entity B aktualizuje alebo pridá riadok do tabuľky A. Pri načítavaní existujúceho získa hodnoty pre obe tabuľky jedným SQL príkazom. V prípade, ak by obe tabuľky obsahovali stĺpce rovnakého názvu, tak ich rozlíšenie nie je podporované. Ak B obsahuje stĺpec `val`, tak po uložení bude prepísaná hodnota v tabuľke A hodnotou z tabuľky B.

6.4 Uživatelské rozhranie

Táto sekcia predstavuje uživatelské rozhranie, ktoré je implementované podľa návrhu. Niektoré stránky sa mierne líšia, pretože takto viac vyhovujú práci správcu systému. Časť obrázkov je otočených o 90 stupňov, aby boli čitateľnejšie prípadne nie sú kompletne a zobrazujú len výsek najdôležitejších informácií. Tiež je možné, že kvôli čitateľnosti sú upravené šírky niektorých elementov. Tabuľka 6.2 zobrazuje zoznam súborov a pohľadov, ktoré implementujú. Pre súbor tiež uvádza, ktorý návrh implementuje. Názvy súborov obsahujú v skutočnosti prefix `knotis_monitoring3_`, ktorý v tabuľke kvôli úspore miesta nie je. URL adresa môže byť napr. `localhost/knotis/index2.php?str=&varStr=X`. Parameter `str` je uvádzaný v tabuľke a označuje stránku, ktorá má byť zobrazená. Parameter `varStr` označuje variáciu stránky a často rozlišuje, či sa jedná o pridanie alebo o úpravu objektu. Ak je hodnota 0, je to pridanie, a ak 1, tak úprava existujúceho objektu. Ďalej sú využívané parametre `smazPol`, `uprPol` a iné, ktoré obsahujú identifikátor objektu pre konkrétnu akciu.

suffix názvu súboru	návrh	URL str	obr.
bbu_attr_grafy.php	–	137	6.22
bbu_attr_historie.php	B.16	136	6.20
bbu_detail.php	B.15	123	6.23
bbu_uprav.php	B.17	124	6.24
bbu_zoznam.php	B.14	122	6.25
disk_detail.php	B.9	113	6.11
disk_self_test_historie.php	B.13	114	6.8
disk_smart_grafy.php	–	135	6.10
disk_smart_historie.php	6.6	134	6.6
disk_uprav.php	B.10	115	6.12
disk_zmaz.php	B.8	127	–
disk_zoznam.php	B.5	112	6.13
eth.php	B.3	132	6.27
hw_raid_detail.php	B.20	120	6.15
hw_raid_zoznam.php	B.19	118	6.17
ip.php	B.4	133	6.28
problemolve_disky.php	B.6	126	6.4
raid_zmaz.php	B.23	121	6.14
server_eth_sprava.php	B.2	131	6.26
smart_agregace.php	B.7	143	6.5
smart_skratka_uprav.php	–	144	6.2
smart_skratka_zoznam.php	–	145	6.3
sprava_bbu.php	B.18	125	6.19
sprava_diskov.php	B.11	116	6.7
sw_raid_detail.php	B.22	119	6.16
sw_raid_zoznam.php	B.21	117	6.18

Tabuľka 6.2: Zoznam súborov, ktoré implementujú jednotlivé wireframe z návrhu

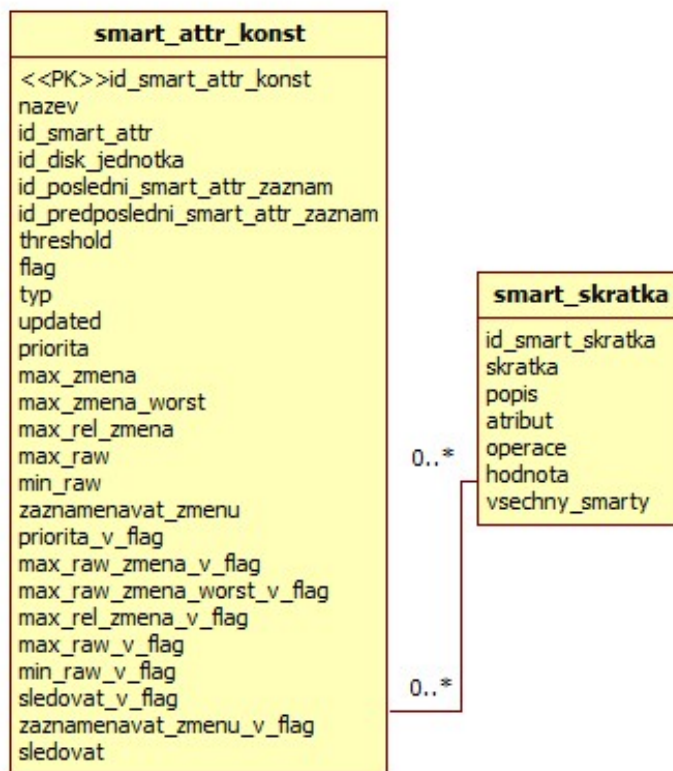
Disky

Obrázok 6.4 zobrazuje pohľad na problémové disky. Pohľad zobrazuje výsek základných údajov v tabuľke, informáciu o probléme, skratky a dátum kontroly. Tabuľka obsahuje ešte stav, bod pripojenia, ... Po prvej implementácii sa ukázalo, že navrhnuté skratky navýšenia parametrov sledovania sú prospešné a urýchľujú prácu. Častý prípad použitia je sledovanie atribútu `Spin_Up_Time` tak, že maximálna raw hodnota je cyklicky navýšovaná o rovnakú hodnotu vždy po jej dosiahnutí. Napr. vždy o 10000, a takto je disk pravidelne sledovaný. To vyústilo do potreby definovania vlastných skratiek, ktoré sú vzťahované k aktuálnej hodnote parametru sledovania. Pôvodne navrhnuté, ktoré upravujú hodnotu parametrov, fungujú nasledovne:

- **+-10%** – navýšenie/zníženie parametru sledovania o 10% oproti aktuálnej raw hodnote
- **=** – nastavenie na aktuálnu hodnotu raw
- **nesledovat** – nastavenie na NULL
- **výchozí** – nastavenie na východiskovú hodnotu

- **události** – nastavenie levelu sledovania na generovanie len udalostí v systéme (a teda nezobrazovanie v zozname problémových diskov)

Skratky sú v systéme uložené v entitách **smart_skratka**, ktorú znázorňuje ER diagram 6.1.



Obr. 6.1: ER diagram SMART skratiek

Správca má možnosť definovať, pre aký parameter sledovania je skratka platná, SMART atribúty, operáciu, hodnotu, ... Teda napr. pri zvolení + skratka navýši aktuálne nastavenú max raw hodnotu o danú hodnotu. Skratka je zobrazená iba pre vybrané SMART atribúty a vybraný parameter sledovania. Pohľad pre pridanie novej, či úpravu existujúcej je možné vidieť na obr. 6.2 a zoznam skratiek, kde je možnosť mazania, na obr. 6.3.

Úprava zkratky pro SMART

Upravte zkratku pro nastavení SMART:

Zkratka: *

Popis: *

Operace: ☒ + ☐ - ☐ = ☐ * *







Atribut: ☐ min raw
☒ max raw
☐ max změna
☐ max rel změna *

Hodnota: *

SMART atributy ☐ vše
☐ 1: Raw_Read_Error_Rate
☐ 7: Seek_Error_Rate
☐ 8: Seek_Time_Performance
☒ 9: Power_On_Hours

Obr. 6.2: Pridanie/úprava SMART skratky

Seznam zkratek pro SMART

<input type="button" value="Přidat zkratku"/>		<input type="button" value="Zpět"/>			
Zkratka	Popis	Akce	Smarty	Operace	Hodnota
+10t	+ 10 000	 	9: Power_On_Hours	+	10000
+5t	+ 5000	 	9: Power_On_Hours 193: Load_Cycle_Count	+	5000
+10	+10	 	192: Power-Off_Retract_Count	+	10

Obr. 6.3: Zoznam SMART skratiek

Problémové disky

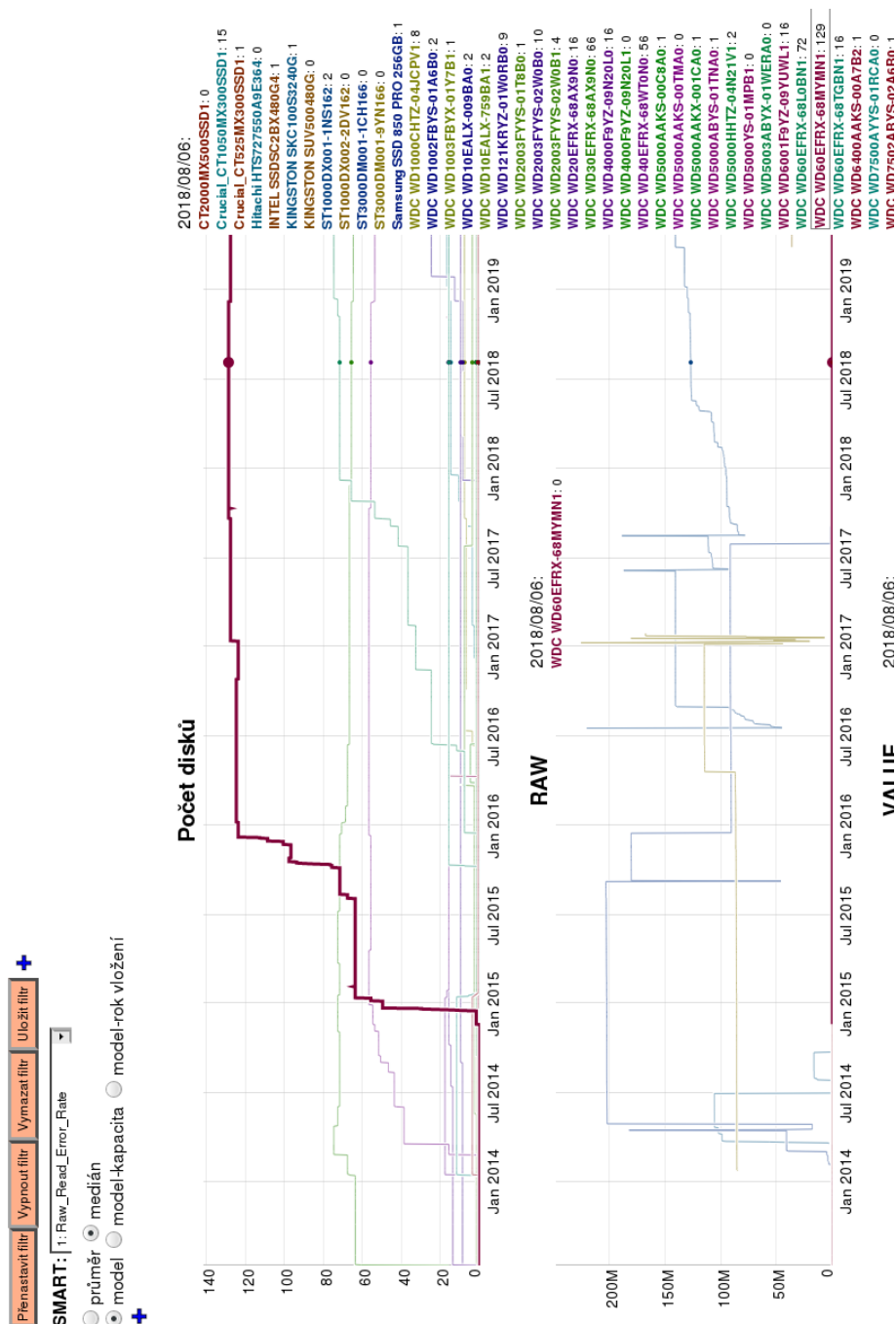
seznam skratek Přidat skratku
 Přenastavit filtr Vypnout filtr Vymazat filtr Uložit filtr +

Model ▲▼▲▼	Sériové číslo ▲▼▲▼	Kapacita ▲▼▲▼	Server ▲▼▲▼	RAID	Chyba	Akce	Datum kontroly ▲▼▲▼
					<input checked="" type="checkbox"/> dosáhnuti MAX RAW <input checked="" type="checkbox"/> dosáhnuti MIN RAW <input checked="" type="checkbox"/> dosáhnuti threshold <input checked="" type="checkbox"/> změna when_failed <input checked="" type="checkbox"/> short self test <input checked="" type="checkbox"/> long self test Nastav vše		
WDC WD2003FYYS-02W0B0	WD-WMA Y03402008	2 TB	minerva1	c0/u0 (minerva1) p0	Power_On_Hours MAX RAW > 60000: 60305	= +10% výchozí události 1t nesledovat	22.04.2019
WDC WD40EFRX-68WT0N0	WD-WCC4E0979366	4 TB	athena2	c0/v1 (athena2) did 35 245:1	Raw_Read_Error_Rate MAX RAW > 1: 3	= +10% výchozí události 1t 1t nesledovat	21.04.2019
WDC WD40EFRX-68WT0N0	WD-WCC4E0977945	4 TB	athena2	c0/v0 (athena2) did 27 21:6	Multi_Zone_Error_Rate MAX RAW > 1: 2	= +10% výchozí události 1t nesledovat	21.04.2019
					Raw_Read_Error_Rate MAX RAW > 1: 50	= +10% výchozí události 1t 1t nesledovat	

Obr. 6.4: Problémové disky

Agregácia SMART atribútov

Obrázok 6.5 zobrazuje agregáciu konkrétneho SMART atribútu pre vybrané disky. Výber diskov spočíva vo filtrácii požadovaných diskov podobne ako v zozname diskov (obr. 6.13). Na obrázku je zoznam diskov možno zobrazíť ikonou +. Grafy majú synchronizovaný výber, teda na obr. je vidieť, že je vybraný model s najväčším počtom diskov (128) a v grafe pre agregované raw hodnoty je automaticky zvýraznený agregovaný priebeh pre rovnaký model. Tiež je zobrazená raw hodnota pre daný model.



Obr. 6.5: SMART agregácia

Detail disku a zoznam diskov

Detail disku zobrazuje obr. 6.11, kde je možno vidieť tlačítka pre rýchlu úpravu sledovaných SMART atribútov, pretože sa po implementácii v zozname problémových diskov ukázali ako užitočné. Tabuľka na obrázku obsahuje ešte ostatné nastavenia sledovania pre jednotlivé atribúty. Pod touto tabuľkou sa nachádza história self testov (max. 21), kde sú tlačítka pre vynútenie spustenia short a long testu pri najbližšej kontrole a odkaz na celú históriu testov (obr. 6.8). Nižšie je história SMART chýb. Tlačítkom + pri atribútoch je možné zobraziť nedávnu históriu zaznamenaných zmien. Na obr. to síce nie je vidieť, ale systém zvýrazňuje konkrétne hodnoty a parametre sledovania, ktoré sú „porušené“. Napr. môže byť zvýraznená hodnota raw a max raw, ak raw prekročila hodnotu max raw. Grafické znázornenie histórie sa tiež ukázalo ako dobré, a teda bolo presunuté z detailnej histórie SMART atribútu (obr. 6.6) do detailu disku. Je ho možné zobraziť ikonkou grafu pre konkrétny atribút. V grafe má správca možnosť priblížiť horizontálny či vertikálny výsek, a to viacnásobne po sebe. Dvojklikom zas oddaľuje. Graf zobrazuje raw hodnoty na ľavej vertikálnej ose a normalizované hodnoty a najhoršie (value, worst) na pravej. To zobrazuje obrázok 6.9. Správca môže zobraziť všetky grafy (odkaz grafy) pod sebou, viď obr. 6.10.

Historie naměřených hodnot disku

Informace o disku:

Model: WDC WD2003FYYS-02W0B0
Sériové číslo: WD-WMAY03402008
Kapacita: 2 TB [2000398934016 B]
Server: [minerva1](#)
RAID: [c0/u0 \(sdb\)](#)
Adresa disku: did :
Stav: OK
Datum kontroly: 17:18 22.04.2019
Body připojení: /mnt/data-in
Zařízení:
Datum vložení: 15:10 15.07.2013
Aktivní: ano
Chybný disk: ne označit jako chybný

[Upravit nastavení disku](#)

[Zpět](#)

Nastavení pro Raw_Read_Error_Rate

Smart ID	Typ	Flag	min Raw	max Raw	Max změna RAW	Max rel. změna RAW	Max změna WORST	Sledování	Sledování RAW změny	priorita
1	Pre-fail	47	---	1	1	---	1	p+u+email	1	20

[Nastavit filtr](#) [Vymazat filtr](#) [Uložit filtr](#) +

RAW	Hodnota	Worst	Kdy selhalo	Čas kontroly
<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="text"/>
0	200	200	-	18:32 21.04.2019
0	200	200	-	15:10 15.07.2013

Obr. 6.6: Historie SMART atribútu disku

Zoznam diskov zobrazuje obrázok 6.13. Mazanie diskov vyzerá prakticky totožne s tým, že obsahuje naviac ikonku pre mazanie s potvrdzovacím dialógom.

Nastavenia parametrov sledovania je možné vidieť na obr. 6.12. Po zaškrtnutí zaškrtávacieho tlačítka pre použitie východiskovej hodnoty je zneaktívnená príslušná kolonka a jej hodnota je nastavená na východiskovú hodnotu. Po zrušení zaškrtnutia je hodnota nastavená na pôvodne zadanú. Pod SMART atribútmi sa nachádza tlačítko pre odoslanie formulára.

Obrázok 6.7 zobrazuje možnosť nastavenia rôznych konštánt (automatické pridávanie serveru, parametre sledovanie bodov pripojenia, ...) a východiskových hodnôt pre SMART atribúty. Zaškrtnutím zaškrtávacieho tlačítka a uložením správca nastaví dané parametre sledovania všetkých diskov tak, aby využívali východiskovú hodnotu. Tento prípad užitia je vnímaný ako akcia a po opätovnom načítaní tlačítko opäť nebude zaškrtnuté. V obrázku nie sú zobrazené všetky východiskové parametre sledovania kvôli čitateľnosti obrázku.

Nastavení správy disků

Výchozí nastavení pro správu disků:

Nastavení výchozích ostatních hodnot:

Self test short: dnů hodin *

Self test long: dnů hodin *

automatický přidání serveru / ☒ autentizace jen dle názvu a domény:

I uzly max zmena: *

I uzly max obsazenost (%): *

I uzly max rel zmena (%): *

I uzly max rel zmena vseh (%): *

Mount kapacita max obsazenost (%): *

Mount kapacita max rel zmena (%): *

Nastavení výchozích hodnot pro SMART:

	Max zmena	Max zmena worst	Max rel zmena (%)	Max raw	Min raw
Raw Read Error Rate SmartID: 1	<input type="text" value="1"/> **	<input type="text" value="1"/> **	<input type="text" value="0"/> **	<input type="text" value="1"/> **	<input type="text" value="0"/>

Obr. 6.7: Správa diskov

Historie self-testu disku

Informace o disku:

Model: Crucial_CT1050MX300SSD1

Sériové číslo: 163413B97A60

Kapacita: 1.05 TB [1050214588416 B]

Server: [athena3](#)

RAID: -

Adresa disku: -

Stav: -

Datum kontroly: 13.04.2019 02:33

Body připojení: /

Zařízení: sdb

Datum vložení: 05.09.2017 17:29

Aktivní: ano

Chybný disk: ne [označit jako chybný](#)

[Upravit nastavení disku](#)

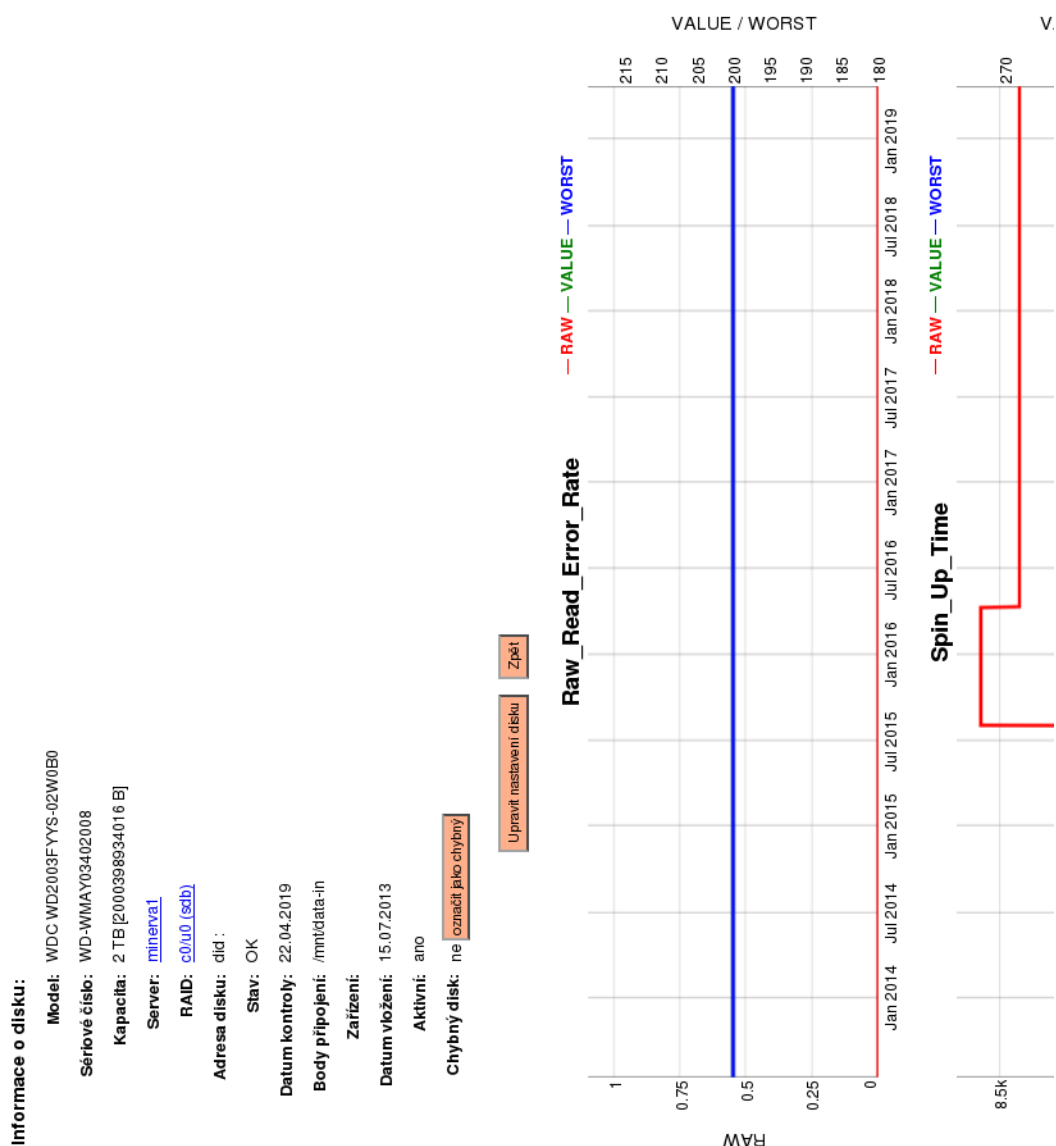
[Zpět](#)

[Nastavit filtr](#) [Vymazat filtr](#) [Uložit filtr](#) [+](#)

Typ	Status	Zbývá	Čas kontroly	Disk hodiny	LBA	text
	<input type="text" value="-"/>		<input type="text"/>			
short	ok	00%	13.04.2019 02:33	13973	-	
short	ok	00%	11.04.2019 02:33	13949	-	

Obr. 6.8: Správa diskov

Historie naměřených hodnot disku



Obr. 6.10: Grafické zobrazení historie všech SMART atribútů disku

Detail disku

Informace o disku:

Model: WDC WD2003FYYS-02W0B0
Sériové číslo: WD-WMAY03402008
Kapacita: 2 TB [2000398934016 B]
Server: [minerva1](#)
RAID: [c0,u0 \(sdb\)](#)
Adresa disku: p0
Stav: OK
Datum kontroly: 17:18 22.04.2019
Body připojení: /mnt/data-in
Zařízení:
Datum vložení: 15:10 15.07.2013
Aktivní: ano
Chybny disk: ne označit jako chybný

Upravit nastavení disku Zpět

Aktuální SMART hodnoty

[grafy](#) [seznam skratek](#) Přidat skratku

	ID: Název atributu	Typ	Flag	RAW	min RAW	max RAW	hodnota	Prah	Nejhorší hodnota	Max změna RAW	Max rel. změna RAW	Čas kontroly
	1: Raw_Read_Error_Rate	Pre-fail	47	0 0	---	1 <div><div>=</div><div>+10%</div><div>-10%</div></div>	200 200 <div><div>=</div><div>+10%</div><div>-10%</div></div> <div>11111</div>	51	200 200	1 <div><div>=</div><div>+10%</div></div>	---	18:32 21-04-2019 15:10 15-07-2013
	3: Spin_Up_Time	Pre-fail	39	8700	---	10000 <div><div>=</div><div>+10%</div><div>-10%</div></div>	253 <div><div>=</div><div>+10%</div><div>-10%</div></div> <div>11</div>	21	253	1 <div><div>=</div><div>+10%</div></div>	---	17:59 11-04-2016 <div><div>=</div><div>+10%</div></div>

Obr. 6.11: Detail disku

Úprava disku



Nastavení pro správu tohoto disku:

Nastavení hodnot pro self testy:

Perioda h short: dnů hodin *

Perioda h long: dnů hodin *

Nastavení hodnot pro SMART:

Nastavit výchozí hodnoty

	Max změna	Max změna worst	Max rel změna (%)	Max raw	Min raw	Zaznamenávat změnu	Sledovat	Priorita
Raw Read Error Rate 0 Smart ID: 1	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value=""/>	<input type="text" value="1"/>	<input type="text" value=""/>	<input type="text" value="1"/>	<input type="text" value="p-u-email"/>	<input type="text" value="20"/>
Reallocated Sector Ct 0 Smart ID: 5	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value=""/>	<input type="text" value="1"/>	<input type="text" value=""/>	<input type="text" value="1"/>	<input type="text" value="p-u-email"/>	<input type="text" value="20"/>
Power On Hours 0 Smart ID: 9	<input checked="" type="checkbox"/>	<input type="text" value="1"/>	<input type="text" value=""/>	<input type="text" value="10000"/>	<input type="text" value=""/>	<input type="text" value="1"/>	<input type="text" value="problémové"/>	<input type="text" value="20"/>
Power Cycle Count 1 Smart ID: 12	<input type="text" value="5"/>	<input type="text" value="1"/>	<input type="text" value=""/>	<input type="text" value="200"/>	<input type="text" value=""/>	<input type="text" value="1"/>	<input type="text" value="problémové"/>	<input type="text" value="20"/>
Unknown Attribute 0 Smart ID: 210	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value=""/>	<input type="text" value="100"/>	<input type="text" value=""/>	<input type="text" value="1"/>	<input type="text" value="p-u-email"/>	<input type="text" value="20"/>

Obr. 6.12: Nastavenie disku

Seznam disků

? Nápověda

Přenastavit filtr		Vypnout filtr		Vymazat filtr		Uložit filtr		+			
Model ▲▼▲▼	Sériové číslo ▲▼▲▼	Kapacita ▲▼▲▼	Server ▲▼▲▼	RAID	Datum kontroly ▲▼▲▼	Stav ▲▼▲▼	Bod připojení ▲▼▲▼	Zařízení ▲▼▲▼	Datum vložení ▲▼▲▼	Aktivní ▲▼▲▼	Chybný ▲▼▲▼
WDC WD30EFRX-68AX9N0			-	-						-	-
WDC WD30EFRX-68AX9N0	WD-WMC1T2917436	3 TB	athena4	md0	03.02.2019	active sync	/mnt/data	sdg	26.02.2014	ano	ne
WDC WD30EFRX-68AX9N0	WD-WMC1T1250716	3 TB	athena5	c0/v0 did 23 245:1	03.02.2019	Onln	/mnt/data		05.12.2018	ano	ne
WDC WD30EFRX-68AX9N0	WD-WMC1T1596775	3 TB	athena5	c0/v0 did 31 245:5	09.01.2019	Onln	/mnt/data		05.12.2018	ne	ne
WDC WD30EFRX-68AX9N0	WD-WMC1T1191049	3 TB	athena5	c0/v0 did 35 245:9	03.02.2019	Onln	/mnt/data		05.12.2018	ano	ne
WDC WD30EFRX-68AX9N0	WD-WMC1T2882842	3 TB	athena5	c0/v0 did 27 245:13	03.02.2019	Onln	/mnt/data		05.12.2018	ano	ne
WDC WD30EFRX-68AX9N0	WD-WMC1T1456807	3 TB	pcknot5	md0	03.02.2019	active sync	/mnt/data	sdb	14.07.2013	ano	ne
WDC WD30EFRX-68AX9N0	WD-WMC1T1610382	3 TB	ivs	-	07.01.2016			sdb	14.07.2013	ne	ne
WDC WD30EFRX-68AX9N0	WD-WMC1T1610731	3 TB	pckorablev	-	27.01.2019		/mnt/data	sdb	15.07.2013	ano	ne
WDC WD30EFRX-68AX9N0	WD-WMC1T1194358	3 TB	athena4	md0	03.02.2019	active sync	/mnt/data	sdc	15.07.2013	ano	ne

Obr. 6.13: Zoznam diskov

Diskové polia

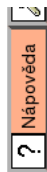
Systém ponúka zoznam HW a SW diskových polí (viď obr. 6.17 a 6.18), odkiaľ vedú odkazy na detaily jednotlivých diskových polí (obr. 6.15 a 6.16). Mazanie je možné pomocou pohľadu na obrázku 6.14. Po odoslaní formuláru a kontrole, či je možné pole zmazať, sú odstránené záznamy o bodoch pripojenia, radič, záznamy o kontrole, ...

Mazázní RAID-ů

Nastavit filtr Vymazať filtr Uložit filtr +				
Server ▲▼▲▼	Zařízení ▲▼▲▼	Operace	Datum kontroly ▲▼▲▼	UUID / adresa ▲▼▲▼
-				
athena1	md1		08.12.2015	52967ead:98c751ed:5801f25b:4354a7c5
athena1	md0		08.12.2015	65230e4f:caffdd00:bed79ed6:26cfae88
athena18	md127		00.00.0000	61cfd677:be908986:60bbd33e:9bee0b2e
athena2			00.00.0000	SV32210187 c0//c0/v
athena2			00.00.0000	SV32210187 c0//c0/v
athena2			00.00.0000	SV32210187 c0//c0/v

Obr. 6.14: Mazanie diskových polí

Detail diskového HW pole



Informace o RAID poli:

Server: [minerva 1](#)
Zařízení: sdd
Bod připojení: /mnt/data-2
RAID level: RAID-6
Adresa: c7/u0
Kapacita: 42 TB [41999840942489 B]
Stav: OK
Datum kontroly: 17:18 22.04.2019
Model: 9750-16i4e
Sériové číslo: SV13300210
Datum vložení: 15:12 15.07.2013

Disky

Model	Serial	Kapacita	Datum kontroly	Stav	Adresa	Datum vložení
WDC WD30EFRX-68AX9N0	WD-WMC1T0883132	3 TB	22.04.2019	OK	p36	15.07.2013
WDC WD30EFRX-68AX9N0	WD-WMC1T0025975	3 TB	22.04.2019	OK	p37	15.07.2013
WDC WD30EFRX-68AX9N0	WD-WMC1T1307212	3 TB	22.04.2019	OK	p38	15.07.2013
WDC WD30EFRX-68AX9N0	WD-WMC1T0028494	3 TB	22.04.2019	OK	p39	15.07.2013

Obr. 6.15: Detail HW diskového pola

Detail diskového SW pole

Informace o RAID:

UUID: 5652a337:3914114b:49ac3275:2fdd0b30

Server: [knot01](#)

Zařízení: md0

Bod připojení:

RAID level: raid5

Kapacita: 18 TB [18003114590208 B]

Stav: clean

Datum kontroly: 02:02 03.02.2019

Datum vložení: 02:39 19.12.2014

Disky

Model	Serial	Kapacita	Server	Datum kontroly	Stav	Zařízení	Datum vložení
WDC WD60EFRX-68MYMN1	WD-WX41D94RN1ES	6 TB	knot01	03.02.2019	active sync	sdb	19.12.2014
WDC WD60EFRX-68MYMN1	WD-WX41D948YJP4	6 TB	knot01	03.02.2019	active sync	sdc	19.12.2014
WDC WD60EFRX-68MYMN1	WD-WX21D9421XV7	6 TB	knot01	03.02.2019	active sync	sdd	19.12.2014
WDC WD60EFRX-68MYMN1	WD-WX41D94RNHY0	6 TB	knot01	03.02.2019	active sync	sde	19.12.2014

Obr. 6.16: Detail SW diskového pola

Seznam HW diskových polí

Přenasavit filtr

Vypnout filtr

Vymazat filtr

Uložit filtr

Obr. 6.17: Zoznam HW diskových polí

Seznam SW diskových polí

<div> <div>Přenasťavití filtr</div> <div>Vypnout filtr</div> <div>Vymazat filtr</div> <div>Uložit filtr</div> </div>											
Server ▲▼▲▼	Zařízení ▲▼▲▼	Bod připojení ▲▼▲▼	Raid level ▲▼▲▼	Kapacita ▲▼▲▼	Stav ▲▼▲▼	Datum kontroly ▲▼▲▼	UUID ▲▼▲▼	Datum vložení ▲▼▲▼	Aktivní ▲▼▲▼		
-			-	>17000					-		
knot01	md0	/mnt/data	raid5	18 TB	clean	03.02.2019	5652a337:3914114b:49ac3275:2fdd0b30	19.12.2014	ano		
knot03	md2	/mnt/data	raid5	18 TB	clean	03.02.2019	c1228878:aa9a927e:f0a7:cf3:00c2f8fc	20.12.2014	ano		
knot04	md0	/mnt/data	raid5	18 TB	clean	03.02.2019	66c25d55:a456d0e6:ce9e91a8:5e2b92a8	08.10.2015	ano		
knot05	md0	/mnt/data	raid5	18 TB	clean	03.02.2019	8a272789:56b66c5d:91572491:5dddeb4f	10.10.2015	ano		
knot06	md3	/mnt/data	raid5	18 TB	clean	03.02.2019	61f8b148:4d299903:e7b802ef:830862e9	20.12.2014	ano		
knot07	md1	/mnt/data	raid5	18 TB	clean	03.02.2019	ab7a2c43:36fd2127:02bb8dbd:02f92d5d	13.01.2015	ano		
knot08	md0	/mnt/data	raid5	18 TB	clean	03.02.2019	e7e14d1a:fab2060c:822df621:855a3219	13.01.2015	ano		
knot09	md0	/mnt/data	raid5	18 TB	clean	03.02.2019	c252f24f:2a604dcf:55c0276b:ab8a34d8	05.09.2017	ano		

Obr. 6.18: Zoznam SW diskových polí

Batérie

Typy pohľadov, ktoré má správca k dispozícii, sú obdobné ako tie pre disky. Je možné zobraziť zoznam batérií (obr. 6.25), kde pribudla akcia pre vymazanie konkrétneho zariadenia. Ďalej je k dispozícii detail batérie (obr. 6.23), pričom správca má možnosť zobraziť graf pre numerické atribúty (obr. 6.21). Ďalšie pohľady sú: nastavenia pre zariadenie (obr. 6.24), história atribútu (obr. 6.20) a grafické zobrazenie histórie všetkých atribútov (ktoré majú numerickú hodnotu), viď obr. 6.22. Do detailu boli pridané tlačítka pre rýchlu manipuláciu s parametrami sledovania, avšak narozdiel od diskov tu nebola potreba definovania vlastných skratiek. Je tu teda možnosť dorovnania na aktuálnu hodnotu alebo navýšenie, prípadne zníženie o 10 %. Obr. 6.19 zobrazuje správu batérií, kde je možné nastaviť východiskové hodnoty pre parametre sledovania.

Nastavení správy baterií

Výchozí nastavení pro správu baterií:

Nastavení výchozích hodnot pro baterie:

	Max zmena	Max rel zmena (%)	Max hodnota	Min hodnota
info_voltage	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
info_current	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
info_temperature	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
info_battery_state	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
firmware_voltage	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
firmware_temperature	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
firmware_lc_requested	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
firmware_lc_active	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
firmware_lc_status	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
firmware_i2c_errors	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
firmware_replacement_required	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>
firmware_remaining_capacity_low	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>

Obr. 6.19: Správa batérií

Historie naměřených hodnot baterie

Informace o baterii:

Sériové číslo: 3808

Název zařízení: bq27541

Název od výrobce: LS36691

Typ: bbu

Server: [athena5](#)

Radič: c0

Napětí: 4100 mV

Kapacita: 1500 mAh

Datum kontroly: 02:36 03.02.2019

Datum vložení: 20:03 07.12.2015

Datum výroby: 13/10/2014

Aktivní: ne (2019-02-14 02:36:47)

[Upravit nastavení](#)

[Zpět](#)

Nastavení pro Voltage

Min hodnota	Max hodnota	Max změna	Max rel. změna	Sledování	Sledování změny	priorita
---	4100	0	---	u+email	1	20

[Nastavit filtr](#)

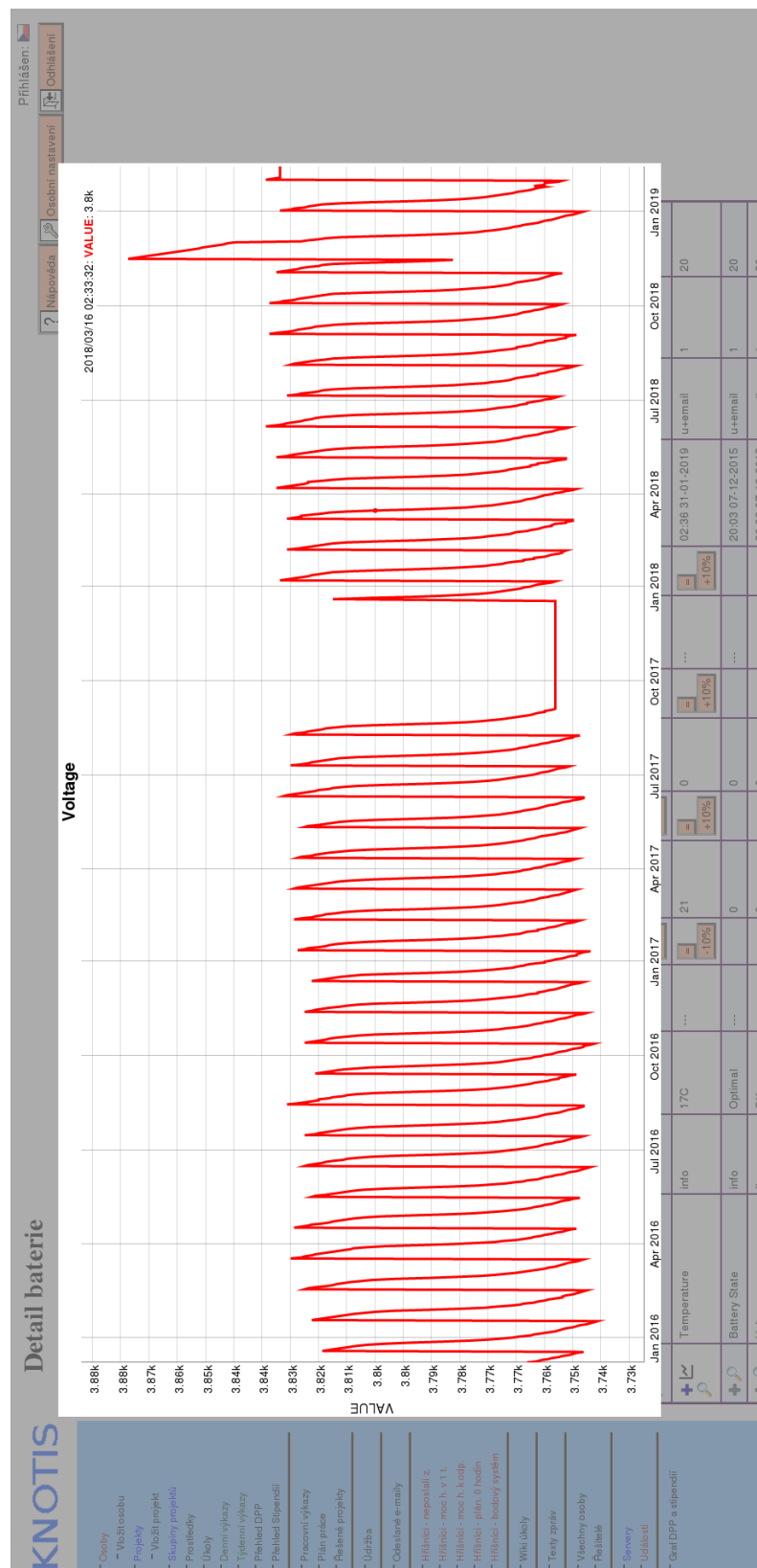
[Vymazat filtr](#)

[Uložit filtr](#)



Hodnota	Čas kontroly
<input type="text"/>	<input type="text"/>
3923	02:36 03.02.2019
3925	02:36 02.02.2019
3927	02:36 01.02.2019
3843	02:36 31.01.2019

Obr. 6.20: História atribútu batérie



Obr. 6.21: Grafické zobrazenie histórie atribútu batérie

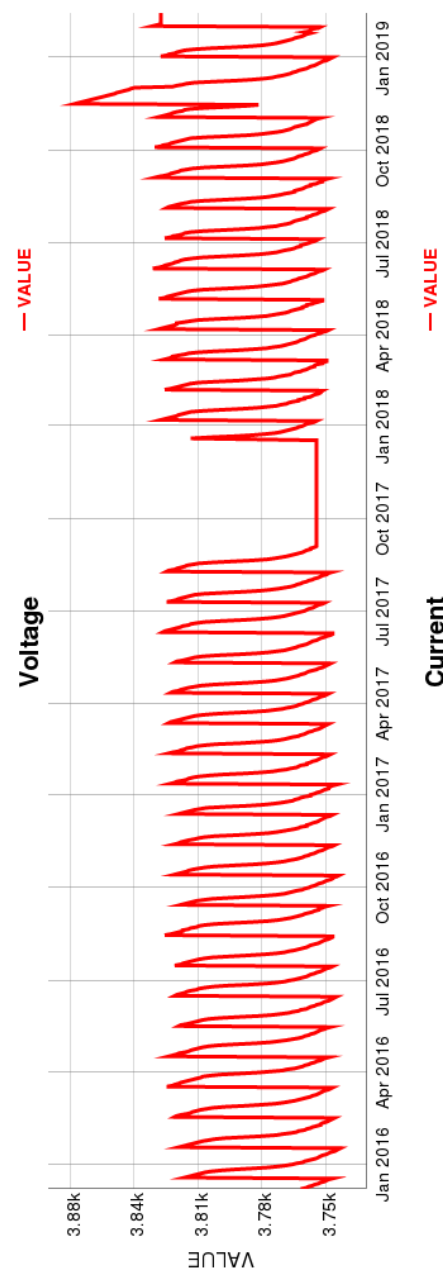
Historie naměřených hodnot baterie

Informace o baterii:

Sériové číslo: 3808
 Název zařízení: bq27541
 Název od výrobce: LS36891
 Typ: bbu
 Server: [athena5](#)
 Radič: c0
 Napětí: 4100 mV
 Kapacita: 1500 mAh
 Datum kontroly: 03.02.2019
 Datum vložení: 07.12.2015
 Datum výroby: 13/10/2014
 Stav: Optimal
 Aktivní: ne (02:36 14.02.2019)

[Upravit nastavení](#)

[Zpět](#)



Obr. 6.22: Grafické zobrazení historie všech numerických atribútů batérie

Detail baterie

Informace o baterii:

Sériové číslo: 3808
 Název zařízení: bq27541
 Název od výrobce: LS36691
 Typ: bbu
 Server: [athena5](#)
 Radič: c0
 Napětí: 4100 mV
 Kapacita: 1500 mAh
 Datum kontroly: 02:36 03.02.2019
 Datum vložení: 20:03 07.12.2015
 Datum výroby: 13/10/2014
 Stav N/A
 Aktivní: ne (02:36 14.02.2019)

[Upravit nastavení](#)
[Zpět](#)

Aktuální hodnoty

[grafy](#)

	Název atributu	Sekce	hodnota	min	max		Max změna		Max rel. změna (%)		Čas kontroly	Sledování	Sledování změny	Priorita
	Voltage	Info	3923mV 3925mV 3927mV 3843mV 3844mV	---	4100		0		---		02:36 03-02-2019 02:36 02-02-2019 02:36 01-02-2019 02:36 31-01-2019 02:36 30-01-2019	U+email	1	20
	Current	Info	0mA	---	0		0		---		20:03 07-12-2015	U+email	1	20

Obr. 6.23: Detail batérie

Úprava baterie

Nastavení pro správu tohoto disku:

Nastavení hodnot pro baterii:

	Max změna	Max rel změna (%)	Max hodnota	Min hodnota	Zaznamenávat změnu	Sledovat	Priorita
Voltage (3848) info_voltage	<input type="text" value="0"/>	<input checked="" type="checkbox"/>	<input type="text" value="4100"/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value="u4email"/>	<input type="checkbox" value="20"/>
Current (0) info_current	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value="u4email"/>	<input type="checkbox" value="20"/>
Temperature (17) info_temperature	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="21"/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value="u4email"/>	<input type="checkbox" value="20"/>
Battery State (Optimal) info_battery_state	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value="u4email"/>	<input type="checkbox" value="20"/>
Voltage (OK) firmware_voltage	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value="u4email"/>	<input type="checkbox" value="20"/>
Temperature (OK) firmware_temperature	<input type="text" value="0"/>	<input type="checkbox"/>	<input type="text" value="0"/>	<input type="text" value=""/>	<input type="checkbox"/>	<input type="text" value="u4email"/>	<input type="checkbox" value="20"/>

Obr. 6.24: Nastavenie batérie

Přihlášen

Nastavit filtr			Vymazat filtr			Uložit filtr			+				
Akce	Sériové číslo	Typ	Server	Raděč	Výrobní název	Název zařízení	Kapacita	Napětí	Datum kontroly	Datum vložení	Datum výroby	Technologie	Aktivní
		<input type="text"/>	-	-	<input type="text"/>	<input type="text"/>			<input type="text"/>	<input type="text"/>		<input type="text"/>	<input type="text"/>
	3808	bbu	athena5	c0 (SV43402534)	LS36691	bq27541	1500 mAh	4100 mV	03.02.2019	07.12.2015	13/10/2014		ne
	4272	cv	minerva2	c0 (SK72771201)	LSI	CVPM02	307 J	0	03.02.2019	19.12.2017	23/06/2017		ano
	551	bbu	athena2	c0 (SV32210187)	LS36691	bq27541	1500 mAh	4100 mV	21.04.2019	07.12.2015	13/09/2014	LION	ano

93

Detail serveru a ethernet rozhrania

Pôvodný súbor generujúci stránku detailu servera bol upravený tak, aby zobrazoval tabuľky obsahujúce dáta z novej verzie monitoringu (obr. 6.29), a to z dôvodu, že pôvodný súbor zobrazuje aj monitoring záťaže. Správca tu vidí všetky informácie pre daný server: ethernet rozhrania, disky, body pripojenia, diskové polia, batérie, LVM zväzky a skupiny. Manažment ethernet rozhraní bol presunutý do samostatného pohľadu (obr. 6.26). Tu je možné vidieť všetky aktívne aj neaktívne rozhrania a ich adresy. Systém poskytuje jednoduché formuláre pre pridávanie či úpravu ethernet rozhraní (obr. 6.27) a IP adres (obr. 6.28).

Správa ETH serveru

Zpět

Přidat rozhraní

Informace o ethernet rozhraní: eth0

Mac: ac:1f:6b:2d:e7:ee







Stav: UP

Aktivní: ano

Datum zneaktivnění: -

Datum vložení: 22.03.2016

Datum kontroly: 13.04.2019

Adresy	Akce	IP/maska	Čas vložení	Aktivní
	✖ 	147.229.8.41/24	22.03.2016	ano
	✖ 	2001:67c:1220:808::93e5:829/64	22.03.2016	ano
	✖ 	fe80::230:48ff:fec8:74a4/64	22.03.2016	ne
	✖ 	fe80::ec4:7aff:feaa:d14e/64	15.06.2016	ne
	✖ 	2001:67c:1220:808:ec4:7aff:feaa:d14e/64	13.07.2016	ne
	✖ 	2001:67c:1220:808:ae1f:6bff:fe2d:e7ee/64	19.12.2017	ne
	✖ 	fe80::ae1f:6bff:fe2d:e7ee/64	19.12.2017	ano

Smazat rozhraní

Upravit rozhraní

Přidat IP adresu

Obr. 6.26: Manažment ethernet rozhraní

Vložení nového ETH rozhraní

Zadejte údaje o novém ETH rozhraní:

Název: *

MAC: *

Stav: *

Aktivita:

Obr. 6.27: Pridanie/úprava ethernet rozhrana

Vložení nové ETH IP adresy

Zadejte údaje o nové IP adrese:

IP: *

Maska: *

Aktivita:

Obr. 6.28: Pridanie/úprava IP adresy

ETH rozhraní

Rozhraní	MAC	Stav	IP adresy	Datum kontroly	Datum vložení	Aktivní	Datum zneaktivnění
eth0	00:25:90:c0:d4:b0	DOWN		22.04.2019	15.07.2013	ano	-

Body připojení

Cesta	Zařízení	Obsazená kapacita	Obsazené i-uzly	Datum kontroly	Datum vložení
/mnt/data-in	data--in-minerva1--in	21.08 TB / 26 TB (81.09%)	21522128 / 19227991780 (0.11%)	22.04.2019	28.01.2019

Disky

Sériové číslo	Model	Zařízení	Kapacita	Body připojení	Obsazená kapacita dle b. připojení	LVM skupina	Datum kontroly	Datum vložení
WD-WMC1T1179992	WDC WD30EFRX-68AX9N0	sda	3 TB	/ (sda1) /tmp (sda2)	65.02 GB / 1.48 TB (4.4%)	-	22.04.2019	15.07.2013

RAID HW

Model	Adresa	Raid level	Zařízení	Kapacita	Body připojení	Obsazená kapacita dle b. připojení	Stav	LVM skupina	Datum kontroly	Datum vložení
9750-16i4e	c0lu0	RAID-6	sdb	26 TB	/mnt/data-in (data--in-minerva1--in)	21.08 TB / 26 TB (81.09%)	OK	data-in	22.04.2019	18.07.2013

LVM svazky

UUID	Název	Zařízení	Kapacita	Body připojení	Obsazená kapacita	Stav	LVM skupina	Datum kontroly	Datum vložení
YdbyMu-IqEI-9eEs-HV8p-jPo4-wDO0-1PM1mc	minerva1-in	data--in-minerva1--in	25.29 TB	/mnt/data-in (data--in-minerva1--in) 26 TB	21.08 TB / 26 TB (81.09%)	available	data-in	22.04.2019	28.01.2019

LVM skupiny

UUID	Název	Kapacita	Stav	Datum kontroly	Datum vložení
z6HoHG-yAST-9eC3-yCY-77ev-0bm4-gykSlr	data-in	25.29 TB	resizable	2019.04.22 17:18:42	2019-01-28 02:49:45

Obr. 6.29: Detail serveru

Kapitola 7

Implementácia modulu monitoringu systémových záznamov

V tejto kapitole je popísaná implementácia modulu pre monitorovanie systémových záznamov, ktorý pozostáva z pravidelnej analýzy definovaných súborov a z užívateľského rozhrania.

Podobne ako modul pre monitorovanie úložísk, aj tento modul je nasadený v produkčnom prostredí a implementácia môže mať isté odlišnosti od návrhu.

7.1 Migrácia dát

Skript `knotis/ostatni_skripty/monitoring_logov_migrace_dat.php` je možné spúšťať opakovane podobne ako skript migrujúci monitoring úložísk. Na začiatku sú vymazané (ak existujú) využívané entity a vytvorené nanovo. Následne sú podľa návrhu migrované entity definujúce súbory systémových záznamov. Počas migrácie sú vytvárané typy súborov – entita `server_log` obsahujúca všeobecné nastavenia.

7.2 Analýza systémových záznamov

Práca prináša 3 skripty umiestnené na strane klienta (monitorovaného počítača):

- `log-monitoring.sh` – spúšťaný cron úlohou každú hodinu,
- `log-monitoring.py`,
- `log_monitoring_update_crontabs.sh` – prepínačom `-c` pridá cron úlohu pre spustenie bash skriptu každú hodinu náhodne v 7.–50. minúte.

Podľa návrhu aj tento modul zabezpečuje monitorovanie ethernet rozhraní, keďže táto funkcionality môže byť zdieľaná, a v prípade problému je správca upozornený skôr. Prvý skript tieto dáta zbiera a python skript ich transformuje do rovnakej štruktúry formátu JSON ako v prípade modulu pre monitoring úložísk. Na začiatku bash skript pošle žiadosť o definície súborov a ich pravidiel na danom servery, čo na strane systému KNOTIS obslúži skript `monitoring-logov-server-data.php`.

Skript pre analýzu logov otvorí súbor s definíciami, načíta ho a každý súbor systémových záznamov analyzuje nasledujúcim spôsobom:

1. ak súbor neexistuje, zapíše chybu a pokračuje definíciou ďalšieho súboru,
2. ak súbor nebol modifikovaný od poslednej analýzy, pokračuje ďalšou definíciou,
3. analyzuje súbor:
 - (a) ak prvý záznam v súbore už bol spracovaný počas predchádzajúcej analýzy (teda nenastala rotácia), preskočí sa X bajtov, kde X je veľkosť súboru zistená prechádzajúcou analýzou,
 - (b) pre každý záznam (riadok) sa zistí čas záznamu, a či už bol analyzovaný, a ak áno, pokračuje ďalším záznamom,
 - (c) ak záznam nebol analyzovaný, aplikuje pravidlá:
 - i. ak vyhovuje reportovaciemu pravidlu, udalosť je zaznamenaná a veľkosť záznamu je pripočítaná do nárastu pred a po filtrácii, a to aj v prípade, ak záznam vyhovuje nejakému ignorovaciemu pravidlu,
 - ii. ak vyhovuje ignorovaciemu pravidlu, veľkosť záznamu je pripočítaná iba do nárastu pred filtráciou,
 - (d) ak záznam nevyhovuje žiadnemu pravidlu, veľkosť je pridaná do nárastu pred aj po filtrácii,
4. ak existuje prvý rotovaný súbor a bol modifikovaný od prvej analýzy, je analyzovaný rovnakým spôsobom,
5. spočíta veľkosť súboru a rotovaných (aj komprimovaných) súborov,
6. v prípade, ak nefiltrovaný nárast prekročil maximálny možný nárast, analyzátor získa výstup z logwatch (ak neexistuje) a uloží ho do JSON,
7. podľa návrhu vytvorí n-ticu pre daný súbor a uloží ju do JSON.

Ak reportovacie pravidlo vyhovuje nejakému záznamu, správca má byť upozornený aj na konkrétne záznamy. Nesmie ich však byť príliš veľa, aby sa v reporte nestrácal. Je teda nastavené, že report bude obsahovať max. 10 unikátnych príkladov pre každé jedno pravidlo (ktorému vyhovuje aspoň jeden záznam). Unikátnosť spočíva v tom, že záznamy sa líšia v obsahu, nie v čase. V reporte teda môže byť napríklad **příklad: 12:51:51 system call /auth/xyz.sh počet hitů: 12**. To znamená, že **##### system call /auth/xyz.sh** sa nachádza v súbore 12x. Aby správca mohol v danom súbore konkrétny string vyhľadať, je pre uvedený príklad v reporte čas zachovaný.

Pre pravidlá v skripte v jazyku python existuje trieda Rule, ktorú majú pravidlá rozširovať. Tiež je vytvorená trieda RuleFactory, ktorá podľa číselného typu pravidla určia, ktorú podtriedu triedy Rule má vytvoriť. V prípade, ak správca bude chcieť vytvoriť nové pravidlo, stačí vytvoriť novú triedu, ktorá toto pravidlo bude reprezentovať, a aktualizovať metódu get v triede RuleFactory. Trieda pre pravidlo musí implementovať metódy `is_ignore` a `match_line`. Prvá informuje, či pravidlo je ignorovacie alebo reportovacie, a druhá, či vyhovuje záznamu.

Spracovanie dát

Spracovanie zabezpečuje skript `monitoring-logov-spracovani.php`, ktorý autentizuje požiadavok a spracuje ethernet rozhrania rovnakým spôsobom ako monitoring úložísk. Následne podľa návrhu uloží nový záznam, ak došlo k nárastu alebo ak nefiltrovaný nárast prekročil maximálne povolenú hodnotu. Ak nejaké reportovacie pravidlo bolo úspešne aplikované, vytvorí udalosť a v e-maile pošle aj obsah výstupu aplikácie logwatch. Správca je informovaný o nárastoch, o jednotlivých úspešne aplikovaných pravidlách a o spomínanom výstupe logwatch.

7.3 Uživatelské rozhranie

Pre konkrétny server systém ponúka grafické zobrazenie nárastu a celkovej veľkosti jednotlivých súborov systémových záznamov. To zobrazuje obrázok 7.1. K tomuto pohľadu vedie odkaz z pohľadu pre detail serveru a zo zoznamu všetkých monitorovaných súborov v systéme (obr. 7.2). V ňom má správca možnosť vybrať konkrétne servere, typy súborov a typy grafov. V tomto pohľade je možné mazať jednotlivé definície súborov, čím prestanú byť dané súbory monitorované.

Vytvorenie novej, či úpravu existujúcej definície súboru pre server je možné vykonať vo formulári zobrazenom na obr. 7.3. V tomto pohľade sú odkazy na úpravu či vytvorenie nového typu súboru (obr. 7.4).

Systém ponúka zoznam existujúcich pravidiel v prehľadnej tabuľke, kde má správca možnosť pridávať nové, či upravovať a mazať existujúce (obr. 7.5). Vytváranie nového pravidla, či úpravu existujúceho umožňuje samostatný formulár (obr. 7.6). Systém ponúka aj jednoduchý pohľad pre mazanie starých záznamov (obr. 7.7).

V tabuľke 7.1 je možné nájsť, ktoré súbory implementujú ktoré navrhnuté pohľady.

suffix názvu súboru	návrh	URL	str	obr.
log_uprav.php	B.28	140		7.4
server_detail_logy.php	B.30	128		7.1
server_log_mazani_zaznamu.php	B.29	146		7.7
server_log_pravidlo_uprav.php	B.25	141		7.6
server_log_pravidlo_zoznam.php	B.24	142		7.5
server_log_uprav.php	B.27	139		7.3
server_log_zoznam.php	B.26	138		7.2

6.18

Tabuľka 7.1: Zoznam súborov, ktoré implementujú jednotlivé wireframe návrhy

Seznam logů

? Návod

Výber logu ☐ vše
☐ access.log ☒ auth.log ☐ error.log ☐ kern.log ☐ ssl_access.log ☐ syslog

Výber serveru ☒ vše

Výběr grafu ☒ nárůst
☐ velikost

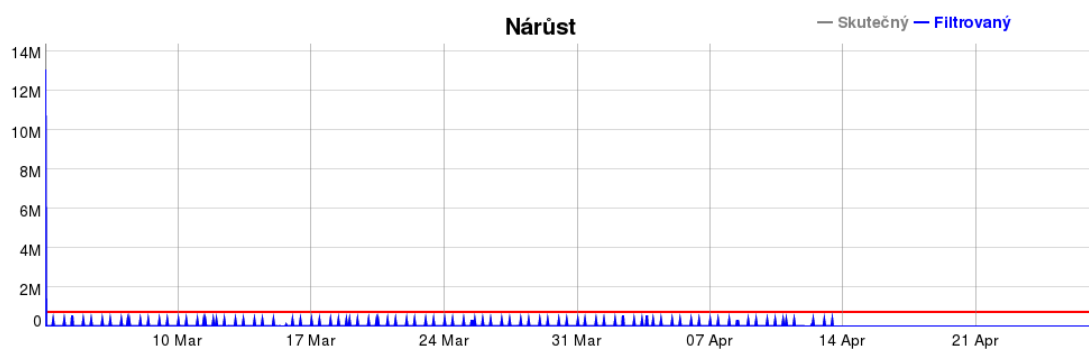
Potvrdit Zpět

athena1

přidat definici logu detail serveru detail serveru - logy

/var/log/auth.log

Max nárůst: 800 KB nastavení smazat



athena10

přidat definici logu detail serveru detail serveru - logy

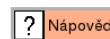
/var/log/auth.log

Max nárůst: 700 KB nastavení smazat



Obr. 7.1: Detail serveru pre systémové záznamy

Detail logů pro server



Informace o serveru:

Název: athena1

Doména: fit.vutbr.cz

Autentizace: athena1_3cCimVRx2qD6itdgztKRYV3U64WUJFrQfSEBnwUS7PUgETIIXh

IPv4 brána: 147.229.8.1 via eth2

IPv6 brána: fe80::204:96ff:fe1d:4e30 via eth2

Datum vložení: 10.07.2013

Datum kontroly: 13.04.2019

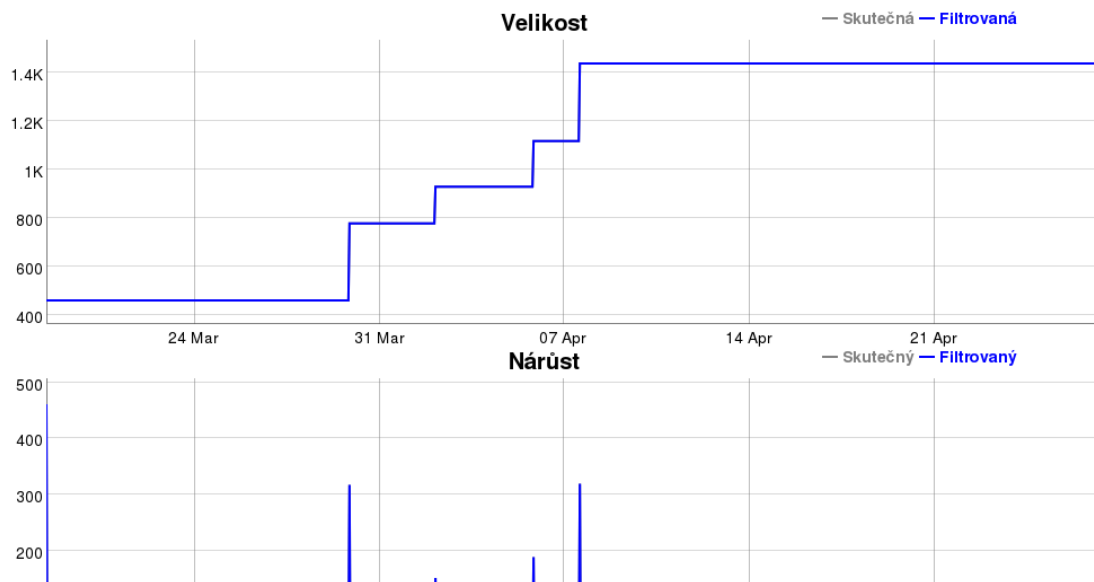
[Zpět](#)[přidat definici logu](#)[vytvořit pravidlo](#)

/var/log/apache2/access.log

Max nárůst: 400 KB [nastavení](#) pravidla —

žádné pravidla

[Ignoruj chage](#)

[Přidat pro log](#)[upravit](#)

Obr. 7.2: Zoznam súborov systémových záznamov

Přidání definice logu pro server

Zadejte definici monitorovaného logu pro: athena1

Typ logu:

Cesta: ☐ výchozí

Datetime regex: ☐ výchozí

Datetime format: ☐ výchozí

Datetime regex 2: ☐ výchozí

Datetime format 2: ☐ výchozí

Max nárůst (B): ☒ výchozí

Info: Datetime regex 2 a datetime format 2 jsou alternativy, pokud čas nelze získat.
Regex bude zkompilován pomocí [re.compile](#)
Kandidáti (řetězce s časem) budou získáni pomocí [Pattern.findall](#).
Formát času: [formát](#).
Z prvního kandidáta, pro který bude možno skonstruovat objekt datetime se
zadaným formátem času, bude získána časová známka.

Obr. 7.3: Přidání/úprava definice pro soubor systémových záznamů

Přidání výchozí definice pro log

Zadejte všeobecnou definici monitorovaného logu (typ logu):

Název: *

Výchozí cesta: *

Výchozí datetime regex: *

Výchozí datetime format: *

Výchozí datetime regex 2:

Výchozí datetime format 2:

Výchozí max nárůst: *

Obr. 7.4: Přidání/úprava typu souboru systémových záznamů

Seznam server log pravidel

vytvořit pravidlo

Nastavit filtr

Vymazat filtr

Uložit filtr

+

Přihlášen: I

?

Nápověda

Osobní nastavení

Odhlášení

Název	Typ	Alce	Servery	Typy logů	Název parametru	Hodnota parametru	Platnost
Ignoruj chage	<div> <div>-</div> <div>↑</div> <div>↓</div> </div>	<div> <div>✖</div> <div>✎</div> </div>	<div> <div>-</div> <div>↑</div> <div>↓</div> </div>	<div> <div>-</div> <div>↑</div> <div>↓</div> </div>			<div> <div>-</div> <div>↑</div> <div>↓</div> </div>
	ignoruj dle reg. výrazu		všechny	access log	regex	.*usr/bin/chage.*	neomezeně - neomezeně

Obr. 7.5: Zoznam pravidiel pre systémové záznamy

Přidání pravidla pro server logy

Zadejte definici pravidla pro server log:

Název: *

Typ:

Platnost od: do:

Logy ☐ vše
☐ access.log ☐ auth.log ☐ error.log ☐ kern.log ☐ ssl_access.log ☐ syslog

Servery ☒ vše

reulární výraz: *

Info: regex bude zkompilován pomocí [re.compile\(\)](#) s příznakem RE.I a záznam bude ignorován, pokud pro záznam funkce [match\(\)](#) vrátí TRUE

Obr. 7.6: Pridanie/úprava pravidla pre systémových záznamov

Mazání server log záznamů

Smazání záznamů starších než:

dub

2019

květen 2019

po	út	st	čt	pá	so	ne	po	út	st	čt	pá	so	ne
1	2	3	4	5	6	7			1	2	3	4	5
8	9	10	11	12	13	14	6	7	8	9	10	11	12
15	16	17	18	19	20	21	13	14	15	16	17	18	19
22	23	24	25	26	27	28	20	21	22	23	24	25	26
29	30						27	28	29	30	31		

Nyní

Zavřít

Obr. 7.7: Mazanie záznamov súborov systémových záznamov

Kapitola 8

Testovanie a vyhodnotenie

Riešenie zo začiatku testoval sám autor bežnou kontrolou dát zobrazovaných užívateľským rozhraním a kontrolou spracovania testovacích dát. Po odstránení objavených chýb bolo riešenie nasadené do ostrej prevádzky skupiny KNOT, ktorá v súčasnosti využíva 508 aktívnych diskov, z ktorých je 418 v diskovom poli. HW diskových polí je 17, z ktorých štyri využívajú cache jednotku spolu so záložným zdrojom energie (batéria a kondenzátor). Softvérových diskových polí je 58. Na serveri minerva1 je využívaná technológia LVM.

Po nasadení boli odstránené posledné chyby a vytvorené rôzne vylepšenia. Medzi ne patrí vytvorenie konceptu definovania vlastných SMART skratiek a pridanie skratiek do pohľadu pre detail disku a batérie. Výpisy boli rôzne prispôbené z dôvodu lepšej čitateľnosti. Napr. zúženia stĺpcov širokých tabuliek (zoznam diskov, diskových polí, ...).

Ostrá prevádzka však odhalila aj menšie či väčšie chyby. Napríklad systém bol schopný detekovať nový disk v SW diskovom poli, avšak ak disk nahradil konkrétny vyradený disk, hoc systém vyradený disk nezobrazoval, počas spracovania aktualizoval stav práve tomuto už vyradenému disku namiesto novému. Navyše disk nebol priradený k danému diskovému polu. Chyba bola spôsobená tým, že v SQL príkaze chýbala podmienka na aktivnosť disku, a teda vybraný bol vždy ten starý. Je však potrebné dodať, že SMART atribúty nového disku boli stále sledované.

Ďalej sa ukázali rôzne malé chyby v hláseniach, ktoré boli najčastejšie spôsobené volaním správnych metód so zlými parametrami. Ostatné menšie či väčšie chyby boli tiež odstránené.

V súčasnosti moduly pre monitoring úložísk a systémových záznamov nevykazujú žiadne chyby. Architektúra zapojenia je zobrazovaná a aktualizovaná správne a dáta sú prehľadne zobrazované a automatizicky spracovávané. Po otestovaní a stabilizovaní riešenia je teda možné vyhodnotiť cieľ zníženia réžie. To je docielené niekoľkými spôsobmi.

Prvým je filtrácia vstupných dát na strane klienta, ktoré sú posielané na spracovanie. Pre modul úložísk bolo 21.04.2019 vykonané meranie veľkosti súborov odosielaných na server pre obe verzie modulov. V danom čase existovalo 71 serverov. Priemer veľkostí súborov pre starú verziu je 55.69KB a pre novú 23.57KB, čo predstavuje pokles o 57,48 %. Modul pre monitoring systémových záznamov v súčasnosti nemá zmysel takto vyhodnocovať z dôvodu, že ešte neboli vytvorené pravidlá, ktoré by odfiltrovali najčastejšiu aktivitu zamestnancov či študentov. V súčasnosti je teda objem prijímaných a odosielaných dát väčší, pretože je odosielaných viac informácií o sledovaných súboroch spolu s výstupom aplikácie logwatch, ktorý využíva aj pôvodná verzia.

Druhým je spracovanie dát. Spracovanie dát o úložiskách zo serveru s najvyšším počtom diskov a zariadení minerva1 trvá priemerne 9 sekúnd novej verzii. Pre pôvodnú je to priemerne 40 sekúnd. Čo sa týka monitoringu logov, tak spracovanie pre každý súbor

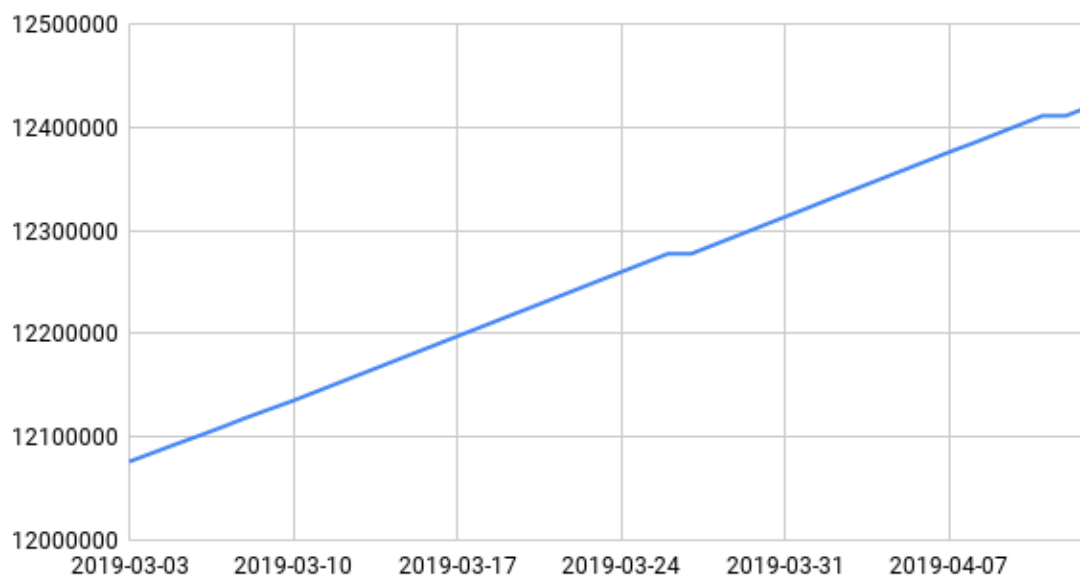
obnáša zopár jednoduchých SQL príkazov: 1 príkaz INSERT pre prípadné uloženie nového záznamu, 1 príkaz UPDATE pre aktualizovanie metadát súboru a 1 dlhší príkaz SELECT, kde je síce 1 podmienka na zhodu reťazcov, avšak klauzuly JOIN sú nad indexovanými primárnymi kľúčmi. Teda opäť nemá zmysel meranie a porovnávanie, pretože záťaž je minimálna.

Ďalším je vizualizácia dát. Stará verzia tiež ponúka možnosť zobrazenia zoznamu problémových diskov. To je však implementované tak, že kým sa načítajú a vyhodnotia dáta, ubehne bezmála 20 sekúnd. Zoznam poskytnutý novou verziou je zobrazený prakticky ihneď. Správca v ňom navyše vidí presnú príčinu a má možnosť manipulovať s parametrami sledovania na mieste. Správca má možnosť zobrazenia dát v grafe, čo stará verzia neumožňovala. Čo sa týka monitoringu systémových záznamov, tiež je poskytované grafické zobrazenie veľkosti súborov v čase, čo predtým nebolo. Žiaľ, ak správca chce zobrazíť grafy pre všetky súbory, knižnici Dygraph trvá dlhšie zobrazíť také veľké množstvo grafov na jednej stránke. Pre 71 serverov, priemerne 4 súbory na jeden a pre každý 2 grafy, dohromady 568 grafov, to trvá zhruba dve desiatky sekúnd. Z toho dôvodu má správca k dispozícii filtrovanie, teda zobrazíť iba vybrané servery, typy grafov a typy súborov, čím sa docieli rýchle zobrazenie grafov.

Posledný spôsob je úspora množstva ukladania nových riadkov pre SMART atribúty do databázy (tabuľka `smart_attr_zaznam`). To je dosiahnuté tým, že je ich možné ukladať iba vtedy, keď nastala zmena. Meranie prebehlo od 03.03.2019 do 13.04.2019. Na začiatku existuje 12 076 058 záznamov v starej verzii a 1 776 927 záznamov v novej. Rozdiel je 10 299 131 záznamov. Na konci obdobia v starej verzii existuje 12 419 916 záznamov a v novej 1 830 687 a rozdiel je 10 589 229 záznamov. Keď sa pozrieme na nárast rozdielu (311 115), je teda možné predpokladať, že stará verzia by bola viac a viac neefektívna. Celý priebeh merania zachytáva graf 8.1 a 8.2. Grafy sú dva, aby bol rozdiel nárastu zreteľnejší. Rozsah je v oboch grafoch 500 000.

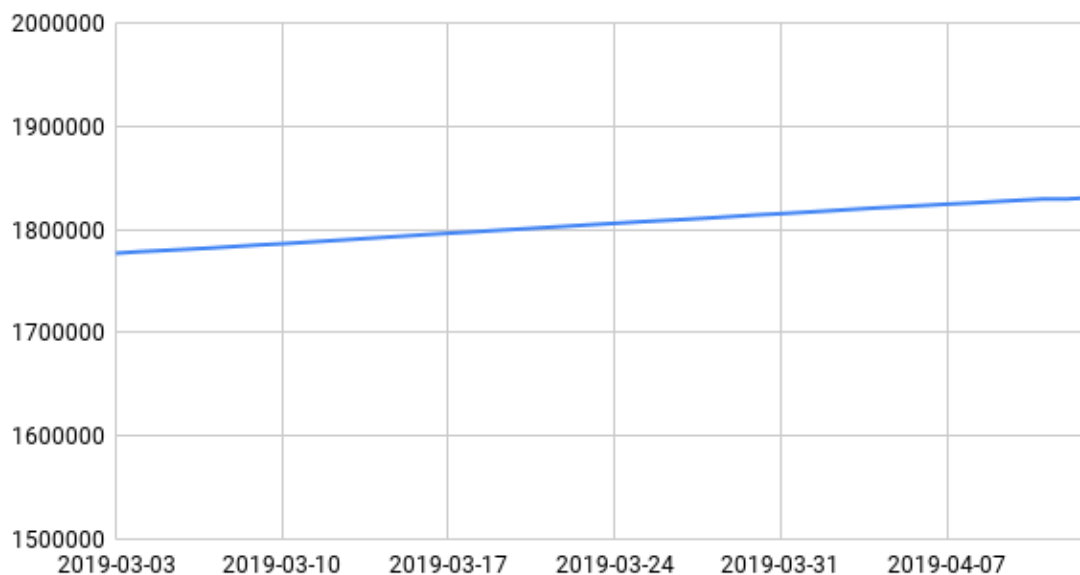
Pre monitoring systémových záznamov to je podobne (tabuľka `server_log_zaznam`). Rozdiely však skúmať nemožno, nakoľko staré záznamy neboli migrované. V grafoch 8.3 a 8.4 vidno, že nárast v novej verzii je menší. Pred dňom 03.03.2019 záznamy neexistovali a 13.04.2019 existovalo 113 955 záznamov. Počet záznamov starého monitoringu sa zvýšil za rovnaké obdobie o 318 156, čo je skoro trojnásobne viac.

Počet SMART záznamov v DB - stará verzia



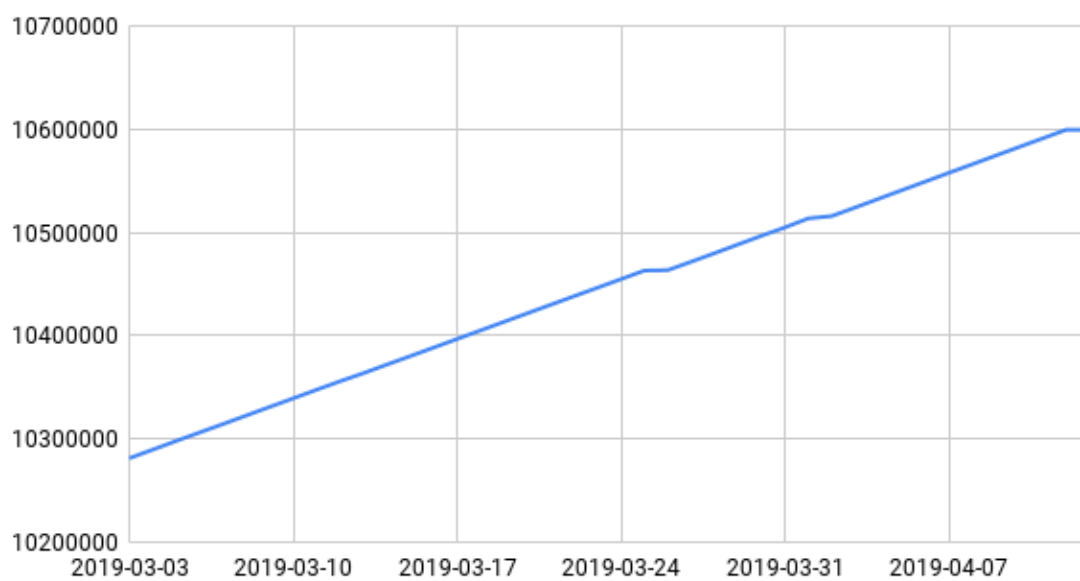
Obr. 8.1: Počet SMART záznamov – stará verzia

Počet SMART záznamov v DB - nová verzia



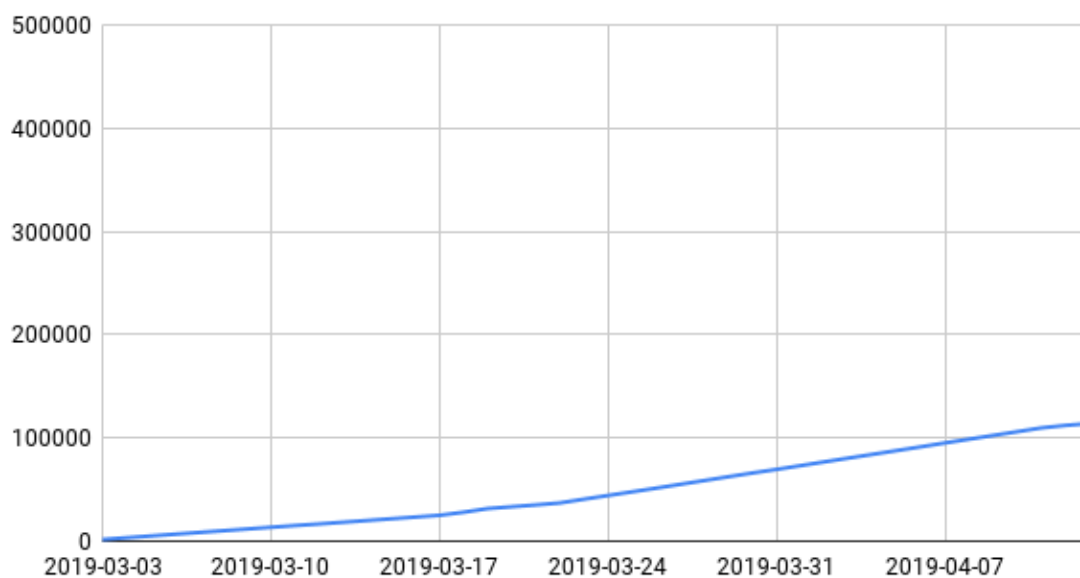
Obr. 8.2: Počet SMART záznamov – nová verzia

Počet záznamov pre monitoring logov - stará verzia



Obr. 8.3: Počet záznamov pre monitoring systémových záznamov – stará verzia

Počet záznamov pre monitoring logov - nová verzia



Obr. 8.4: Počet záznamov pre monitoring systémových záznamov – nová verzia

Kapitola 9

Záver

Cieľom práce bol návrh a implementácia modulov do informačného systému KNOTIS, ktoré sa zaoberajú monitoringom úložísk a systémových záznamov. V rámci práce som podrobil kritike súčasný stav, z ktorého vyplynulo niekoľko nedostatkov. Napríklad databázový model nereprezentuje realitu, monitorovanie systémových záznamov sleduje iba zmeny veľkosti súborov a v rámci modulu pre monitoring úložísk má správca sťaženú prácu so systémom.

V práci som definoval ciele a požiadavky na nové moduly a popísal technológie, ktoré sú využité pre riešenie. Následne som navrhol a implementoval nové užívateľské rozhranie, ktoré poskytuje viac možností pre nastavenia sledovania, prehľadnejšie zobrazuje informácie a umožňuje rýchlejšiu prácu s parametrami sledovania. ER diagram reprezentuje realitu korektnými vzťahmi medzi jednotlivými entitami a štruktúra databázy umožňuje efektívne ukladanie dát. Práca prináša skript zabezpečujúci automatický zber informácií a automatickú detekciu nových zariadení. Dáta sú filtrované, čím sa odľahčí sieťová záťaž oproti súčasnému stavu.

Pre monitorovanie systémových záznamov som navrhol pravidlá, ktoré môžu vyustiť do upozornenia správcu alebo ignorovania záznamu. Pravidlá môžu byť aplikované buď v informačnom systéme po obdržaní vstupu alebo priamo na serveri, kde sa pracuje so súborami systémových záznamov. Práca prináša možnosť definovať súbory systémových záznamov, ktoré majú byť pravidelne každú hodinu monitorované. Na strane klienta sa nachádza skript, ktorý vždy získa informácie o súboroch a pravidlách a následne zabezpečí vykonanie analýzy.

Pre zistenie stavu diskových jednotiek práca využíva technológiu SMART. V prípade, že sa zhoršuje stav vzhľadom na hodnoty parametrov sledovania, systém upozorní správcu na možnú potrebu výmeny jednotky.

Porovnanie s pôvodným riešením ukázalo v prípade modulu pre monitorovanie úložísk niekoľko zlepšení. Medzi ne patrí úspora záťaže siete o viac než 56 %, rýchlejšie spracovanie monitorovaných dát, a to približne troj až štvornásobne, rýchlejšie zobrazenie zoznamu možných problémov s diskami a nakoniec o rád nižšia náročnosť na počet záznamov v databáze.

Možnosti monitoringu systémových záznamov ešte nie sú využívané naplno, teda ešte nie sú vytvorené pravidlá pre efektívnu filtráciu známych činností. Aj tak sa však modul ukázal byť lepší v náročnosti na databázu. Po viac než mesiaci fungovania uložil približne trikrát menej záznamov ako pôvodné riešenie, pričom úspora v množstve ukladaných dát v oboch moduloch nie je na úkor množstva informácií.

Po zhodnotení a ohliadnutí sa späť je na mieste otázka ako ďalej. Kam možno riešenie posunúť? Mohlo by byť zaujímavé vytvoriť model, ktorý by s využitím strojového učenia automaticky nastavoval parametre sledovania pre jednotlivé disky.

Ďalej by mohlo byť prínosné vytvoriť nástroj, ktorý by automaticky generoval pravidlá pre ignorovanie systémových záznamov. Vstupom by boli jednotlivé záznamy pre ignorovanie zadané správcom. A nakoniec aj tu by mohlo stať za uvaženie využiť strojové učenie. Správca by zadával aj podozrivé záznamy, a keďže by boli k dispozícii obe triedy (podozrivá a bežná aktivita), postupne by mohol byť trénovaný klasifikátor aktivity. Samozrejme to predpokladá veľké množstvo dát a pravdepodobne aj vytvorenia ďalších typov pravidiel vhodných pre takýto účel.

Literatúra

- [1] Allen, B.: Monitoring Hard Disks with SMART. *Linux Journal*, ročník 2004, č. 117, 2004: s. 74–77, ISSN 1075-3583.
URL <http://www.linuxjournal.com/magazine/monitoring-hard-disks-smart?page=0,0>
- [2] AMCC, Sunnyvale: *CLI Guide*. 2006, [Online; navštívené 20.12.2018].
URL <https://www.digiliant.com/docs/3ware9650SE-UsrGuide.pdf>
- [3] Chaffer, J.; Swedberg, K.; Baše, K.; aj.: *Mistrovství v jQuery*. Brno: Computer Press, první vydání, 2013, ISBN 978-80-251-4103-8.
- [4] HTML & CSS. MIT, ERCIM, Keio, Beihang, 2016, [Online; navštívené 03.12.2018].
URL <https://www.w3.org/standards/webdesign/htmlcss>
- [5] Dijkstra, F.: Logical Interface Names. 2014, [Online; navštívené 04.12.2018].
URL http://www.macfreek.nl/memory/Logical_Interface_Names
- [6] Evans, M.: Self-Monitoring, Analysis and Reporting Technology (S.M.A.R.T.). 1996, [Online; navštívené 04.12.2018].
URL ftp://ftp.linux-mips.org/pub/linux/mips/people/macro/S.M.A.R.T./8035R2_0.PDF
- [7] Faulkner, S.; Eicholz, A.; Leithead, T.; aj.: HTML 5.1 2nd Edition. W3C ®, 2017, [Online; navštívené 10.12.2018].
URL <https://www.w3.org/TR/html5/>
- [8] GNU coding standards. 2013, [Online; navštívené 13.12.2018].
URL <https://www.gnu.org/prep/standards/>
- [9] Get S.M.A.R.T. for Reliability, 1999, [Online; navštívené 02.12.2018].
URL http://argusmonitor.com/help/enhanced_smart.pdf
- [10] Gilmore, W. J.; Pokorný, J.: *Velká kniha PHP 5 a MySQL*. Brno: Zoner Press, nové, 3. vyd. vydání, 2011, ISBN 978-80-7413-163-9.
- [11] Gkioka, I.; Jahoda, M.; Heves, J.; aj.: Red Hat Enterprise Linux 7. 2017, [Online; navštívené 04.12.2018].
URL https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/pdf/networking_guide/Red_Hat_Enterprise_Linux-7-Networking_Guide-en-US.pdf

- [12] Hruška, T.: Pojem informačního systému, 2017, [Online; navštívené 13.12.2018].
URL <https://www.fit.vutbr.cz/study/courses/WAP/private/podklady/Prednasky/IIS/IIS010Pojem.pdf>
- [13] Hughes, G.; Murray, J.; Kreutz-Delgado, K.; aj.: Improved disk-drive failure warnings. *IEEE Transactions on Reliability*, ročník 51, č. 3, 2002: s. 350–357, ISSN 00189529, doi:10.1109/TR.2002.802886.
URL <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1028408>
- [14] Levine, S.: Red Hat Enterprise Linux 7. 2017, [Online; navštívené 01.12.2018].
URL https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/pdf/logical_volume_manager_administration/Red_Hat_Enterprise_Linux-7-Logical_Volume_Manager_Administration-en-US.pdf
- [15] LSI Corporation, San Jose: *StorCLI*. 2013 vydání, 2013, [Online; navštívené 05.04.2018].
URL <https://docs.broadcom.com/docs/12352476>
- [16] Mdadm(8) - Linux man page. [Online; navštívené 15.12.2018].
URL <https://linux.die.net/man/8/mdadm>
- [17] MySQL. Oracle Corporation, 2018, [Online; navštívené 15.12.2018].
URL www.mysql.com
- [18] Nzomo, J.: Linux network interface naming. 2015, [Online; navštívené 08.12.2018].
URL http://tdt.rocks/linux_network_interface_naming.html
- [19] Payment Card Industry (PCI) Data Security Standard. 2016, [Online; navštívené 16.12.2018].
URL https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2.pdf?agreement=true&time=1515969691490
- [20] Pinheiro, E.; Weber, W.-D.; Barroso, L. A.: Failure Trends in a Large Disk Drive Population. In *5th USENIX Conference on File and Storage Technologies (FAST 2007)*, 2007, s. 17–29, [Online; navštívené 07.12.2018].
URL https://research.google.com/archive/disk_failures.pdf
- [21] Písek, S.: *JavaScript*. Praha: Grada, první vydání, 2001, ISBN 80-247-0014-X.
- [22] Red Hat Enterprise Linux 5. 2013, [Online; navštívené 17.12.2018].
URL https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/ch-raid
- [23] S.M.A.R.T. Wikimedia Foundation, 2001, [Online; navštívené 11.12.2018].
URL https://en.wikipedia.org/wiki/S.M.A.R.T.#Standards_and_implementation
- [24] Smartmontools Documentation. [Online; navštívené 04.12.2018].
URL <https://www.smartmontools.org/wiki/TocDoc>
- [25] Usage statistics and market share of PHP for websites. 2018, [Online; navštívené 11.12.2018].
URL <https://w3techs.com/technologies/details/pl-php/all/all>

- [26] Žára, O.: *JavaScript*. Brno: Computer Press, první vydání, 2015, ISBN 978-80-251-4573-9.

Príloha A

Sledované atribúty batérie a kondenzátora HW diskového poľa

Atribút	Typ zariadenia
Voltage (stav batérie)	batéria
Current (stav batérie)	batéria
Temperature	oboje
State	oboje
Charging status	batéria
Voltage (stav firmware)	batéria
Temperature (stav firmware)	batéria
Learn Cycle Status	batéria
I2C Errors Detected	batéria
Battery Pack Missing	batéria
Replacement required	oboje
Remaining Capacity Low	batéria
Periodic Learn Required	batéria
No space to cache offload	oboje
Pack is about to fail & should be replaced	batéria
Module microcode update required	oboje
Fully Discharged	batéria
Fully Charged	batéria
Discharging	batéria
Initialized	batéria
Over Temperature	batéria
Over Charged	batéria
Charging Terminated	batéria
Relative State of Charge	batéria
Charging current	batéria
Absolute State of Charge	batéria
Max Error	batéria
Battery backup charge time	batéria
Pack Energy	kondenzátor
Capacitance	kondenzátor
Relative State of Charge	batéria
Absolute State of charge	batéria
Remaining Capacity	batéria
Full Charge Capacity	batéria
Run time to empty	batéria
Average time to empty	batéria
Average Time to full	batéria
Cycle Count	batéria

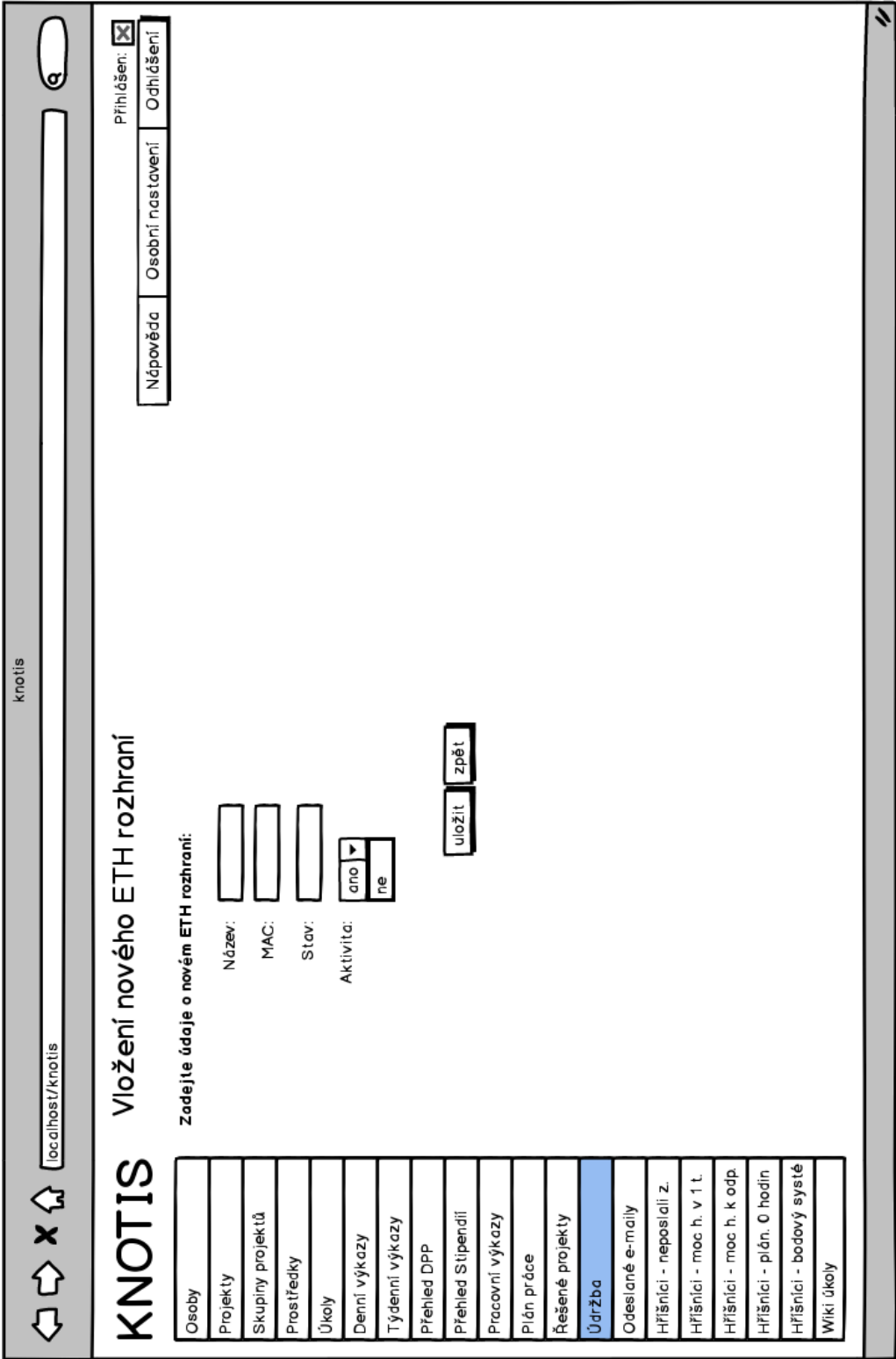
Tabuľka A.1: Zoznam sledovaných parametrov batérií a kondenzátorov

Príloha B

Obrázky wireframe

V tejto prílohe sa nachádzajú obrázky typu wireframe – návrh užívateľského rozhrania pre správcu.

Obr. B.1: Detail servera



Obr. B.3: Pridanie/úprava ethernet rozhraní servera

Obr. B.5: Zoznam diskov

seznam problemových disků

Obr. B.6: Zoznam problémových diskov



Obr. B.9: Detail disku

Obr. B.10: Nastavenie parametrov sledovania disku

Knotted

localhost/knotis

knotted

Osoby

Projekty

Skupiny projektů

Prostředky

Úkoly

Denní výkazy

Týdenní výkazy

Přehled DPP

Přehled Stipendií

Pracovní výkazy

Plán práce

Řešené projekty

Uživatel

Odeslané e-maily

Hříšníci - neposlali z.

Hříšníci - moc h. v 1 t.

Hříšníci - moc h. k odp.

Hříšníci - plán. 0 hodin

Hříšníci - bodový systém

Wiki úkoly

Model

Sériové číslo

Kapacita

Server

Raid

Stav

Bod připojení

Zařízení

Adresa

Datum vložení

Short test

Long test

deaktivovat

nastavení

zpět

Historie RAW hodnoty

1W

1M

13M

6M

1Y

ALL

RAW

hodnota

min

max

hodnota

Prům

Nejhorší

hodnota

Max

abs. změna

Max

relativní změna

Max

změna

nejhorší

hodnota

Selhala

Sledování

Sledování

změny

Nastavení

Typ	RAW	hodnota	min	max	hodnota	Prům	Nejhorší	hodnota	Max	abs. změna	Max	relativní změna	Max	změna	nejhorší	hodnota	Selhala	Sledování	Sledování	změny
Pre-fail	0			1	200	51	200		1				1				email		1	

Hodnoty

RAW	hodnota	hodnota	worst	selhala	kontrola
2	98	98	98	2016-10-14 02:21:03	
1	99	99	99	2016-10-13 02:21:03	
0	100	100	100	2016-09-14 02:21:03	

Přihlášen: X

Odhlášení

Nápověda

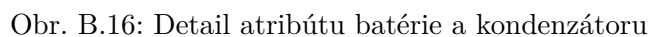
Osobní nastavení

Obr. B.12: Detail SMART atributu disku

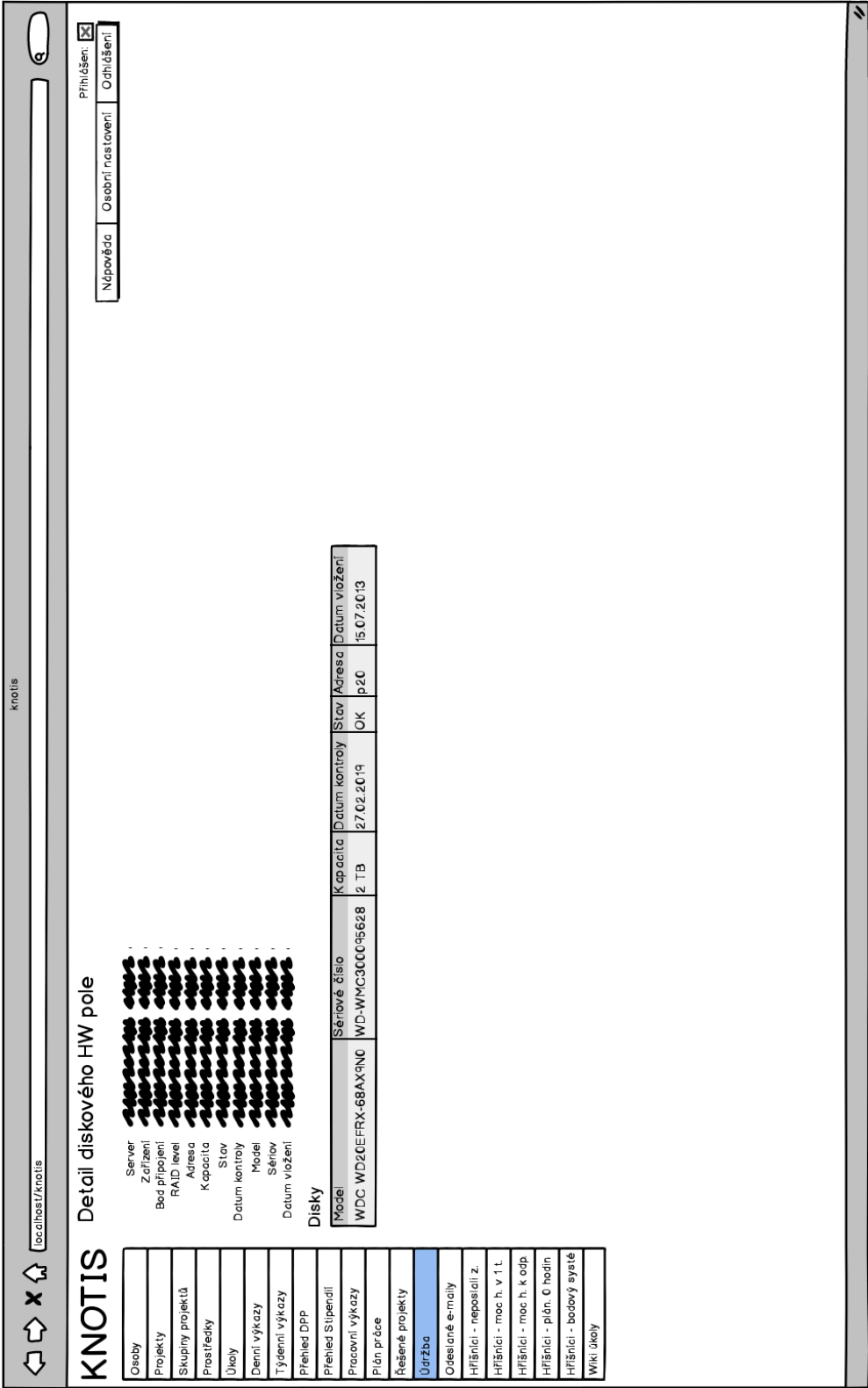
128

Obr. B.14: Zoznam batérií a kondenzátorov





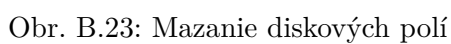
[illegible]



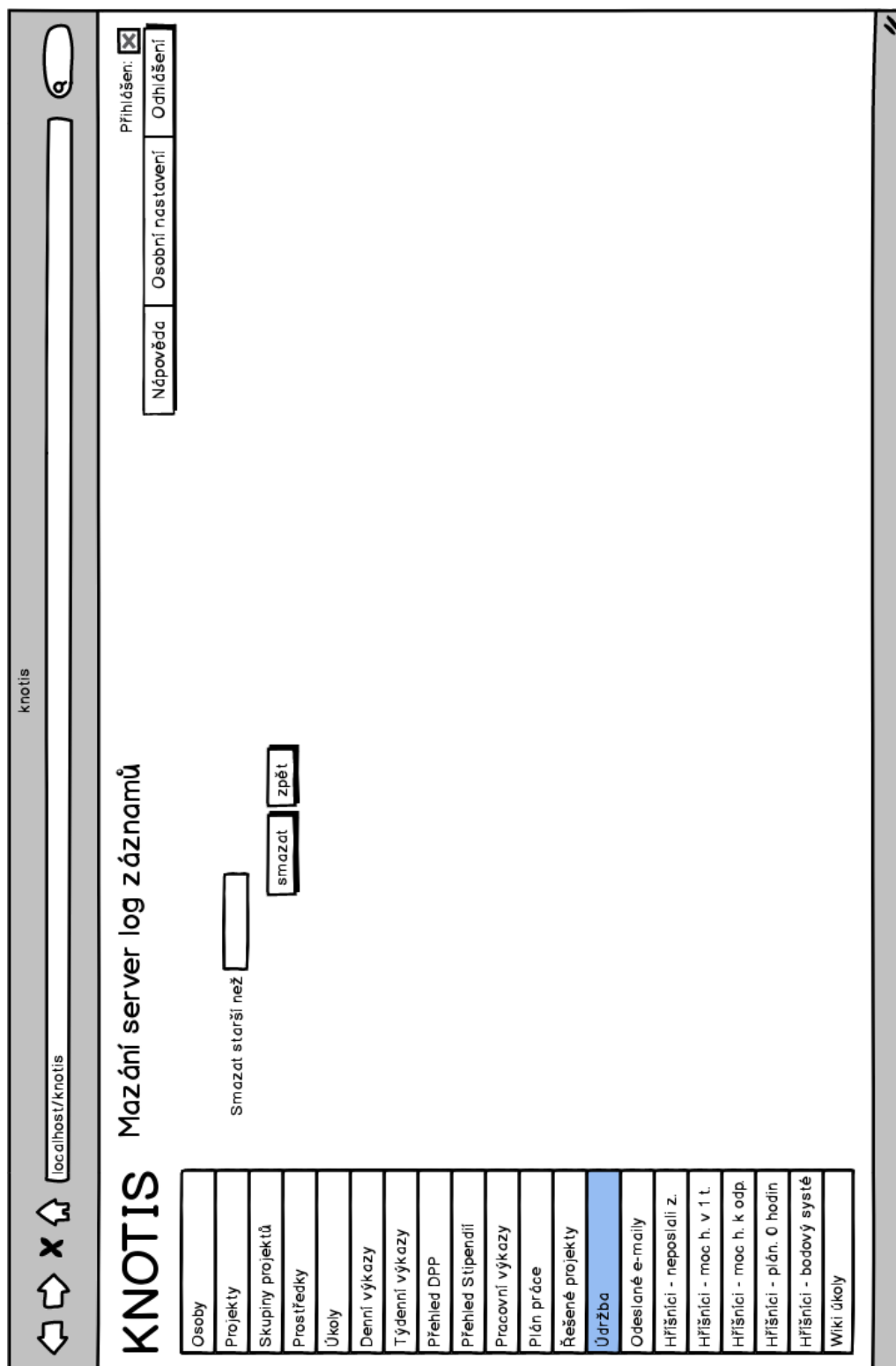
Obr. B.20: Detail HW diskového pola

[illegible]

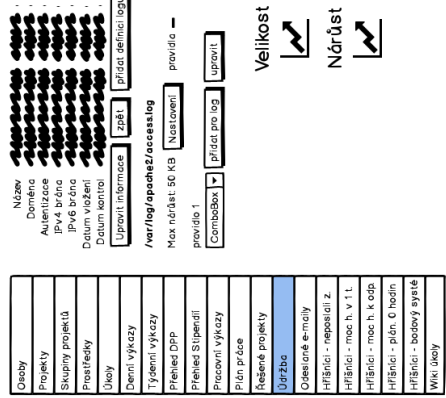




Obr. B.25: Vytvorenie/úprava pravidla pre systémové záznamy



Obr. B.29: Mazanie záznamov o systémových záznamov



146