



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

OPTIMALIZACE INFORMAČNÍHO SYSTÉMU PRO SLEDOVÁNÍ SPOTŘEB ENERGIÍ

OPTIMIZATION OF ENERGY CONSUMPTION MONITORING SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Martin Hrbek

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Fiedler, Ph.D.

BRNO 2019

Diplomová práce

magisterský navazující studijní obor **Kybernetika, automatizace a měření**
Ústav automatizace a měřicí techniky

Student: Bc. Martin Hrbek

ID: 138765

Ročník: 2

Akademický rok: 2018/19

NÁZEV TÉMATU:

Optimalizace informačního systému pro sledování spotřeb energií

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je optimalizovat a rozšířit aktuální informační systém pro sledování spotřeb energií.

- 1) Seznamte se s problematikou chytrých sítí a monitorováním spotřeby energií.
- 2) Seznamte se s aktuálním systémem firmy ModemTec pro sledování spotřeb energií.
- 3) Analyzujte nové požadavky na systém a navrhnete optimalizované řešení i pro zařízení s vyšší hustotou vzorkování.
- 4) Navrhnete nové možnosti pro prezentaci naměřených dat. Zaměřte se na vhodnou prezentaci velkého objemu naměřených dat a dlouhých časových úseků.
- 5) Vybraná navržená řešení implementujte a otestujte funkčnost a výkon na reálných nebo simulovaných datech.
- 6) Zhodnotte dosažené výsledky a diskutujte další možný vývoj tohoto systému.

DOPORUČENÁ LITERATURA:

1] Robin Nixon: Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 (Learning Php, Mysql, Javascript, Css & Html5) 4th Edition, O'REILLY, ISBN-13: 978-1491918661, ISBN-10: 1491918667

[2] Regina O. Obe, Leo S. Hsu: PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database 3rd Edition, O'REILLY, ISBN-13: 978-1491963418, ISBN-10: 1491963417

Termín zadání: 4.2.2019

Termín odevzdání: 13.5.2019

Vedoucí práce: doc. Ing. Petr Fiedler, Ph.D.

Konzultant:

doc. Ing. Václav Jirsík, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Fakulta elektrotechniky a komunikačních technologií, Vysoké učení technické v Brně / Technická 3058/10 / 616 00 / Brno

ABSTRAKT

Cílem práce je optimalizovat stávající systém pro sledování spotřeb energií a rozšířit možnosti prezentace naměřených dat tak, aby bylo možné přehledně zobrazovat velký objem dat a dlouhé časové úseky. Optimalizace se týká především zpracování naměřených dat a jejich prezentace.

KLÍČOVÁ SLOVA

Monitorování spotřeb energií, C++, PostgreSQL, Optimalizace, JavaScript, PHP, zpracování dat, prezentace dat

ABSTRACT

The goal of this thesis is to optimize the existing energy consumption monitoring system and expand the presentation options for the measured data in a way that would be suitable for large volume of data and a long period of time. Optimization concerns especially data processing and data presentation.

KEYWORDS

Data consumption monitoring, C++, PostgreSQL, Optimization, JavaScript, PHP, data processing, data presentation

Bibliografická citace:

HRBEK, Martin. *Optimalizace informačního systému pro sledování spotřeb energií* [online]. Brno, 2019 [cit. 2019-05-12]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/119306>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Petr Fiedler.

Prohlášení autora o původnosti díla

„Prohlašuji, že svou diplomovou práci na téma „Optimalizace systému pro monitorování spotřeb energií“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **10. května 2019**

.....
Jméno a příjmení autora, podpis

Poděkování

Děkuji vedoucímu diplomové práce panu Doc. Ing. Petru Fiedlerovi, Ph.D. za účinnou metodickou a pedagogickou pomoc a další cenné rady při zpracování mé diplomové práce.

Děkuji firmě ModemTec s.r.o. za umožnění vývoje tohoto systému.

Brno

.....

podpis autora(-ky)

Obsah

1	Chytré sítě a monitorování spotřeb energií	12
1.1	Výhody chytrých sítí	12
1.2	Monitorování spotřeb energií	13
1.3	Chytré měření a jeho komponenty	13
1.3.1	Chytré měřicí přístroje a chytré elektroměry	14
1.3.2	Vodoměry, plynoměry, měřiče tepla, kalorimetry a další	15
1.3.3	Počítače, LCD panely, chytré telefony	15
1.3.4	Data-koncentrátory	15
1.3.5	Servery	16
2	Systém firmy ModemTec pro monitorování spotřeb energií	17
2.1	Pro koho je systém určen	17
2.2	Rozdělení systému do modulů a jejich popis	17
2.2.1	Modul pro přihlášení uživatelů	18
2.2.2	Modul pro správce	18
2.2.3	Modul pro spotřebitele	18
2.2.4	Modul pro příjem SOAP zpráv z adaptéru chytré sítě	19
2.2.5	Modul pro vyhodnocení denních spotřeb z naměřených dat	19
2.3	Použité technologie	19
2.3.1	HTML5	20
2.3.2	CSS	20
2.3.3	PHP	21
2.3.4	Nette	21
2.3.5	JavaScript	21
2.3.6	jQuery	22
2.3.7	PostgreSQL	22
2.3.8	Cron	22
2.3.9	Ubuntu	23
2.3.10	Apache HTTP Server	23
2.3.11	Webové služby a SOAP protokol	23
2.3.12	Composer	24
3	Analýza nových požadavků na systém pro monitorování spotřeb energií	25
3.1	Nové požadavky zadané firmou ModemTec	25
3.1.1	Značné zrychlení celého systému (optimalizace)	25
3.1.2	Nové možnosti prezentace dat	25
3.1.3	Nové možnosti zpracování dat a vyhodnocení stavů a událostí z dat	26
3.2	Posouzení stavu systému vzhledem k realizaci nových požadavků	26
3.3	Volba přístupu k řešení	28

3.4	Sestavené požadavky na modul pro správu dat	28
3.4.1	Komunikace s adaptérem chytré sítě	28
3.4.2	Komunikace s nadřazeným systémem	28
3.4.3	Automaticky vykonávané operace s daty	29
3.4.4	Forma provedení.....	30
4	Návrh optimalizovaného řešení.....	31
4.1	Modul pro správu dat.....	31
4.1.1	Deklarativní rozdělení tabulek do oddílů a jeho využití v systému	31
4.2	Datový model pro správu dat.....	33
4.2.1	Schéma <i>data_tables</i>	34
4.2.2	Schéma <i>data_control</i>	37
4.2.3	Schéma <i>evaluated_data</i>	41
4.3	Obslužná aplikace	46
4.3.1	Přijímání dat z chytré sítě.....	46
4.3.2	Provádění definovaných operací s daty	46
4.3.3	Komunikace s nadřazeným systémem	46
5	Návrh nových možností prezentace naměřených dat.....	47
5.1	Stávající možnosti zobrazení dat a návrh na jejich zlepšení.....	47
5.2	Výběr vhodného pluginu pro prezentaci dat v grafech.....	48
5.3	Návrh nových možností zobrazení dat v grafech	48
6	Implementace navrženého řešení.....	53
6.1	Obslužná aplikace modulu pro správu dat	53
6.2	Instalační balíček modulu pro správu dat.....	54
6.3	Prezentace velkého objemu naměřených dat a dlouhých časových úseků	55
6.4	Vylepšení stávajících možností zobrazení dat.....	58
6.5	Optimalizace zobrazování spotřeb	59
7	Otestování naimplementovaného řešení	60
7.1	Testování modulu pro správu dat na simulovaných datech	60
7.2	Testování prezentace dat na simulovaných datech	60
7.3	Zátěžové testy a otestování celého řešení i na reálných datech	61
8	Závěr - Zhodnocení dosažených výsledků a diskuse dalšího vývoje systému .	62
8.1	Zhodnocení dosažených výsledků.....	63
8.1.1	Zpracování a zobrazování naměřených hodnot.....	63
8.1.2	Šetření úložného prostoru.....	63
8.1.3	Optimalizace zobrazování spotřeb.....	64
8.2	Diskuse dalšího možného vývoje systému.....	65
8.2.1	Zpracování dat	65
8.2.2	Zobrazení naměřených a zpracovaných dat.....	65
8.2.3	Spotřeby.....	65

8.2.4	Vyhodnocení událostí z dat.....	65
Literatura	67
Seznam příloh	70

Seznam obrázků

Obr. 1.1 Pyramidová struktura technologie [20]	14
Obr. 4.1 Příklad dynamicky generovaného obsahu schématu <i>data_tables</i>	35
Obr. 4.2 Schéma <i>data_control</i> datového modelu pro správu dat – část 1.....	43
Obr. 4.3 Schéma <i>data_control</i> datového modelu pro správu dat – část 2.....	44
Obr. 4.4 Schéma <i>evaluated_data</i> datového modelu pro správu dat.....	45
Obr. 5.1 Aktuální nepřehledné zobrazení naměřených hodnot měřicího přístroje	47
Obr. 5.2 Návrh zobrazení naměřených hodnot v tabulce	48
Obr. 5.3 Příklad zobrazení průběhu pomocí 5minutových agregovaných dat.....	51
Obr. 5.4 Zobrazení průběhu za vybraný úsek „A“ z Obr. 5.3. Průběh je zobrazen za hodinu pomocí 15sekundových agregovaných dat.	51
Obr. 5.5 Zobrazení průběhu za vybraný úsek „B“ z Obr. 5.3. Průběh je zobrazen za hodinu pomocí 15sekundových agregovaných dat.	52
Obr. 6.1 Zobrazení průběhu zvolené veličiny pomocí denních agregovaných dat (4. úroveň oddálení)	56
Obr. 6.2 Zobrazení průběhu zvolené veličiny pomocí hodinových agregovaných dat (3. úroveň oddálení).....	56
Obr. 6.3 Zobrazení průběhu zvolené veličiny pomocí 5minutových agregovaných dat (2. úroveň oddálení).....	57
Obr. 6.4 Zobrazení průběhu zvolené veličiny pomocí dat agregovaných za 15 sekund (1. úroveň oddálení).....	57
Obr. 6.5 Zobrazení původních (syrových) dat (bez oddálení).....	57
Obr. 6.6 Přiblížení průběhu zobrazeném na grafu z Obr. 6.5	58
Obr. 6.7 Ukázka nového zobrazení naměřených hodnot v tabulkách.	58

Seznam tabulek

Tab. 5.1 Návrh nastavení úrovní oddálení pro zobrazení sekundových záznamů ..	50
Tab. 6.1 Porovnání použití jednotlivých programovacích jazyků	53
Tab. 8.1 Porovnání potřebného místa pro uložení dat před a po optimalizaci.....	64
Tab. 8.2 Porovnání rychlosti načítání různých dat před a po optimalizaci.	64

Úvod

Systémy pro monitorování spotřeb energií jsou neodmyslitelnou součástí inteligentního měření (smart metering), které je základní složkou inteligentní sítě (smart grid). Tyto systémy, které jsou v nejvyšší úrovni měřicího řetězce, shromažďují naměřená data. Tato data po řádné analýze a zpracování prezentují ve vhodné formě koncovým uživatelům. V dnešní době jsou už běžně schopny z naměřených dat vyhodnotit různé stavy a události, které následně reportují uživatelům například formou SMS. Některé lepší systémy jsou schopny provést i vyúčtování.

Systémy pro monitorování spotřeb energií prošly současně s celým řetězcem inteligentního měření velkým vývojem a některé funkcionality, které byli dříve nadstandardní, jsou v dnešní době zcela běžné.

Cílem práce je optimalizovat aktuální informační systém společnosti ModemTec pro sledování spotřeb energií a splnit nové požadavky na zpracování a prezentaci dat. Aktuálně má systém velké problémy zpracovat a zobrazit i malé množství hodnot a to i přesto, že nejčastější záznamy měřených dat jsou po 15minutách. Zobrazení naměřených a zpracovaných dat probíhá v některých případech až několik minut. Nové řešení by mělo být schopno do jedné sekundy načíst stránku i s velmi komplexními grafy a to i pro nové typy měřidel, která mohou mít periodu vzorkování krátkou až jednu sekundu. Pro takové množství dat nebyl systém vůbec navržený.

V první kapitole této práce je sepsán stručný teoretický úvod do problematiky chytrých sítí a systémů pro monitorování spotřeb energií.

Druhá kapitola popisuje stávající systém firmy ModemTec pro monitorování spotřeb energií. Zaměřuje se na jednotlivé moduly systému. Kapitola dále obsahuje rešerši technologií, které systém používá.

Ve třetí kapitole jsou nejprve uvedeny nové požadavky na systém, které byly zadány firmou ModemTec. Poté je posuzován stav systému vzhledem k realizaci těchto nových požadavků a je volen nejvhodnější přístup k řešení s ohledem na vhodnost integrace řešení do aktuálního systému. Nakonec je podrobněji diskutována nejvhodnější varianta, kterou je vytvoření nového, samostatného modulu pro správu dat.

Ve čtvrté kapitole je velice podrobně popsán návrh modulu pro správu dat včetně detailního popisu navrženého datového modelu a obslužné aplikace. Modul je navrhován tak, aby za nadřazený systém řešil veškerou práci s naměřenými daty. Je navržen tak, aby byl schopen především velmi rychle zpracovávat data i z měřicích přístrojů s periodou vzorkování 1 sekunda a poskytoval tato data ve

vhodné formě tak, aby je systém pro monitorování spotřeb energií mohl z databáze co nejrychleji vyčítat a zobrazovat.

V páté kapitole je uveden návrh nových možností prezentace naměřených a zpracovaných dat. Je kladen důraz především na nejnáročnější případ, kterým je potřeba rychle a přehledně zobrazovat sekundové záznamy, které mají být v systému k dispozici alespoň 6 měsíců. Tomu odpovídá přibližně 15 miliónů naměřených záznamů jednoho měřicího přístroje, ve kterých musí mít uživatel možnost se snadno orientovat.

V šesté kapitole je popsáno řešení, které bylo naimplementováno podle návrhů vypracovaných v kapitolách 4 a 5. Jsou uvedeny použité programovací jazyky, jaká část řešení byla jakým jazykem realizována a přibližný počet řádků kódu v daných jazycích pro danou část řešení. Na obrázcích je přehledně ukázán příklad zobrazení velkého objemu dat a dlouhých časových úseků. Nakonec je sepsáno jaké části systému byly optimalizovány i bez použití modulu pro správu dat.

V sedmé kapitole je vysvětleno jakým způsobem bylo naimplementované řešení testováno. Jsou stručně popsány tři fáze testování. Nejdříve byl testován modul pro správu dat na simulovaných datech. Poté byla testována prezentace naměřených a zpracovaných dat na simulovaných datech. Nakonec byly provedeny zátěžové testy a otestování celého řešení i na reálných datech. U posledně zmíněného je stručně popsán i stav zatížení serveru, doba zpracování velkého objemu dat a doba, jakou se načítali grafy během největšího testovaného zatížení serveru.

Na závěr v osmé kapitole jsou zhodnoceny dosažené výsledky. Stručně je uveden stav nového řešení i při velké zátěži včetně doby, která je potřeba pro zobrazování grafů a to při zátěži back-endu (během zpracování dat) i bez zátěže. Dále je porovnáváno staré řešení s novým. Porovnáváno je využití úložného prostoru a rychlost zobrazení některých dat, tabulek a grafů. Nakonec je diskutován další možný vývoj tohoto systému.

1 CHYTRÉ SÍTĚ A MONITOROVÁNÍ SPOTŘEB ENERGIÍ

V případě pojmu chytrá síť (smart grid), někdy nazývána také jako inteligentní síť, je síť myšlena elektrická síť přenosových vedení, rozvoden, transformátorů a dalších součástí, kterými je dohromady dodávána elektrická energie z elektrárny až ke koncovým spotřebitelům (do obytných domů, paneláků, firem, obchodních center a tak podobně). Dříve plnila elektrická síť svoji funkci dostatečně, ale vzhledem ke stále se zvyšujícím nárokům, tato síť přestává vyhovovat moderním potřebám a začíná být špatně udržovatelná a neefektivní.

Chytrá síť je pak elektrická síť, která využívá digitálních technologií k zajištění interaktivní obousměrné komunikace mezi výrobními zdroji a koncovými uživateli, a která disponuje snímáním podél přenosového vedení, ovládacími prvky, počítači a automatizačními prostředky, díky čemuž může být zvýšena kapacita sítě, zlepšena spolehlivost, efektivita, udržitelnost a vyrovnány rozdíly mezi spotřebou a výrobou elektřiny. [17]

1.1 Výhody chytrých sítí

Výhody chytrých sítí jsou následující:

- Efektivnější přenos elektrické energie [17]
- Rychlejší obnova dodávky po výpadku [17]
- Méně operací a levnější správa výrobních zdrojů [17]
- Zvýšená bezpečnost [17]
- Redukce poptávky ve špičce [17][19]
- Lepší zákaznická podpora a přesnější vyúčtování [19]
- Detekce neoprávněných odběrů energie a detekce stočení měřidla [19]
- Šetrnější k životnímu prostředí a redukce uhlíkových emisí snížením dodávky energie. [19]
- Zapojení zdrojů alternativní energie a jejich efektivní kombinování s tradičními zdroji. Lepší integrace systémů pro výrobu energie, jejichž vlastníci jsou zákazníci. [18] [17]
- Snížení přetížení a předcházení výpadku automatickým přesměrováním toku energie po detekci přetížení distribuční sítě. [18]
- Monitorování dějů a technického stavu distribuční sítě [18]
- Vzdálené ovládání měřidel a vzdálený odečet dat [20]
- Okamžité sledování spotřeb a sběr dat pro následné analýzy, zpracování a vyhodnocení [20]

- Řešení poruch a výpadků distribuční sítě [18]
- Komunikace se zákazníkem v reálném čase [18]
- Optimalizace spotřeb zákazníka vzhledem k ceně a životnímu prostředí [18]

1.2 Monitorování spotřeb energií

Chytré sítě umožňují monitorování spotřeb energií s vysokou přesností, díky kterému lze odhalit zařízení, která spotřebovávají více energie, než je žádoucí, nebo odhalit situace, kdy je s energiemi plýtváno, a tak provést změny a ušetřit peníze. Sledováním výroby energie z vlastních obnovitelných zdrojů, sledováním kapacit baterií a aktuálních tarifů, které se odvíjí podle vytížení chytré sítě a jsou nejhorší ve špičce, je možné naplánovat běh spotřebičů tak, aby svoji práci vykonávali v čase, který je výhodný z hlediska financí. Monitorováním spotřeb energií tedy spotřebitelé získávají lepší představu o svých energetických potřebách a pomáhají jim najít původ neefektivního využití energií a způsoby, jak provést nápravu.

Systémy pro monitorování spotřeb energií by měli spotřebitelům nabízet především:

- Detailní zobrazení aktuálních i historických spotřeb a jejich porovnání. Přehledné zobrazení těchto informací v tabulkách i grafech.
- Zobrazení přehledu zpracovaných statistických údajů a vyhodnocených reportů z naměřených dat a spotřeb. Například reporty měsíčních spotřeb pro účely zúčtování nebo agregované hodnoty spotřeby za definované období (minimum, maximum, průměr atd.).
- Nastavení alarmů, kterými je spotřebitel upozorněn o událostech, které nastaly. Například upozornění, že byly překročeny stanovené meze spotřeby (spodní a vrchní hranice spotřeby, nepřetržitá spotřeba po stanovený čas atd.). Příchod upozornění formou SMS, e-mailů atd.
- Možnosti pro dálkové ovládání zařízení ovlivňující spotřebu, jako například ovládání ventilace, osvětlení, topení atd.
- Dálkové ovládání a vzdálené prohlížení veškerých informací v reálném čase (on-line) z jakéhokoliv místa pomocí chytrých telefonů, tabletů, počítačů atd.

1.3 Chytré měření a jeho komponenty

Chytré měření (Smart metering) je měření, které je prováděno chytrými měřicími přístroji, které je možno, narozdíl od obyčejných měřidel, použít pro obousměrnou komunikaci s centrálou (serverem). Mezi měřicím přístrojem a serverem mohou být ještě další body řetězce jako například data-koncentrátor.

Systém, který je v rámci této práce optimalizován, se nachází na vrcholu měřicího řetězce. Systém tedy běží na serveru.

Na Obr. 1.1 je v pyramidové struktuře zobrazena základní koncepce chytrého měření. Jsou zde uvedeny čtyři vrstvy a jaké jsou možnosti komunikace mezi těmito vrstvami. [20]



Obr. 1.1 Pyramidová struktura technologie [20]

1.3.1 Chytré měřicí přístroje a chytré elektroměry

Chytrý měřicí přístroj je přístroj, který naměřená data dokáže předat vyšší vrstvě chytré sítě (data-koncentrátoru, serveru atp.) ať už posláním dat vyšší vrstvě anebo poskytnutím možnosti pro jejich vyčtení.

Mezi další moderní funkcionality chytrých měřicích přístrojů patří:

- Vyčítání registrů v pravidelných časových intervalech. Pokud vyčtení není z jakýchkoliv důvodů provedeno, je možné ho provést zpětně znovu později. Měřicí přístroj si pamatuje dokumentací specifikované množství posledních naměřených hodnot registrů.
- Limitování odběru pro zákazníka, vzdálené odpojení zákazníka, vzdálené ovládání relé a nastavení pravidel a kalendáře tarifů v měřicím přístroji.
- Dobré zabezpečení proti nelegálnímu vniknutí, únikům informací nebo jejich změnám atp. [21]
- Aktualizování komunikačního software přístroje přes komunikační síť. [21]
- Kromě posílání naměřených dat také posílání informací důležitých ke správě chytré sítě a posílání událostí, které přístroj vyhodnotil (Například detekce neoprávněného vniknutí).
- Možnost provést některé vzdálené funkcionality i off-line, tedy lokálně pomocí terminálu, jako například vyčtení registrů, vyčtení i nastavení ovládání stavu relé a kalendáře tarifů.

Chytrý měřicí přístroj může s data-koncentrátorem chytré sítě komunikovat například pomocí PLC, GPRS, RS-485, M-BUS, Ethernet a další. [20]

1.3.2 Vodoměry, plynoměry, měřiče tepla, kalorimetry a další

Patří sem všechny měřicí přístroje kromě elektroměru. Tato zařízení komunikují s elektroměrem pomocí ZigBee, Wireless M-BUSu, metalického M-BUSu nebo v případě impulsních měřidel pomocí impulsů. Vždy dochází k přenosu dat směrem z těchto měřidel do elektroměru. [20]

1.3.3 Počítače, LCD panely, chytré telefony

Jedná se o zařízení, pomocí kterých mohou být zákazníkovi (spotřebitelovi) prezentovány například:

- jeho aktuální nebo i historické spotřeby energií, případně i zpracovaná data a různé vyhodnocené údaje
- informace o jeho aktuálním tarifu
- dostupné možnosti k přerušení a obnově odběru energie
- nastalé události

1.3.4 Data-koncentrátoři

Data-koncentrátor je prostředníkem mezi chytrými měřicími přístroji a servery. S chytrými měřicími přístroji může komunikovat přes PLC, GPRS nebo RS485. Komunikace mezi data-koncentrátorem a serverem může být zajištěna prostřednictvím LAN, WIFI nebo GPRS. Mezi základní funkce data-koncentrátoru patří: [20]

- Periodické a okamžité vyčítání dat z měřicích přístrojů. [20]
- shromažďování, zpracování a vyhodnocování dat. [20]
- Provádění příkazů z nadřazeného systému. [20]
- Upgrade firmware v DC a měřicích. [20]
- Přeposílání alarmů z měřicích přístrojů do nadřazeného systému na serveru. [20]
- Zabezpečená komunikace s měřicími přístroji i se servery. [20]
- Automatické nalezení a zavedení nových měřicích přístrojů. [20]
- Pravidelná synchronizace s nadřazeným systémem a synchronizace času v elektroměrech. [20]
- Provádění diagnostiky elektroměrů prostřednictvím data-koncentrátoru. [20]

1.3.5 Servery

Servery se nacházejí na vrcholu měřicího řetězce. Servery přijímají veškerá data a události, které jim data-koncentrátory odeslali. I u menších sítí se může jednat o obrovské množství dat, tak je důležité mít řešení vhodně optimalizované. Události, a především data jsou po přijetí analyzována a roztržděna kam náležitě patří. Nad některými jsou prováděny složitější operace pro zpracování a vyhodnocení dat. Nakonec musejí být správná data (nebo co z nich bylo vyhodnoceno) prezentována ve správný čas správnému uživateli nebo předhozena správné operaci pro další zpracování.

2 SYSTÉM FIRMY MODEMTEC PRO MONITOROVÁNÍ SPOTŘEB ENERGIÍ

Tato kapitola popisuje stávající systém firmy ModemTec pro monitorování spotřeb energií, který má být v rámci této práce optimalizován. Tento systém nese jméno SYMUR a jedná se o webový systém, který umožňuje analýzu a vizualizaci odběru elektrické energie a dalších médií, jako jsou voda, teplo a plyn.

Systém přijímá naměřené hodnoty a nastalé události z adaptéru chytré sítě a nabízí práci nad více projekty, mezi kterými je možné se přepínat. V každém projektu je možné provádět správu jednotlivých uživatelů, měřicích přístrojů, umístění, spotřebitelských jednotek, odběrných míst, v rámci systému provádět výměnu měřicího přístroje na odběrných místech a přiřazovat uživatele ke spotřebitelským jednotkám. Po přiřazení uživatele ke spotřebitelské jednotce je tomuto uživateli umožněn přístup k relevantním naměřeným hodnotám a informacím o spotřebě, které mohou být přehledně zobrazeny v tabulkách i grafech a v některých případech i stažené do souboru.

2.1 Pro koho je systém určen

Cílenou skupinou uživatelů jsou: [16]

- dodavatelé energií
- správci budov, obchodní centra a rekreační střediska
- bytová družstva a společenství vlastníků jednotek
- developři
- školy a nemocnice
- pronajímatelé nemovitostí, města a obce
- správci areálů průmyslových, administrativních, sportovních a dalších
- pořadatelé masových akcí a veletrhů
- projektanti navrhující projekty budov a sídlišť

2.2 Rozdělení systému do modulů a jejich popis

Systém je rozdělený do následujících pěti modulů:

- Modul pro přihlášení uživatelů
- Modul pro správce
- Modul pro spotřebitele
- Modul pro příjem SOAP zpráv z adaptéru chytré sítě
- Modul pro vyhodnocení denních stavů spotřeb z naměřených hodnot

2.2.1 Modul pro přihlášení uživatelů

Jedná se o malý, ale důležitý modul, kterým je zajištěno bezpečné přihlášení a autorizace uživatelů, blokace uživatelů po neúspěšných pokusech o přihlášení, ošetření případu, kdy uživatel zapomene heslo, a další podobné akce.

2.2.2 Modul pro správce

Modulem pro správce je prováděna celá správa jednotlivých projektů a navíc jsou jím nabízeny:

- možnosti pro sledování měsíčních a ročních spotřeb všech odběrných míst vybraného projektu za účelem zúčtování
- možnosti pro přizpůsobení GUI a některých funkcí pro jednotlivé projekty
- sub-moduly pro technickou a uživatelskou podporu, kterými je nabízeno prohlížení SOAP logů přijatých z adaptéru chytré sítě, správa uživatelů typu uživatelská podpora, možnosti pro úpravu řetězců do různých jazyků a další.

Správa projektu je rozdělena na správu umístění, přístrojů a uživatelů. V další části je popsán přehled možností jednotlivých správ.

Správa umístění:

- Vytváření, úprava a mazání adres, spotřebitelských jednotek a odběrných míst. Nastavení sledovaných veličin na odběrných místech.
- Přiřazování uživatelů ke spotřebitelským jednotkám za účelem danému uživateli povolit přístup k informacím o spotřebě dané spotřebitelské jednotky.

Správa přístrojů:

- Přehled nalezených, aktivních i vyřazených měřicích přístrojů.
- Kontrola správného příchodu naměřených dat po zavedení nového měřicího přístroje do chytré sítě.
- Přiřazování přístrojů k odběrným místům. Výměna přístrojů na odběrných místech v rámci systému.

Správa uživatelů:

- Vytváření, úprava a mazání uživatelských rolí. Nastavování oprávnění jednotlivým uživatelským účtům.
- Vytváření, úprava, mazání, zablokování a odblokování uživatelských účtů.

2.2.3 Modul pro spotřebitele

Tento modul přehledně prezentuje veškeré informace o spotřebě a naměřených hodnotách, které jsou relevantní pro aktuálně přihlášeného uživatele. Jsou

zobrazeny přehledy spotřeb těch spotřebitelských jednotek, ke kterým je uživatel přiřazen správcem projektu. Modul pro každé odběrné místo spotřebitelské jednotky prezentuje aktuální stav připojeného měřáku, naměřené hodnoty, spotřebu za den, týden, měsíc, rok a jejich porovnání, a to v různých grafech i tabulkách.

Úvodní stránka modulu pro spotřebitele zobrazuje seznam spotřebitelských jednotek, které jsou přihlášenému uživateli dostupné. Pro každou takovou spotřebitelskou jednotku zobrazuje seznam odběrných míst, a to i s nejaktuálnějšími informacemi o spotřebě. Je možné se odtud přesunout na detail spotřeb zvoleného odběrného místa.

Stránka s detailem spotřeb odběrného místa umožňuje detailní prohlížení informací o spotřebě, a to od počátku měření až do současnosti. Je možné prohlížet přehledy denních, týdenních, měsíčních a ročních spotřeb, a to v tabulkách i grafech.

2.2.4 Modul pro příjem SOAP zpráv z adaptéru chytré sítě

Tímto modulem jsou systémem přijímány naměřené hodnoty, nastalé události a informace o nových měřicích přístrojích připojených do chytré sítě. Přijaté informace jsou zapsány do databáze. Pro účely pozdějšího ladění je uložena i celá SOAP zpráva.

2.2.5 Modul pro vyhodnocení denních spotřeb z naměřených dat

Modulem jsou jednou denně pro všechna odběrná místa provedeny výpočty, jejichž výsledkem jsou denní spotřeby pro jednotlivá odběrná místa. Současně jsou za účelem šetření úložného prostoru odstraňovány staré SOAP zprávy, o kterých se ví, že už nebudou pro ladění nutné.

2.3 Použité technologie

Prezentační vrstva (front end) systému je napsána v jazycích HTML5, CSS, PHP a JavaScript. Je využito PHP frameworku Nette a JavaScript knihovny jQuery.

Datovou vrstvu (back end) systému obstarává objektově relační databázový systém PostgreSQL a unixový démon pro periodické spouštění úloh Cron.

Systém běží na Ubuntu Server. Pro hostování webových stránek systému je použitý webový server Apache. Systém přijímá data a události z adaptéru chytré sítě formou XML zpráv podle SOAP protokolu. Při instalaci systému se pro kontrolu závislostí a stažení všech balíčků, na kterých systém závisí, stará PHP Composer.

2.3.1 HTML5

HTML (Hypertext Markup Language) je základní značkovací jazyk pro tvorbu webových stránek v systému World Wide Web, které mohou být propojeny hypertextovými odkazy. [1][2]

HTML5 je verze značkovacího jazyka HTML. HTML5 představuje podstatný krok kupředu v návrhu, rozložení a používání webu. Třemi hlavními elementy představenými v HTML5 jsou `<audio>`, `<video>` a `<canvas>`, ale jsou i další. HTML5 oproti svým předchozím verzím nabízí především jednoduchý způsob, jak bez použití plug-inů vložit do stránek zvuk a video nebo možnosti pro manipulaci s grafikou. Dále nabízí skripty pro provádění úkolů na pozadí (web workers), vylepšenou práci s formuláři, přístup ke svazkům lokálního úložného prostoru, možnosti udělat z webových stránek webové aplikace pro mobilní prohlížeče, podpora pro běh webových aplikací v offline režimu (bez přístupu k internetu) s uložením dat na lokální uložení a pozdější synchronizací dat, jakmile je internetové připojení k dispozici a další. [2][1]

2.3.2 CSS

CSS (Cascading Style Sheets, česky kaskádové styly) je jazyk, který popisuje, jakým způsobem se mají zobrazovat elementy v dokumentech napsaných ve značkovacích jazycích, jako je například HTML. Název „kaskádové“ mají proto, že se na sebe mohou vrstvit jednotlivé definice stylu, ale platí jenom ta poslední definice. [1] Použitím CSS stylů je tedy možné podle představ upravit vzhled webových stránek napsaných v HTML5 a to změnou stylu jakéhokoliv elementu. Kromě různých úprav statického charakteru je pomocí CSS možné do webových stránek přidat také různé animované přechody a transformace. [2]

Existuje více způsobů pro přidání stylů do webové stránky:

- Přímý zápis stylu pomocí atributu „style“ k danému elementu. Daný styl bude platit pouze na dotýčný element. [2][1]
- Vložení stylů do hlavičky webové stránky. Styly budou platit pouze v rámci celé webové stránky. [2][1]
- Vložení stylů do zvlášť do souborů oddělených od samotných webových stránek a poté do jednotlivých webových stránek importovat soubory těch stylů, které jsou potřeba. Tento způsob je používán v případech, kdy jsou styly potřeba v rámci všech webových stránek nebo celé webové aplikace. Dále změnou výběru souboru se styly je možné použít různé styly pro různé požadavky na rozložení stránky (rozložení stránky pro web nebo pro tisk) aniž by se musel změnit HTML kód. [2][1]

2.3.3 PHP

PHP (Hypertext Preprocessor, česky hypertextový preprocesor) je open-source skriptovací jazyk, který je vhodný především pro programování dynamických internetových stránek a webových aplikací. Může být vepsán přímo do HTML kódu. Na rozdíl od jazyku JavaScript, který je spouštěn na straně klienta, je PHP spouštěno na straně serveru, kde generuje HTML kód, který je poté poslán klientovi. Klient se tedy k PHP kódu vůbec nedostane a přijme pouze jeho výsledek, který je serverem vyhodnocen po spuštění daného PHP skriptu.

PHP skripty jsou používány v následujících oblastech:

- Skriptování na straně serveru – je hlavním cílem PHP.
- Použití PHP skriptů v příkazovém řádku – PHP skript může být spuštěn i bez serveru a prohlížeče. Vhodné například pro pravidelné spuštění skriptů přes Cron.
- Psaní desktopových aplikací – je možné díky PHP rozšíření PHP-GTK.

2.3.4 Nette

Nette je rodina vyspělých a samostatně použitelných komponent pro PHP, které dohromady tvoří framework. Nette je český framework, který je open-source a běží na něm mnoho významných českých webů. [15]

2.3.5 JavaScript

JavaScript je skriptovací jazyk, který umožňuje vytvářet dynamické a interaktivní elementy webových stránek. Syntaxe jazyka je podobná jazyku C. JavaScript běží na straně klienta, tedy zdrojový kód jazyka je zpracováván až webovým prohlížečem klienta, nikoliv na webovém serveru, jak je tomu v případě serverového jazyka PHP. [4][3]

Pomocí jazyka JavaScript je možné měnit obsah webové stránky přímo u uživatele, čehož je využíváno například:

- Pro validaci formuláře, kdy je pomocí jazyka JavaScript zkontrolováno, zda jsou všechny pole formuláře správně vyplněna, a to ještě před tím, než je formulář odeslán. JavaScript může ve webovém prohlížeči zobrazit uživateli hlášení o daném problému a nepovolit tak odeslání formuláře, tedy server není tímto zbytečně zatěžován. [3][2] Protože však JavaScript běží na straně klienta, může být uživatelem vypnutý, proto by měl být formulář zkontrolován znovu i na serverové straně. [4]
- Tvorba dynamických menu, roletek a dalších pohyblivých kontejnerů [4][2]
- Různé aplikace ve webové stránce jako jsou editory textu, hry, a další. [4]

JavaScript kód může být vepsán kamkoliv přímo do HTML kódu webové stránky a nebo samostatně do „.JS“ souboru. V obou případech může být kód v prohlížeči zobrazen. [3]

V poslední době je stále populárnější nástroj Node.js, který umožňuje spouštět aplikace napsané v jazyce JavaScript jak na straně klienta, tak i na serverové straně. [5]

2.3.6 jQuery

jQuery je nejpopulárnější a velmi spolehlivá knihovna pro jazyk JavaScript s širokou podporou webových prohlížečů. [2]

2.3.7 PostgreSQL

PostgreSQL je objektově-relační databázový systém (ORDBMS – Object Relational Database Management System). PostgreSQL je open-source, podporuje velkou část SQL standardu a nabízí mnoho moderních funkcí jako například: [6]

- komplexní dotazy (komplexními dotazy jsou označovány dotazy, kde je kromě jednoduchého použití SELECT a WHERE příkazů použito i komplexní provázání s jinými tabulkami pomocí různých variant příkazů JOIN, případně i s použitím složitých pod dotazů nebo velkého množství klauzulí AND a OR)
- cizí klíče
- trigger (spouštěče) a spouštěcí události
- aktualizovatelné pohledy do databáze
- transakce
- rozdělení tabulek do oddílů (table partitioning)

PostgreSQL může být rozšířeno uživatelem o nové: [6]

- datové typy
- funkce
- agregační funkce
- operátory
- indexové metody
- procedurální jazyky

2.3.8 Cron

Cron je unixový systémový démon, který v definovaných okamžicích spouští na pozadí příkazy, procesy, skripty, programy a podobně. Používá se na opakované spouštění periodicky se opakujících procesů. [7]

2.3.9 Ubuntu

Ubuntu je open-source operační systém založený na distribuci Debian GNU/Linux. Ubuntu je navržen pro osobní počítače (edice Ubuntu Desktop) a servery (edice Ubuntu Server). [8]

2.3.10 Apache HTTP Server

Apache HTTP Server, který bývá nazýván jednodušeji jako Apache, je open-source webový server, který v současné době dodává prohlížečům na celém světě většinu internetových stránek. [9][10] Apache prohlížečům zvládne dodávat HTML soubory, obrázky, flash soubory, různé multimediální soubory a spousty dalších. [2]

Apache také podporuje velkou škálu vlastních modulů, jako jsou například:

- mod_security - Modul pro zajištění bezpečnosti serveru. Modul může server ochránit před různými útoky. [11]
- mod_rewrite - modul, který je používán k přepsání adres URL tak, že může být docíleno přesměrování. [11]
- mod_deflate – Modul je používán pro kompresi výstupu z webového serveru před tím, než je poslán klientovi. Klientovi je tak umožněno stažení souboru s menší velikostí. [11]
- mod_cache – Modul pro zlepšení výkonu serveru. Modul uloží často používaný obsah na snadno přístupné místo, aby měl klient k těmto datům přístup rychlejší. [11]
- mod_proxy - Proxy modul, který může být použit pro rychlé poskytnutí často žádaných stránek z paměti cache, čímž je docíleno menší zátěže serveru. [2]
- mod_ssl – Modul umožňuje šifrování přes SSL (Secured Sockets Layer) a TLS (Transport Layer Security) za pomoci open-source SSL/TLS nástroje OpenSSL. [11]

2.3.11 Webové služby a SOAP protokol

Technologie webových služeb umožňuje volat funkce na vzdáleném systému, zajišťuje jednoduchou komunikaci mezi aplikacemi na nejrůznějších platformách a je platformě zcela nezávislá. Platformní nezávislost je dosažena použitím XML jako formátu pro vzájemnou výměnu dat. Prostřednictvím XML zpráv spolu aplikace komunikují dohodnutým protokolem. [12]

Webové služby jsou postaveny na následujících technologiích:

- SOAP (Simple Object Access Protocol) – Protokol pro komunikaci pomocí XML zpráv. Protokol SOAP je základem webových služeb. [12]
- WSDL (Web Services Description Language) – Standardní formát pro popis rozhraní webové služby. [12]

- UDDI (Universal Description, Discovery and Integration) – Standardní mechanismus registrace a vyhledávání webových služeb. [12]

2.3.12 Composer

Composer je nástroj pro správu balíků a závislostí pro PHP projekty. Závislostmi mohou být například různé frameworky, knihovny, a další. Není tedy nutné ručně stahovat a rozbalovat balíčky a spravovat i jejich verze. Poté, co je Composer seznámen s tím, na čem všem je nový projekt závislý, je pak možné dát nástroji Composer pokyn k instalaci všech závislostí nebo i k jejich aktualizaci na nejnovější verzi. [13]

3 ANALÝZA NOVÝCH POŽADAVKŮ NA SYSTÉM PRO MONITOROVÁNÍ SPOTŘEB ENERGIÍ

Nové požadavky na systém pro monitorování spotřeb energií byly zadány přímo firmou ModemTec. Jelikož se jedná o požadavky na systém, který je již plně v provozu, je nutné zhodnotit stav systému vzhledem k těmto požadavkům a poté zvolit takový přístup k řešení, aby ho bylo možné do aktuálního systému vhodně integrovat. V této kapitole je:

1. uveden seznam s novými požadavky
2. posouzen stav systému vzhledem k novým požadavkům
3. prezentován zvolený přístup k řešení, který byl vybrán s ohledem na vhodnost integrace nových požadavků do aktuálního systému
4. po zvolení modulárního přístupu k novému řešení jsou definovány požadavky na nový modul pro správu dat

3.1 Nové požadavky zadané firmou ModemTec

Nové požadavky lze rozdělit do třech kategorií:

- Značné zrychlení celého systému (optimalizace)
- Nové možnosti zpracování dat a vyhodnocení různých stavů a událostí z dat
- Nové možnosti prezentace dat

3.1.1 Značné zrychlení celého systému (optimalizace)

Systém musí běžet rychle jak pro malé projekty s několika nenáročnými měřicími přístroji, tak i pro projekty, které čítají desítky měřidel s periodou vzorkování 1 sekunda. Z těchto měřidel je nutné si pamatovat záznamy v každé sekundě po dobu alespoň 6 měsíců. To je přibližně 15 milionů záznamů s naměřenými a zpracovanými daty z jednoho měřicího přístroje.

3.1.2 Nové možnosti prezentace dat

- Vhodné prezentování velkého objemu naměřených a zpracovaných dat za dlouhý časový úsek. Nejnáročnějším případem jsou sekundové záznamy za období až 6 měsíců. Těchto 15 milionů záznamů musí být zobrazováno takovým způsobem, aby se v nich mohl uživatel snadno a rychle orientovat.
- Různé možnosti prezentování událostí, které byly vyhodnoceny z dat a to tabulkou, upozorněním, e-mailem a SMS zprávou.

3.1.3 Nové možnosti zpracování dat a vyhodnocení stavů a událostí z dat

- výpočet energie z naměřených výkonů
- výpočet statistických údajů, a to konkrétně průměrné, minimální a maximální hodnoty za zvolené období
- archivace naměřených a zpracovaných dat
- automatické odstraňování (zapomínání) dat po zvoleném čase
- detekce hodnoty mimo zvolené meze
- detekce úniku nebo nepřetržitého průtoku vody
- detekce stočení měřidla
- detekce ztráty komunikace s přístrojem
- označování typů událostí zaznamenávaných v logu přístroje pro reportování nastalé události upozorněním, e-mailem nebo SMS zprávou.

3.2 Posouzení stavu systému vzhledem k realizaci nových požadavků

V rámci požadavku k celkovému zrychlení systému bylo po prostudování a otestování systému na reálných datech zjištěno, že rychlost systému strádá v případech, kdy se z databáze vyčítají naměřená data anebo denní stavy spotřeb vyhodnocené z těchto dat.

Následně bude uveden seznam, ve kterém jsou uvedeny případy, kdy je systém příliš pomalý, včetně času, který je potřeba pro dokončení dané operace. Uvedené časy platí pro nezatížený server s parametry odpovídajícím 2 GB RAM a 2 GHz procesoru s 2 jádry. Dále byl v systému zvolen projekt přibližně se 100 odběrnými místy, se 70 000 denními stavy spotřeb a s 1 000 000 záznamů naměřených dat z měřicích přístrojů. Seznam je následující:

- Až 1 minuta a 15 sekund pro načtení základních informací o spotřebě jednoho odběrného místa na úvodní stránce modulu pro spotřebitele. Tyto informace pro dané odběrné místo zahrnují spotřebu za aktuální den, předchozí den, aktuální měsíc a předchozí měsíc. Dále porovnání těchto spotřeb a poslední naměřenou hodnotu měřicím přístrojem.
- Až 1 minuta pro načtení tabulky nebo grafu denních a týdenních spotřeb v detailu odběrného místa v modulu pro spotřebitele. Tabulka i graf denních, respektive týdenních spotřeb zobrazuje k danému odběrnému místu spotřebu zvolené veličiny za každý den, respektive týden od počátku měření až do současnosti.

- Až 10 sekund pro načtení tabulky nebo grafu měsíčních a ročních spotřeb pro zvolené odběrné místo. Tabulka i graf měsíčních, respektive ročních spotřeb zobrazuje k danému odběrnému místu spotřebu zvolené veličiny za každý měsíc, respektive rok od počátku měření až do současnosti.
- Až 2 minuty pro načtení tabulky s přehledem spotřeb v modulu pro správce. Tabulka zobrazuje spotřebu za zvolený měsíc nebo rok pro všechna odběrná místa, která náleží projektu, ke kterému je správce přihlášen.
- Až 1 hodina pro operaci pro vyhodnocení denních stavů spotřeb z naměřených dat pro všechna odběrná místa všech projektů, která se provádí vždy jednou denně v noci.
- Až 10 sekund pro načtení 50-500 záznamů dat do jedné z dynamicky načítaných stran tabulky v detailu měřicího přístroje v modulu pro spotřebitele. Jeden záznam tabulky zobrazuje naměřenou hodnotu, typ hodnoty, jednotku, čas měření a registrovou mapu, ve které hodnota dorazila.
- Až 2 minuty pro načtení tabulky SOAP logů. Záznamy této tabulky informují o úspěšnosti zpracování přijatých SOAP zpráv z adaptéru chytré sítě, kterými je systém informován o naměřených datech.

Problém pomalého běhu systému pro výše uvedené případy je způsoben nejen neoptimalizovanými dotazy do databáze, ale také samotným datovým modelem, který neumožňuje data zpracovávat, vyhodnocovat ani prezentovat optimálním způsobem. Pro co největší zrychlení systému bude zapotřebí upravit část datového modelu, poté provést odpovídající změny pro načítání dat do prezentační vrstvy systému, a nakonec převést naměřená data ze staré verze databáze do nové podle provedených změn.

V rámci požadavku k novým možnostem prezentace dat bylo po prostudování systému zjištěno, že systém obsahuje pouze statické grafy, které zobrazují vývoj naměřené hodnoty v čase nebo zobrazení denních, týdenních, měsíčních a ročních spotřeb.

Pro vhodné prezentování velkého objemu naměřených dat a dlouhých časových úseků je nutné najít řešení především na problém, kdy pro uživatelem zvolené období je pro zobrazení k dispozici mnohem více dat, než kolik je možné vykreslit na zobrazovaný časový úsek. V takových případech bude nutné použít dynamické grafy a agregovaná data.

V rámci nových požadavků na zpracování a vyhodnocení dat a pro prezentaci událostí, které byly z dat vyhodnoceny, bude nutné celé řešení navrhnout a implementovat.

3.3 Volba přístupu k řešení

Všechny nové požadavky na systém pro monitorování spotřeb energií se týkají komplexního problému analýzy, zpracování, vyhodnocení a prezentace dat. Aktuální možnosti systému v této oblasti musejí být v rámci optimalizací nahrazeny. Vzhledem k těmto skutečnostem, byl zvolen modulární přístup k řešení, kdy bude zcela od začátku navržen a implementován samostatný modul, který bude řešit veškerou analýzu, zpracování a vyhodnocení dat za nadřazený systém pro monitorování spotřeb energií.

Vzhledem k tomu, že aktuální způsob řešení, kdy příjem SOAP zpráv z adaptéru obstarávají PHP skripty a vyhodnocení stavů z naměřených dat je vykonáváno pomocí PHP skriptů volanými přes Cron, není optimální, bude muset být pro nový modul zvolen jiný programovací jazyk, který bude nabízet více kontroly a optimálnější práci s databází. Z tohoto důvodu bude muset být navrženo a implementováno nějaké komunikační rozhraní mezi modulem a nadřazeným systémem. Takto bude navíc modul nezávislý na aktuálním nadřazeném systému a bude ho moci využít jakýkoliv jiný nadřazený systém, který má implementované dané komunikační rozhraní.

Vzhledem ke své komplexnosti je nový modul pojmenován obecněji jako „Modul pro správu dat“ a ještě před jeho návrhem je nutné přesně specifikovat všechny požadavky na tento modul.

3.4 Sestavené požadavky na modul pro správu dat

Požadavky na modul pro správu dat lze rozdělit do čtyř kategorií:

- Komunikace s adaptérem chytré sítě
- Komunikace s nadřazeným systémem
- Automaticky vykonávané operace s daty
- Forma provedení

3.4.1 Komunikace s adaptérem chytré sítě

Přijímání a potvrzování příchozích SOAP zpráv z adaptéru chytré sítě. Uložení všech důležitých informací z těchto zpráv, které se týkají naměřených dat, událostí a informací o nově připojených zařízeních v chytré síti.

3.4.2 Komunikace s nadřazeným systémem

Přijímání a potvrzování příchozích SOAP zpráv z nadřazeného systému. Po přijetí těchto zpráv musí být modul schopný realizovat požadavky nadřazeného systému:

- Nadřazený systém může dát pokyn k automatickému provádění vybraných operací nad vybranými daty.

- Nadřazený systém může parametrizovat operace, které se nad daty provádějí.
- Nadřazený systém může dát pokyn k deaktivaci nebo odstranění operací prováděnými nad daty.
- Nadřazený systém může dát pokyn ke smazání dat, stavů a událostí vybraných podle zvoleného zařízení, ke kterému jsou data, stavy a události vztahovány, nebo podle zvolené operace, ze které byly vygenerovány.

Posílání SOAP zpráv nadřazenému systému s očekáváním potvrzení. Posílání zpráv znovu později, pokud modul nedostane od nadřazeného systému potvrzení o jejich přijetí. Požadavky na modul týkající se této komunikace:

- Informování nadřazeného systému o stavu splnění požadavků, které modulu pro správu dat zadal nadřazený systém (úspěšně hotovo / stala se chyba).
- Reportování událostí nadřazenému systému formou SOAP zpráv ihned potom, co tyto události byly z dat vyhodnoceny.

Poskytování databázových tabulek s daty s oprávněním „jen pro čtení“ nadřazenému systému, a to ve vhodně formátované podobě tak, aby mohla být data nadřazeným systémem čtena a prezentována optimálně, a to i pro případ měřidel s periodou vzorkování až 1 sekunda.

Poskytování i všech ostatních databázových tabulek s oprávněním „jen pro čtení“ nadřazenému systému. Především aby si mohl nadřazený systém zjistit:

- historii všech vygenerovaných událostí a stavů
- nad jakými daty je aktuálně aktivováno provádění jakých operací
- jaké možnosti má pro parametrizaci operací pro analýzu, zpracování a vyhodnocení dat
- jaké operace pro analýzu, zpracování a vyhodnocení dat mohou být vykonávány nad vybranými daty

3.4.3 Automaticky vykonávané operace s daty

Vykonávání operací s daty v rámci jejich analýzy:

- rozdělování dat přijatých z adaptéru chytré sítě podle registrové mapy, registru a zařízení, ze kterého pocházejí
- označování dat čekajících na zpracování nebo vyhodnocení

Vykonávání operací s daty v rámci jejich zpracování:

- počítání veličiny z naměřených veličin podle zvoleného vzorce (například výpočet energie z naměřeného výkonu)

- počítání statistických údajů, a to konkrétně průměrné, minimální a maximální hodnoty za zvolené období

Vykonávání operací s daty v rámci jejich vyhodnocení, kdy jsou výsledkem události, které má modul ihned reportovat nadřazenému systému:

- detekování naměřené nebo zpracované hodnoty mimo zvolené meze
- detekování neustále rostoucí hodnoty měřené veličiny po zvolený časový úsek se zvolenou rezervou nárůstu hodnoty (například pro detekci úniku nebo nepřetržitého průtoku vody)
- detekování poklesu naměřené hodnoty kumulativní veličiny (například pro detekování stočení měřidla)
- detekování chybějících dat po zvolený časový úsek (například pro detekci ztráty komunikace s přístrojem)
- přeposílání vybraných událostí zaznamenávaných v logu přístroje

Vykonávání operací s daty v rámci jejich vyhodnocení, kdy jsou výsledkem stavy:

- vyhodnocování denních stavů spotřeb na odběrných místech
- vyhodnocování týdenních stavů spotřeb na odběrných místech
- vyhodnocování měsíčních stavů spotřeb na odběrných místech
- vyhodnocování ročních stavů spotřeb na odběrných místech

Vzhledem k neustálému vývoji technologií a přibývání nových požadavků, které zasahují i do kategorie zpracování a vyhodnocení dat, musí být modul sestaven tak, aby do něho bylo kdykoliv možné jednoduchým způsobem doimplementovat nové operace pro analýzu, zpracování a vyhodnocení dat.

3.4.4 Forma provedení

Modulární řešení – Oddělení modulu od aktuálního nadřazeného systému pro sledování spotřeb energií tak, aby modul bylo možné použít s jakýmkoliv jiným nadřazeným systémem, který má implementované dané komunikační rozhraní (komunikace pomocí SOAP zpráv a s omezením na oprávnění „pouze pro čtení“ i pomocí SQL dotazů do databáze).

Pro databázi použití objektově relačního databázového systému PostgreSQL kvůli zachování jednotnosti s nadřazeným systémem, který PostgreSQL používá.

Optimální běh celého modulu takový, aby zvládl rychle vykonávat velké množství operací prováděnými nad obrovským množstvím dat. Za tímto účelem musí být navrhován jak datový model, tak i zvoleny optimalizační techniky a vhodný programovací jazyk pro implementaci modulu.

4 NÁVRH OPTIMALIZOVANÉHO ŘEŠENÍ

Při analýze nových požadavků na systém pro monitorování spotřeby energií, která byla provedena v předchozí kapitole, bylo zjištěno, že všechny nové požadavky se týkají komplexního problému analýzy, zpracování a prezentace dat a vyhodnocení různých událostí a stavů z těchto dat. Bylo rozhodnuto, že nejlepším přístupem k řešení bude modulární přístup, kdy bude navržen a implementován samostatný modul pro správu dat, který bude za nadřazený systém řešit veškerou práci s naměřenými daty přijatými z adaptéru chytré sítě, a který bude poskytovat data ve vhodně formátované podobě tak, aby je mohl nadřazený systém optimálně načíst a prezentovat.

V této kapitole je popsán návrh modulu pro správu dat podle požadavků, které byly definovány na konci předchozí kapitoly.

4.1 Modul pro správu dat

Navržený modul pro správu dat je složen z databáze a obslužné aplikace.

Pro obslužnou aplikaci byl zvolen jazyk C++, protože nabízí mnohem více kontroly a optimálnější práci s databází než aktuální řešení, u kterého příjem SOAP zpráv z adaptéru obstarávají PHP skripty a vyhodnocení stavů z naměřených dat je vykonáváno pomocí PHP skriptů volanými přes Cron. Dalším důvodem pro zvolení C++ bylo, že autor práce v něm má dobré znalosti i zkušenosti, stejně jako pracovníci firmy ModemTec, kteří na jeho práci mohou v budoucnu navázat.

Pro databázi byl zvolen objektivě relační databázový systém PostgreSQL především kvůli zachování jednotnosti s nadřazeným systémem, který PostgreSQL používá. Kvůli potřebě využití deklarativního rozdělení tabulek do oddílů (declarative partitioning) byla zvolená nejnovější verze PostgreSQL 11. Verze PostgreSQL 9.5, kterou systém používá, nepodporuje deklarativní rozdělení tabulek do oddílů a bude tak muset být v systému aktualizována také na verzi 11.

4.1.1 Deklarativní rozdělení tabulek do oddílů a jeho využití v systému

Deklarativní rozdělení tabulek do oddílů je způsob, jakým PostgreSQL umožňuje rozdělit jednu tabulku na více částí, které se nazývají oddíly. Při vložení nového záznamu do tabulky se pak tento nový záznam vloží do patřičného oddílu a to podle pravidla, podle kterého byly tyto oddíly vytvořeny. Pokud je pak při vyhledávání záznamů v této tabulce zadán i filtr, který nějak souvisí s tímto pravidlem, pak je plánovač databáze dopředu informován o tom, ve kterém oddílu má záznamy hledat a zbytečně se nezatěžuje ostatními částmi tabulky. Toto rozdělení je výhodné pokud

má tabulka obrovské množství záznamů. Množství záznamů, při kterých začíná být rozdělení tabulky do oddílů výhodné závisí na aplikaci i na výkonu serveru. [6]

Nastavení deklarativního rozdělení tabulek do oddílů je tedy potřeba řádně promyslet a někdy i experimentálně prozkoušet. V rámci návrhu modulu pro správu dat byly v PostgreSQL databázi, která běžela na virtualizovaném serveru, vytvořeny tabulky se simulovanými daty, nad kterými byli poté otestovány vybrané operace, které bude nový systém běžně provádět. Byla vyzkoušena varianta s obyčejnou tabulkou a pak varianty s rozdělením tabulky do oddílů (různé nastavení celkového počtu oddílů dané tabulky; různé nastavení počtu záznamů na jeden oddíl; různé nastavení rozdělení oddílů na pod-oddíly). Byl také otestován vliv přidělených prostředků virtuálnímu serveru (velikost paměti RAM, počet jader procesoru a jeho omezení).

Z výsledků testování byl vyvozen závěr, že modul pro správu dat bude tuto vlastnost PostgreSQL (declarative partitioning) využívat k rozdělení všech tabulek, které budou sloužit pro ukládání naměřených nebo zpracovaných dat s periodou menších než 1 minuta.

Nejkratší perioda vzorkování, pro kterou má být systém optimalizován, je jedna sekunda. Tyto sekundové záznamy si má systém pamatovat alespoň 6 měsíců. To je přibližně 15 miliónů záznamů v jedné tabulce pro jeden měřicí přístroj. Takové množství záznamů by v jedné nerozdělené tabulce zapříčinilo, že by i vyčtení několika hodnot pro zobrazení jednoduchého grafu trvalo desítky sekund až jednotky minut. „Skutečná doba ale závisí na spoustě faktorů, například i na velikosti fyzické paměti databázového serveru.“ [6] Dále je potřeba tyto data také pravidelně zpracovávat (výpočet agregovaných dat, kontrola limitů, výpočty dalších veličin z dat jako například energie z naměřených třífázových výkonů, atd...). Rozdělením tabulky do oddílů budou všechny tyto operace značně urychleny.

Rozdělení tabulky do oddílů bude provedeno podle času naměřených dat tak, aby jeden oddíl obsahoval nejvýše 86 400 záznamů. To v případě záznamů v každé sekundě odpovídá datům za jeden celý den a celkem 180 oddílům rozdělené tabulky pro období 6 měsíců. Při vyčtení dat z databáze je pak možné zadat název (hlavní, rozdělené) tabulky a časový rozsah naměřených dat, které chceme vyčíst, upravit nebo zpracovat. „Na základě zadaného časového rozsahu bude plánovač PostgreSQL databáze dopředu informován o tom, ve kterém oddílu data hledat.“ [6]

Celé řešení modulu pro správu dat je navrženo tak, aby skrze jeden SQL dotaz nebylo nikdy nutné pracovat s větším časovým rozsahem, než jaký odpovídá maximálně dvěma oddílům rozdělené tabulky. Nikdy se tedy nebude prohledávat celá (hlavní) rozdělená tabulka. Například pro zobrazení průběhu veličiny za 10 dní si nadřazený systém nebude z tabulky se sekundovými záznamy vyčítat vůbec nic, ale vyčte si data agregovaná za každou 1 hodinu, která jsou uložena v jiné tabulce.

Dále například pro výjimečné případy, kdy by náhodou do systému dorazila některá naměřená data například o měsíc později a je tak potřeba, aby již zpracovaná data byla přepočítána za celý měsíc, je pak zpracování dat rozděleno na více úseků tak, aby vlákno pro zpracování dat průběžně vykonávalo i ostatní méně časově náročné operace.

Další výhodou rozdělení tabulek do oddílů, která je v novém modulu využívána, je časově méně náročná údržba tabulky. „Při mazání starých záznamů, které již nejsou potřeba, je mnohem rychlejší upustit a odstranit celý jeden oddíl rozdělené tabulky než mazat záznam po záznamu. Úklid databáze a aktualizování statistik pro plánovač (Příkaz VACUUM ANAZYLE) je také výhodné volat pouze na jeden vybraný oddíl namísto zavolání na celou tabulku.“ [6] Tímto vybraným oddílem je pak oddíl, který je určený pro naměřená a zpracovaná data za aktuální den, kde je zřejmé, že dochází k velkým změnám v záznamech. Naproti tomu v oddílech starších, než je aktuální den (+ rezerva na dokončení některých operací pro zpracování) se za normálního běhu záznamy již měnit nebudou.

4.2 Datový model pro správu dat

Navržený datový model:

- Poskytuje vhodně formátované tabulky pro uložení naměřených i zpracovaných dat tak, aby jakékoliv operace s těmito daty probíhali optimálně a současně aby data nezabírala na disku více úložného prostoru, než je nutné.
- Uchovává pro obslužnou aplikaci informace o tom, jaké operace s jakými parametry mají být prováděny nad jakými daty.
- Pamatuje si různá nastavení, parametry operací a data, která byla zapsána obslužnou aplikací na podnět nadřazeného systému, který jednal na přání uživatele.

Navržený datový model je rozdělen do tří schémat:

- *data_control* – obsahuje různá nastavení a data pro řídicí logiku modulu
- *data_tables* – úložiště pro naměřená a zpracovaná data
- *evaluated_data* – úložiště pro stavy a události, které byly vyhodnoceny z naměřených nebo zpracovaných dat

Ve schématech *data_control* a *evaluated_data* jsou tabulky vytvořeny a provázány jen jednou, a to při vytvoření databáze. Ve schématu *data_tables* nejsou po vytvoření databáze tabulky žádné. V tomto schématu jsou tabulky vytvářeny a odstraňovány dynamicky za běhu systému v závislosti na aktivitě měřicích přístrojů a podle nastavení operací pro analýzu a zpracování dat ve schématu *data_control*.

4.2.1 Schéma *data_tables*

Schéma *data_tables* je určeno pro tabulky s naměřenými a zpracovanými daty. Naměřená data modul přijímá z adaptéru chytré sítě. Zpracovaná data jsou generována modulem na základě výpočtů z naměřených dat nebo z jiných již dříve zpracovaných dat.

Pro zajištění co nejrychlejšího vykonávání různých SQL dotazů, které budou pracovat s naměřenými a zpracovanými daty, bylo navrženo řešení, kdy každý měřicí přístroj pro každou registrovou mapu, ve které z daného přístroje přicházejí data, má jednu tabulku s naměřenými daty. Jinými slovy má jeden měřicí přístroj alespoň tolik tabulek s naměřenými daty, z kolika registrových map přicházejí jeho naměřená data. Dále podle nastavených operací pro zpracování dat se mohou k přístroji vztahovat i další tabulky, a to se zpracovanými daty. V případě přístrojů s krátkou periodou vzorkování (až 1 sekunda) jsou tabulky s naměřenými a zpracovanými daty rozdělené do oddílů, kterými jsou data automaticky tříděna podle období, do kterých patří. Tím je zajištěna optimální práce s daty i v případě, kdy má tabulka milióny záznamů.

Jak již bylo zmíněno, po vytvoření databáze v rámci nové instalace systému, je schéma *data_tables* prázdné (neobsahuje žádné tabulky), protože modulem pro správu dat ještě nebyla přijata žádná měřidla, a tedy ani žádná data.

První tabulka s naměřenými daty je vytvořena v okamžiku, kdy modul pro správu dat přijme z adaptéru chytré sítě první naměřená data z měřidla, které je již zaevidované jako aktivní. Název tabulky je sestaven tak, aby bylo už na první pohled zřejmé, odkud pocházejí data, které tabulka obsahuje. Název tabulky je tedy složen z identifikačních údajů měřicího přístroje a registrové mapy. Veškerá budoucí naměřená data z tohoto měřicího přístroje, která přijdou v této registrové mapě, budou zapsána do této tabulky. Další tabulky s naměřenými daty jsou vytvářeny po přijetí naměřené hodnoty z jiné registrové mapy nebo z jiného měřidla.

Jedna anebo i více tabulek se zpracovanými daty jsou vytvořeny ve chvíli, kdy dojde k vytvoření nové operace pro zpracování dat. Vytvořené tabulky jsou místa, kam daná operace pro zpracování dat ukládá svoje výsledky, případně mezivýsledky. Všechny nové operace pro zpracování dat nemusejí způsobit vytvoření nových tabulek s daty, ale zpracování dat mohou provádět v rámci již existujících tabulek. Jedna tabulka s daty tak může být současně jak vstupem, tak i výstupem pro několik různých operací pro zpracování dat.

Zpracování dat je více vysvětleno později, a to při popisu schématu *data_control*, kde je také detailněji vysvětlen mechanismus vytváření těchto tabulek s daty včetně vlastností a parametrů, podle kterých jsou generovány.

Vzhledem k tomu, že jeden měřicí přístroj může mít více tabulek s naměřenými i zpracovanými daty, může schéma *data_tables* co do počtu tabulek značně narůst.

Příklad dynamicky vygenerovaného obsahu schématu *data_tables* je vidět na Obr. 4.1, který bude dále v této části podrobně popsán a vysvětlen.

Table Name	Columns
<i>mtsim_bon_wmc_000119_1</i>	<i>tstamp</i> timestamp with time zone « pk », <i>_243</i> real, <i>_244</i> real, <i>_354</i> integer
<i>mtsim_bon_wmc_002033_1</i>	<i>tstamp</i> timestamp with time zone « pk », <i>_243</i> real, <i>_244</i> real, <i>_354</i> integer
<i>mtsim_bon_wmh_002033_2</i>	<i>tstamp</i> timestamp with time zone « pk », <i>_244</i> real, <i>_426</i> integer
<i>mtsim_mtx_em_003014_3</i>	<i>tstamp</i> timestamp with time zone « pk », <i>_1023</i> real, <i>_1024</i> real, <i>_1025</i> real, <i>_1396</i> real, <i>_1398</i> real, <i>energy</i> double precision
<i>mtsim_mtx_em_003014_3_agr30s</i>	<i>tstamp</i> timestamp with time zone « pk », <i>avg_1023</i> real, <i>min_1023</i> real, <i>max_1023</i> real, <i>avg_1024</i> real, <i>min_1024</i> real, <i>max_1024</i> real, <i>avg_1025</i> real, <i>min_1025</i> real, <i>max_1025</i> real, <i>avg_energy</i> double precision, <i>min_energy</i> double precision, <i>max_energy</i> double precision
<i>mtsim_mtx_em_003014_3_arg10m</i>	<i>tstamp</i> timestamp with time zone « pk », <i>avg_1023</i> real, <i>min_1023</i> real, <i>max_1023</i> real, <i>avg_1024</i> real, <i>min_1024</i> real, <i>max_1024</i> real, <i>avg_1025</i> real, <i>min_1025</i> real, <i>max_1025</i> real, <i>avg_energy</i> double precision, <i>min_energy</i> double precision, <i>max_energy</i> double precision

Obr. 4.1 Příklad dynamicky generovaného obsahu schématu *data_tables*

Na Obr. 4.1 je zobrazeno 6 tabulek, které uchovávají data pro tři měřicí přístroje. Měřicí přístroje jsou identifikovány podle svého UID, ze kterého lze postupně vyčíst následující údaje, které jsou v UID přístroje odděleny dvojtečkou:

1. projekt, k jakému měřidlo patří
2. výrobce měřidla
3. typ měřidla
4. sériové číslo měřidla

Na příkladu z Obr. 4.1 jsou tedy data z měřidel *mtsim:bon:wmc:000119*, *mtsim:bon:wmh:002033* a *mtsim:mtx:em:003014*. Názvy tabulek jsou složeny ze tří částí, z nichž každá je oddělená podtržítkem. Tyto části jsou:

1. UID měřidla, kde jsou všechny dvojtečky nahrazeny za podtržítka.
2. Celočíselný identifikátor registrové mapy, ke které naměřená data náleží. Jedná se o lokální identifikátor (ID) v rámci databáze, podle kterého lze dohledat i globální identifikátor (UID), ten však pro název tabulky není vhodný, protože je příliš dlouhý.
3. Přípona, pokud existuje (více v popisu schématu *data_control*).

Každá tabulka s daty obsahuje sloupec „*tstamp*“, který je primárním klíčem. Jedná se o čas, ke kterému jsou naměřená a zpracovaná data vztažena (čas naměřených dat nebo v případě zpracovaných dat například začátek období, za které byla z dat počítána průměrná, minimální a maximální hodnota). Časová značka naměřených dat přichází do systému ve formě unix timestamp. Po příchodu je tato

hodnota předána PostgreSQL databázi, která se současně stará i o správnou reprezentaci času v případě změn času z letního na zimní a naopak. V případě, kdy do systému z nějakého důvodu dorazí duplicitní záznamy, jsou tyto záznamy ignorovány. Duplicitní záznamy jsou záznamy, které jsou již podle časové značky v databázi uloženy. Tato situace může výjimečně nastat například pokud datový koncentrátor nedostane z nadřazeného systému potvrzení o přijatých datech. V takovém případě datový koncentrátor odešle data znovu a znovu čeká na jejich potvrzení.

Tabulky s názvem bez přípony pak obsahují sloupce, které reprezentují jednotlivé vyčítané registry daného přístroje, jejichž hodnoty přicházejí v dané registrové mapě. Název těchto sloupců je složen z podtržítka a UID registru (například „_1023“). Typ sloupce je nastaven podle daného registru.

Tabulky s názvem s příponou obsahují sloupce, které slouží pro uchování nikoliv naměřených, ale zpracovaných hodnot. Hodnoty těchto sloupců nejsou tedy hodnoty registrů v registrové mapě, ale dodatečně počítané hodnoty (zpracovaná data). Názvy těchto sloupců nezačínají podtržítkem. V případě, kdy je pro daný sloupec zpracování dat prováděno nad jedním registrem za účelem hodnotu daného registru za nějaké období nějakým způsobem agregovat, pak název obsahuje agregační metodu, podtržítka a UID registru (například „max_1023“).

Některé tabulky obsahují kombinaci obou druhů sloupců.

Na základě výše uvedených pravidel by již nyní měl být zřejmý zbytek příkladu vygenerovaného obsahu schématu *data_tables* z Obr. 5.1. Z obrázku je poznat následující:

- Přístroj *mtsim:bon:wmc:000119* má v databázi pouze jednu tabulku s daty a to *mtsim_bon_wmc_000119_1*. Data z tohoto měřicího přístroje chodí do modulu v registrové mapě s lokálním identifikátorem 1 a tato mapa obsahuje celkem 3 registry (2 typu real a 1 typu integer).
- Přístroj *mtsim:bon:wmh:002033* má v databázi dvě tabulky s daty a to *mtsim_bon_wmh_002033_1* a *mtsim_bon_wmh_002033_2*. Data z tohoto měřicího přístroje chodí do modulu ve dvou různých registrových mapách, jejichž lokální identifikátory jsou 1 a 2. Tabulka odpovídající registrové mapě s lokálním identifikátorem 1 má stejné sloupce, jako tabulka pro přístroj *mtsim:bon:wmc:000119*, protože z obou přístrojů přicházejí data ve stejné registrové mapě. Data jsou však oddělena tak, aby měl každý přístroj pro každou registrovou mapu jednu tabulku s daty.
- Přístroj *mtsim:mtx:em:003014* má v databázi tři tabulky s daty a to *mtsim_mtx_em_003014_3*, *mtsim_mtx_em_003014_3_arg30s* a *mtsim_mtx_em_003014_3_arg10m*. Data z tohoto měřicího přístroje chodí do modulu pouze v jedné registrové mapě s lokálním identifikátorem 3 a tato

mapa obsahuje celkem 5 registrů z nichž každý je typu real. Hodnoty registrů této registrované mapy tohoto přístroje jsou ukládány do tabulky *mtsim_mtx_em_003014_3*, která obsahuje navíc jeden sloupec s názvem „energy“ pro zpracovaná data, která jsou počítána periodicky podle zvolené operace. Tabulka *mtsim_mtx_em_003014_3_arg30s* obsahuje agregovaná data za období 30 sekund, jak lze poznat podle konce názvu tabulky (více informací v popisu schématu *data_control*). Těmito agregovanými daty jsou pro tři registry a sloupec „energy“ jejich minimální, maximální a průměrné hodnoty za období 30 sekund. Tabulka *mtsim_mtx_em_003014_3_arg10min* obsahuje stejná agregovaná data, ale za období 10 minut.

4.2.2 Schéma *data_control*

Schéma *data_control* obsahuje všechny informace, které modul pro správu dat potřebuje pro svoji činnost a současně informuje nadřazený systém především o stavu modulu pro správu dat, jeho možnostech a dostupných operacích s daty tak, aby mu mohl zadávat různé pokyny a přizpůsobovat si jeho činnost. Více bylo uvedeno v požadavcích na modul pro správu dat v kapitole 3.4.2 Komunikace s nadřazeným systémem. Možnosti schématu *data_control* lze stručně uvést následovně:

- Narozdíl od schématu *data_tables*, které obsahuje tabulky s naměřenými a zpracovanými daty, schéma *data_control* obsahuje podrobné informace o těchto tabulkách včetně pravidel, podle kterých jsou tyto tabulky vytvářeny.
- Schéma informuje o tom, jaké různé typy tabulek a typy operací mohou být vytvořeny a k jakým tabulkám je povoleno přiřadit jaké operace zpracování dat a vyhodnocení stavů a událostí z dat.
- Schéma popisuje, jaké operace pro analýzu, zpracování a vyhodnocení dat se mají vykonat nad jakými daty, včetně toho, kdy nebo s jakou periodou se mají vykonávat a s jakými parametry.
- Pro účely ladění obsahuje tabulky pro uložení SOAP zpráv přijatých z adaptéru chytré sítě.

Schéma *data_control* je zobrazeno na Obr. 4.2 a Obr. 4.3 a dále bude podrobně popsáno.

4.2.2.1 Správa naměřených a zpracovaných dat

V tabulce *data_table* jsou záznamy o existujících tabulkách s naměřenými a zpracovanými daty. Na jeden záznam tabulky *data_table* současně připadá jedna tabulka s daty ve schématu *data_tables*. Při vytvoření, respektive odstranění záznamu je současně vytvořena, respektive odstraněna i odpovídající tabulka s daty

ve schématu *data_tables*. Zatímco ve schématu *data_tables* se nachází celá tabulka se všemi odpovídajícími naměřenými a zpracovanými daty, záznam v tabulce *data_table* tuto tabulku s daty ze schématu *data_tables* propojuje s celým zbytkem datového modelu. Přes tento záznam lze velmi rychle určit informace o datové tabulce, kterou záznam reprezentuje:

- Z jakého zařízení všechna data pochází – odkazem na tabulku *device*, která obsahuje všechna objevená zařízení.
- Z jaké registrové mapy všechna data pochází – odkazem na tabulku *register_map*, která obsahuje seznam všech registrových map všech měřidel.
- Jaký je přesný název tabulky s daty ve schématu *data_tables* a z čeho byl název poskládaný.
- Zda je tabulka s daty rozdělená do oddílů (declarative partitioning). Pokud ano, pak jaká je perioda oddílů, po jakém čase je mazat a jak dlouho dopředu je vytvářet. Odkazem na tabulku *data_table_partition* lze zjistit podrobné informace o oddílech, jako jejich název a přesný časový rozsah dat, pro který je oddíl určen.
- Zda tabulka s daty obsahuje naměřená nebo zpracovaná data.
 - Pokud se jedná o naměřená data, pak z jakých registrů data pochází – odkazem přes již zmíněnou tabulku *register_map* na tabulku *register_map_has_register*, která obsahuje seznam všech registrů, ze kterých někdy došla data a k jaké registrové mapě daný registr patří.
 - Pokud se jedná o naměřená data, pak z jaké SOAP zprávy jednotlivá data pochází – odkazem přes tabulku *soap_log_has_data* na tabulku *soap_log*, která obsahuje veškeré informace o přijatých SOAP zprávách a je primárně určena pro ladění.
 - Pokud se jedná o zpracovaná data, pak jakou operací jsou data zpracovávána. Jinými slovy, pro jakou operaci analýzy a zpracování dat jsou data z této tabulky výstupním argumentem a jaká data to jsou.
- Podrobnější informace o jednotlivých sloupcích tabulky – odkazem na tabulku *data_table_column*, která obsahuje názvy sloupců, jejich datový typ a jednotku veličiny, jejíž hodnoty jsou do daného sloupce ukládány.
- Pro jakou operaci zpracování a vyhodnocení dat jsou data z této tabulky vstupním argumentem a jaká data to jsou.
- Jakého typu daná tabulka s daty je, z čehož lze dále zjistit všechny kompatibilní operace zpracování a vyhodnocení dat. Kompatibilní operace jsou operace, které mohou být k dané tabulce s daty přiřazeny.

Tabulky s naměřenými a zpracovanými daty jsou vždy nějakého typu. Tyto typy jsou definovány v tabulce *data_table_type*. Modul pro správu dat vytváří nové

tabulky s naměřenými a zpracovanými daty vždy podle předpisů, které jsou v tabulce *data_table_type* uloženy. Vytvořená tabulka s daty má pak nějaké vlastnosti charakteristické pro daný typ. Podle typu se od sebe tabulky naměřených a zpracovaných dat mohou lišit:

- Mírou zpracování dat – syrová naměřená data (nezpracovaná data), vypočtená data, mezi výpočty, smíšená data
- Zvoleným optimalizačním řešením – pro zařízení s více registrovými mapami, pro zařízení s krátkou periodou vzorkování (až 1 sekunda)
- Účelem – prezentace dat, škálovatelnost dat, vyhodnocení stavu, mezi výpočty
- Povolenými typy operací pro analýzu, zpracování a vyhodnocení dat.

4.2.2.2 Správa operací pro zpracování dat a vyhodnocení stavů a událostí

Více možných typů tabulek s naměřenými a zpracovanými daty v některých případech značně ztěžuje návrh a implementaci operací pro analýzu, zpracování a vyhodnocení dat. Především vzhledem k optimalizacím někdy není možné vyhnout se situacím, kdy operaci s daty přesně specializovanou pro jeden typ tabulek nebude možné použít pro jiný typ tabulek. Odpovědí na tento problém je tabulka s poněkud dlouhým názvem *data_processing_type_compatiblewith_data_table_type*, která definuje, jaké typy operací pro zpracování a vyhodnocení dat je nad jakými typy tabulek s daty povoleno provádět a jaké se mají vytvářet automaticky ihned po vytvoření tabulky dat daného typu.

Tabulka *data_processing_type_compatiblewith_data_table_type* je dále propojena s tabulkou *data_processing_type*, která definuje různé typy operací pro zpracování a vyhodnocení dat včetně pořadí, v jakém by se měli dané typy operací provádět a jak často by se měli provádět.

Každý záznam v tabulce *data_processing_type* je jeden typ operace s daty. Definované typy operací by měli odpovídat požadavkům na zpracování a vyhodnocení dat, které byly definované v kapitole 3.4.3 Automaticky vykonávané operace s daty.

Ke každému typu operace by měla být v obslužné aplikaci modulu pro správu dat naimplementována funkcionality taková, že by aplikace měla být schopna vytvořit novou operaci daného typu a periodicky ji vykonávat.

Vytvoření operace s daty je velmi komplexní záležitost a podle typu vytvářené operace může dojít k vytvoření různých záznamů v různých tabulkách. Navíc některé operace je možno parametrizovat z nadřazeného systému, jak bylo uvedeno v požadavcích na modul v kapitole 3.4.2 Komunikace s nadřazeným systémem. Seznam všech tabulek, do kterých mohou být vloženy nové záznamy při vytváření nové operace:

- *data_processing* – Jednotlivé operace zpracování dat a vyhodnocení stavů a událostí z dat, které má modul pro správu dat okamžitě nebo periodicky vykonávat. Obsahuje informace o tom, kdy byla operace naposledy vykonána a čas posledních dat, nad kterými byla vykonána. Dále i informaci o stavu vykonání. Počet operací může značně narůst. Například když bude mít modul pro správu dat za úkol kontrolovat spotřebu v daných mezích pro 10 měřicích přístrojů, vznikne zde 10 operací pro vyhodnocení dat z nichž každá bude mít v argumentu (*dpa*, *dpa_data*) jiná data k vyhodnocení. Při vytvoření nové operace bude do této tabulky vložen vždy jeden nový záznam.
- *dpa* – Celým názvem „Data Processing Argument“, je argumentem pro operaci nad daty. Jedna operace může mít více argumentů. Obsahuje pořadí argumentu v rámci dané operace a typ argumentu. Podle typu argumentu lze určit, zda se jedná o výběr dat pro operaci (*dpa_data*) nebo zda je operace parametrizována (*dpa_formula*). Při vytvoření nové operace bude do této tabulky vložen jeden nebo více nových záznamů.
- *dpa_data* – Specifikuje, nad jakými daty se má operace provádět. Přesněji řečeno nad jakou tabulkou s daty se má provádět (odkaz na tabulku *data_table*) a přesné označení nových dat, která teprve čekají na zpracování (odkazování z tabulky *dpa_data_to_process*). Při vytvoření nové operace bude do této tabulky vložen jeden nebo více nových záznamů.
- *dpa_formula* a všech 7 tabulek, které mají tento název jako prefix – Nabízí možnosti si operace přizpůsobit. Při vytvoření nové operace do těchto tabulek nemusí být vloženy vůbec žádné nové záznamy, nebo jich může být vloženo i hodně. To záleží především na typu operace. Každý typ operace je možné si přizpůsobit jinak, což je ovlivněné především samotnou implementací jednotlivých typů operací. Pomocí těchto tabulek je umožněno především následující:
 - Pro měřicí přístroj nebo tabulku s daty vybrat registry, s jejichž naměřenými hodnotami bude operace pracovat.
 - Pro tabulku s daty vybrat její jednotlivé sloupce, s jejichž hodnotami bude operace pracovat.
 - Zvolit odběrné místo, pro které je operace prováděna.
 - Zadat korekci zpracovávané hodnoty.
 - Zadat offset a násobitel k vybraným naměřeným a zpracovávaným hodnotám a pro vyhodnocené stavy spotřeb odběrných míst.
 - Parametrizovat operace pro vyhodnocení událostí z naměřených dat. Například zadání maximální hodnoty pro operaci pro detekci odběru mimo stanovené meze.

4.2.3 Schéma *evaluated_data*

Schéma *evaluated_data* obsahuje tabulky pro uložení stavů a událostí, které byli vyhodnoceny z naměřených a zpracovaných dat. Schéma je zobrazeno na Obr. 4.4 a dále bude podrobně popsáno.

4.2.3.1 Úložiště pro vyhodnocené události

Výstup z operací pro vyhodnocení událostí z naměřených a zpracovaných dat je složen ze dvou částí. Základní část je ukládána vždy do tabulky *event* a obsahuje následující informace:

- V rámci jakého odběrného místa a měřicího přístroje událost nastala.
- Kdy událost nastala (kdy byla z naměřených dat vyhodnocena) a kdy byla přijata nadřazeným systémem.
- Typ události.

Detailní informace o nastalé události jsou podle typu události ukládány do jedné z následujících tabulek:

- *event_dataoutofrange* – hodnota (odběr, statistické údaje atd.) mimo stanovené meze
- *event_datadecreased* – pokles hodnoty (stočení měřidla)
- *event_continuousconsumption* – nepřetržitá spotřeba (detekce úniku vody, plynu atd.)
- *event_datahavenotarrivedforawhile* – chybí data, ztráta komunikace

4.2.3.2 Úložiště pro vyhodnocené stavy spotřeb odběrných míst

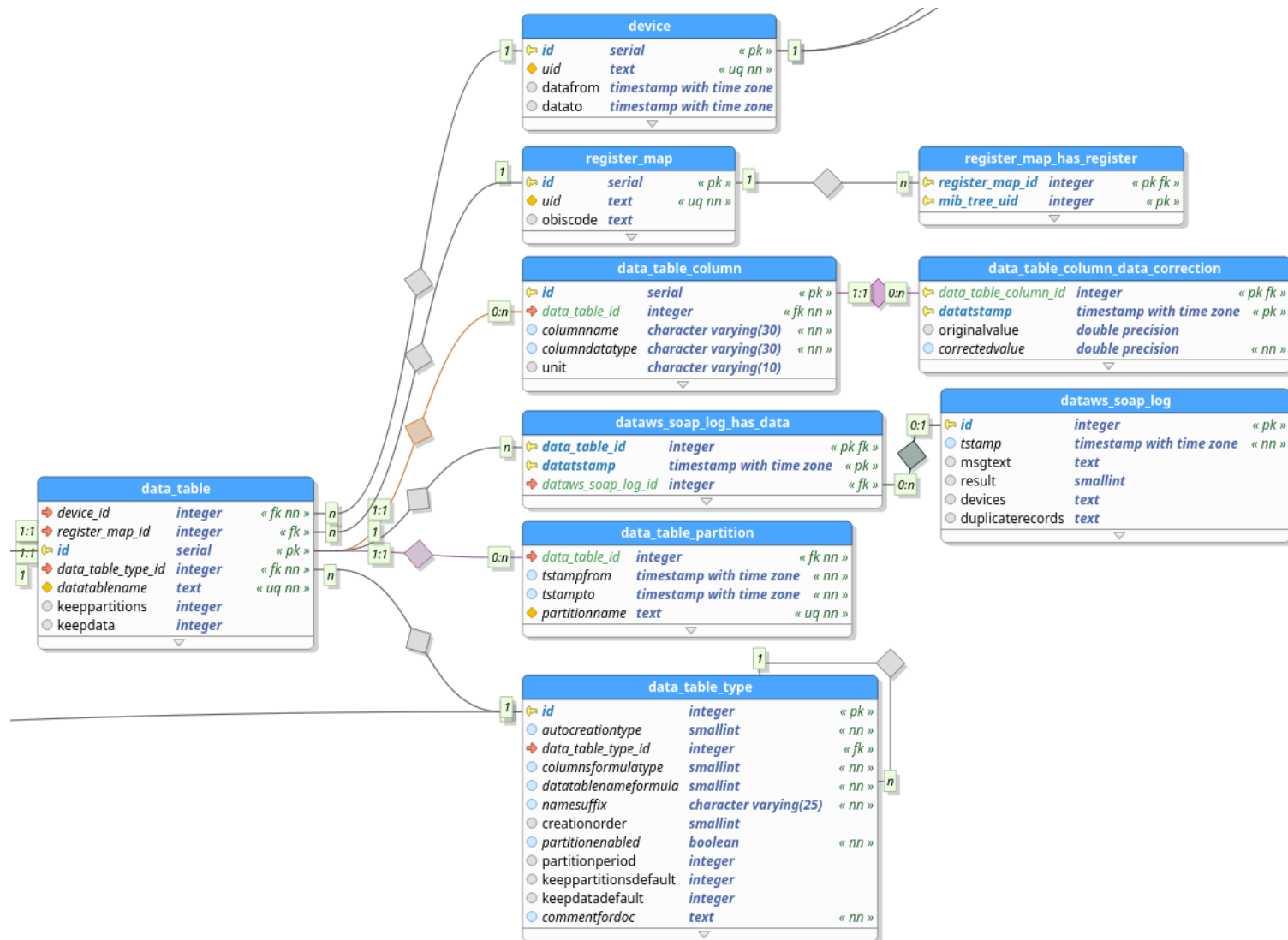
Výstup z operací pro vyhodnocení stavů spotřeb odběrných míst je ukládán do tabulky *mp_consumptionstate*. Tabulka obsahuje všechna důležitá data pro zobrazení stavů spotřeb:

- datum denního stavu spotřeby nebo datum začátku delšího období v případě týdenních, měsíčních a ročních stavů spotřeb
- naměřená hodnota a čas měření
- spotřeba za dané období
- rozdíl spotřeb mezi aktuálním a předchozím obdobím
- stav daného stavu spotřeby (dokončeno, zpracovává se, průměrná spotřeba, chybí data)

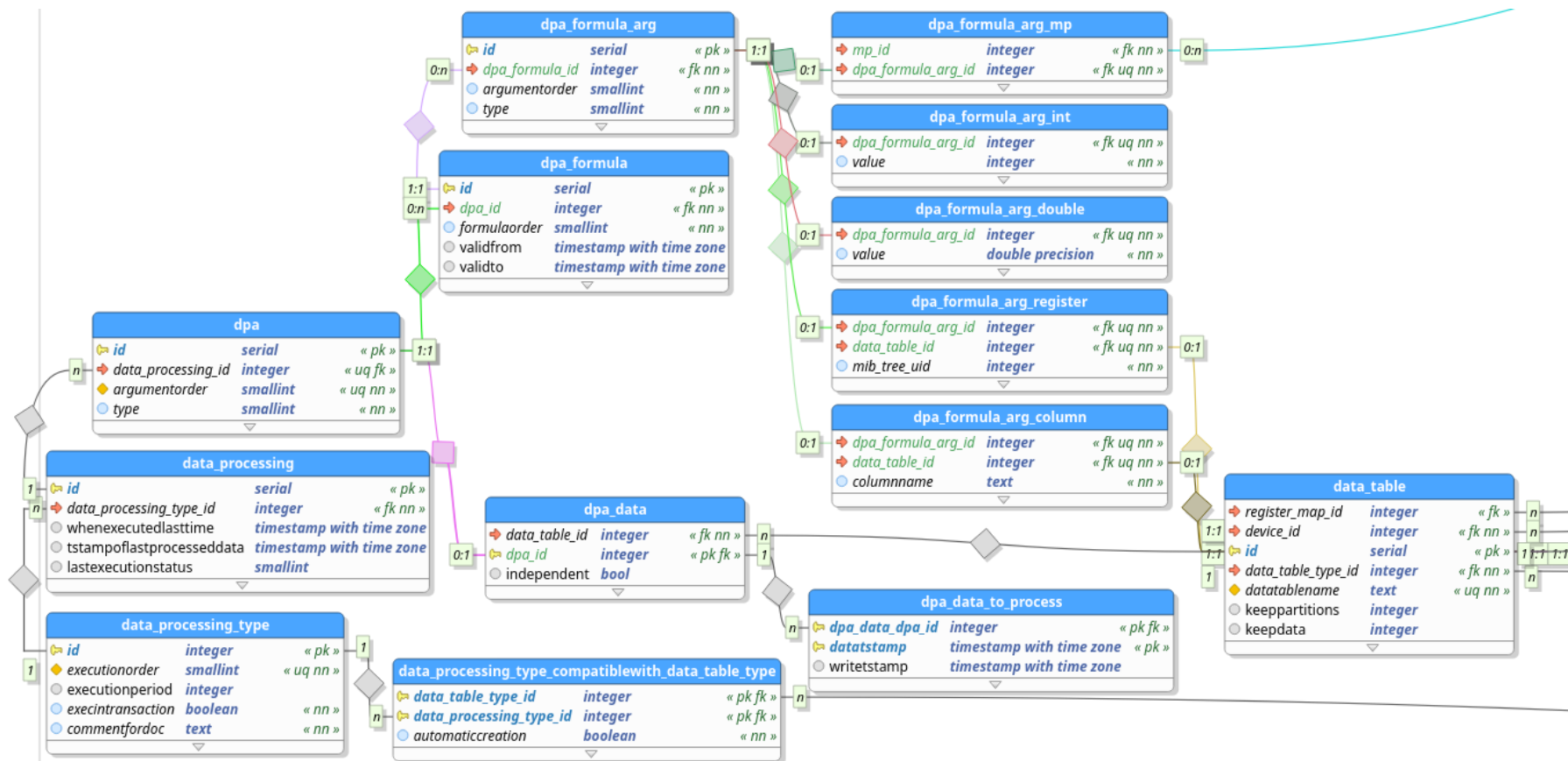
K odběrným místům mohou být kromě stavů spotřeb přiřazeny i události od uživatelů. Těmito událostmi jsou nastavení offsetu a násobitele sledované veličiny a přiřazení, vyřazení a výměna měřicího přístroje se zadáním stavu sledované veličiny v době manipulace s měřidly. Tyto události ovlivňují i samotné vyhodnocování stavů

spotřeb z naměřených dat. Tyto události odběrných míst jsou zadávány uživatelem v nadřazeném systému, který je v SOAP zprávě odešle do modulu pro správu dat, který zprávu zpracuje, upraví operace, kterých se změny týkají (jak již bylo popsáno výše přidá nebo upraví záznamy v tabulkách začínajících názvem *dpa_formula*) a nakonec pro pohodlné a optimální čtení všech dat souvisejících s odběrnými místy vloží odpovídající záznamy do tabulek *mp_event_values_by_user*, *mp_event_by_user* a *mp_consumptionstate_has_mp_event_by_user*.

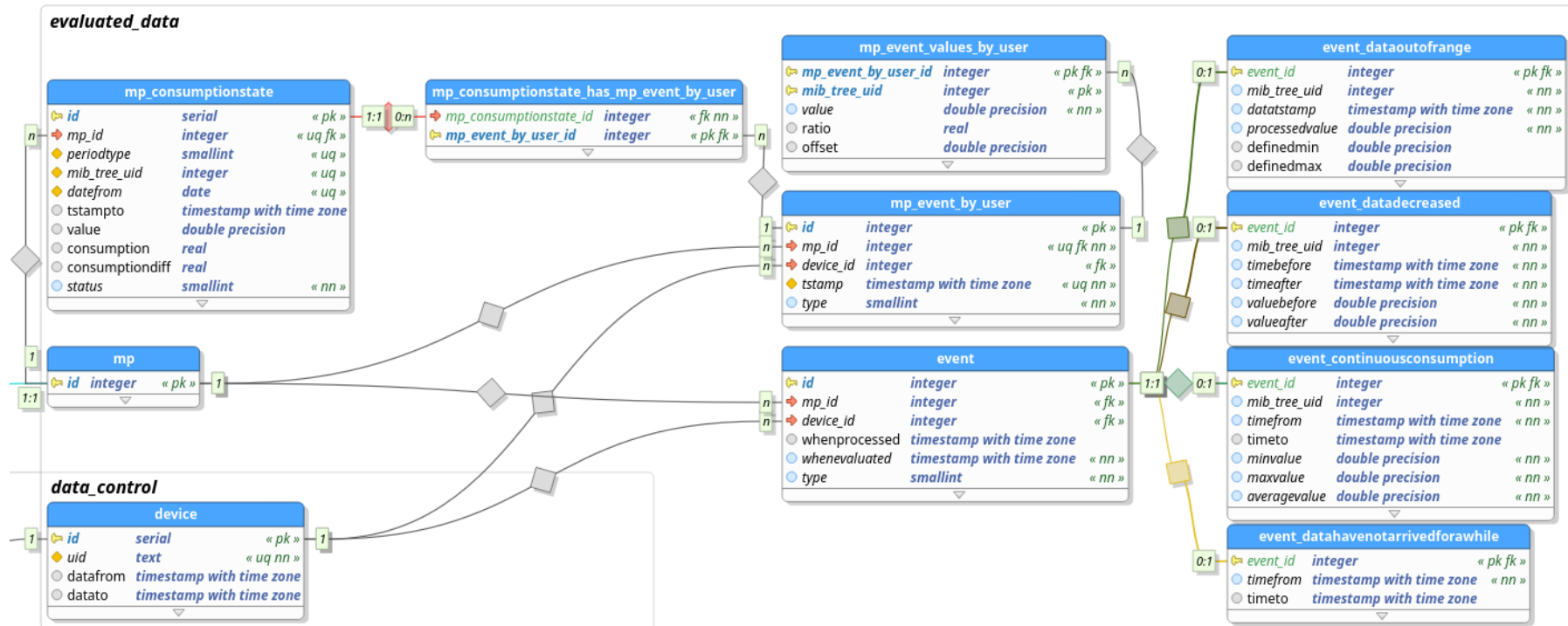
Ačkoliv modul pro správu dat má na starosti data odběrných míst, samotná odběrná místa jsou spravována nadřazeným systémem. Z toho důvodu je v modulu ve schématu *evaluated_data* tabulka *mp*, která obsahuje identifikátory jednotlivých odběrných míst z nadřazeného systému. Jednotlivé identifikátory jsou vytvářeny ve chvíli, kdy modul pro správu dat dostane z nadřazeného systému pokyn k provádění operace pro vyhodnocování stavů nad novým odběrným místem.



Obr. 4.2 Schéma *data_control* datového modelu pro správu dat – část 1



Obr. 4.3 Schéma *data_control* datového modelu pro správu dat – část 2



Obr. 4.4 Schéma *evaluated_data* datového modelu pro správu dat

4.3 Obslužná aplikace

Úkolem navržené obslužné aplikace je přijímání dat z adaptéru chytré sítě, periodické provádění definovaných operací s daty a komunikace s nadřazeným systémem pro monitorování spotřeb energií.

Vývojové diagramy obslužné aplikace lze nalézt v příloze na CD.

4.3.1 Přijímání dat z chytré sítě

Z adaptéru chytré sítě aplikace přijímá SOAP zprávy s naměřenými daty. Adaptér je informován o přijetí těchto zpráv.

V rámci analýzy jsou příchozí data z chytré sítě tříděna do patřičných tabulek a současně jsou některá z nich označena jako data čekající na zpracování. Vždy je přesně specifikováno jaká operace má zpracovat jaká data.

4.3.2 Provádění definovaných operací s daty

Tyto operace provádí zpracování dat a vyhodnocení stavů a událostí z těchto dat a to vždy podle definovaných nastavení v datovém modelu:

- V rámci zpracování dat jsou periodicky v definovaných intervalech prováděny definované operace s daty čekajícími na zpracování. Výsledkem jsou různé statistické údaje, veličiny vypočtené na základě jiných veličin nebo mezivýsledky, tedy hodnoty určené pro další zpracování.
- V rámci vyhodnocení jsou podobně jako u zpracování dat prováděny definované operace s daty čekajícími na zpracování. Výsledky těchto operací jsou vyhodnocené stavy nebo události.

Aplikace si všechny dotazy do databáze předpřipravuje a také na vybrané tabulky pravidelně volá službu pro úklid a databázový kolektor pro sběr statistik o databázových tabulkách (příkaz `VACUUM ANALYZE`), „čímž aktualizuje databázového plánovače tak, aby mohl vždy zvolit optimální způsob provedení SQL dotazů.“ [6]

4.3.3 Komunikace s nadřazeným systémem

Mezi aplikací a nadřazeným systémem bude probíhat komunikace ve formě SOAP zpráv. Aplikace musí být schopná plnit pokyny nadřazeného systému a také ho informovat o nově vyhodnocených událostech. Všechny možnosti komunikace byly popsány v kapitole 3.4.2 Komunikace s nadřazeným systémem.

5 NÁVRH NOVÝCH MOŽNOSTÍ PREZENTACE NAMĚŘENÝCH DAT

V předchozí kapitole byl navržen modul pro správu dat, který dokáže velmi rychle analyzovat, ukládat a zpracovávat i záznamy veličin měřených v každé sekundě. V této kapitole je vypracován návrh jak tato naměřená data prezentovat uživateli takovým způsobem, aby se v nich mohl snadno a rychle orientovat. Nejnáročnějším případem jsou sekundové záznamy za období až 6 měsíců, což odpovídá přibližně 15miliónům záznamům.

5.1 Stávající možnosti zobrazení dat a návrh na jejich zlepšení

Aktuálně je systém dostatečně schopen prezentovat denní, týdenní, měsíční a roční spotřeby v tabulkách i grafech. Vždy se však jedná o zobrazení maximálně několika desítek hodnot na jednu stránku tabulky nebo na jeden graf. Například zobrazení denních spotřeb za 1 měsíc (30 hodnot) nebo týdenních spotřeb za 1 rok (52 hodnot). Pro takové případy může prezentace dat zůstat beze změny.

Složitější situace nastává v případě potřeby zobrazit všechna naměřená data. To se týká jak hodnot kumulativních průběhů veličin, ze kterých se denní, týdenní, měsíční a roční spotřeby počítají, tak i okamžitých hodnot veličin. Tato data jsou zatím měřena nejčastěji jednou za 15 minut a jejich zobrazení je možné pouze v tabulce, kde je možné zobrazit v jeden okamžik pouze hodnoty jedné veličiny. To je vidět na Obr. 5.1. Takové zobrazení je velice nepřehledné i pro záznamy po 15minutách. Pro sekundové záznamy, pro které je systém v rámci této práce optimalizován, by přehlednost byla ještě horší.

Datum a čas	Hodnota	Typ hodnoty	Registrová mapa	Stav	Detail
09.04.2019 10:30:20	1.700	real	3bb3065189ea4eed64e6c7518f9d247f	R	
09.04.2019 10:30:16	1.700	real	3bb3065189ea4eed64e6c7518f9d247f	R	
09.04.2019 10:30:12	1.700	real	3bb3065189ea4eed64e6c7518f9d247f	R	
09.04.2019 10:30:08	1.700	real	3bb3065189ea4eed64e6c7518f9d247f	R	

Zobrazuji 1 801 až 1 804 z celkem 1 804 záznamů Zobraz záznamů 10

První Předchozí 1 ... 177 178 179 180

181 Další Poslední

Obr. 5.1 Aktuální nepřehledné zobrazení naměřených hodnot měřicího přístroje

Z Obr. 5.1 je patrné, že stávající zobrazování naměřených hodnot měřidel není cílené na koncového uživatele, ale je cílené na vývojáře a technické pracovníky firmy ModemTec, kteří si pomocí tabulky mohou zkontrolovat, zda data přicházejí do systému správně. Zobrazení hodnot v tabulce by mělo vypadat podle návrhu na Obr. 5.2, kde je pro zvolený měřicí přístroj možné zvolit registrovou mapu a období a poté se v tabulce zobrazí naměřené hodnoty všech veličin, které k dané registrové mapě náleží.

Měřicí přístroj: <UID měřicího přístroje>

Registrová mapa: <UID registrové mapy> ▾ Období: <Den> ▾ Datum: 04.02.2019 📅

Unix Timestamp	Datum a čas	Veličina 1 [Jednotka 1]	Veličina 2 [Jednotka 2]	Veličina ... [Jednotka ...]	Veličina N [Jednotka N]
1556402580	27.4.2019 22:03:00	hodnota	hodnota	hodnota	hodnota
1556401680	27.4.2019 21:48:00	hodnota	hodnota	hodnota	hodnota
1556400780	27.4.2019 21:33:00	hodnota	hodnota	hodnota	hodnota
...
1556385480	27.4.2019 17:18:00	hodnota	hodnota	hodnota	hodnota

Zobrazení 1 až 20 z celkem 89 záznamů. Předchozí 1 2 3 4 5 Další

Obr. 5.2 Návrh zobrazení naměřených hodnot v tabulce

5.2 Výběr vhodného pluginu pro prezentaci dat v grafech

Pro vytváření grafů a prezentaci dat v těchto grafech byl zvolen plugin jqPlot. Jedná se o plugin knihovny jQuery jazyka JavaScript. Grafy jqPlot jsou mezi open-source grafy velmi oblíbené, mají dobrou dokumentaci a nabízí spousty různých možností pro zobrazení včetně pokročilých funkcí jako je přibližování, proxy grafy a další. Nevýhodou pluginu je, že nabízí řešení pouze pro 2D grafy, nikoliv pro 3D. Použití 3D grafů však není pro tento systém plánováno.

5.3 Návrh nových možností zobrazení dat v grafech

Pomocí pluginu jqPlot budou přehledně v jednom grafu zobrazovány až 4 průběhy současně. Průběhy budou barevně odlišeny a bude možné je podle potřeby schovávat a ukazovat. Grafy bude možné si kdykoliv a kdekoliv přiblížit a oddálit. Měřítko os se budou měnit automaticky.

Především pro měřidla s vysokou hustotou vzorkování (až 1 sekunda) mohou nastat situace, kdy bude dat pro zvolené přiblížení (oddálení) grafu takové množství, že by jeden pixel na obrazovce reprezentoval mnoho hodnot. Tyto situace

budou řešeny zobrazením agregovaných dat. Agregovanými hodnotami je průměr, minimum a maximum z hodnot za dané období. V tomto případě tedy bude graf zobrazovat namísto jednoho průběhu tři (odpovídající minimálním, maximálním a průměrným hodnotám).

Dlouhé průběhy s velkým množstvím dat budou prezentovány za pomoci proxy grafů, kdy v jednom grafu bude průběh přiblížený a ve druhém grafu oddálený. Kliknutím na bod v oddáleném grafu se zobrazí daná oblast v přiblíženém grafu.

Na jedné stránce bude tedy zobrazeno více grafů, z nichž první graf bude oddálen nejvíce a bude tedy zobrazovat nejdelší období pomocí agregovaných dat za nejdelší intervaly. Druhý graf bude o něco přiblížen a bude tedy zobrazovat o něco kratší období a to pomocí agregovaných dat za o něco kratší intervaly. Třetí graf bude ještě více přiblížen a tak dále až k poslednímu grafu, který už nebude zobrazovat agregovaná data, ale bude zobrazovat původní (originální, syrová) data.

Celá funkcionalita by se dala přirovnat k pohybu v mapách, kdy je k dispozici mini mapa, hlavní mapa a dále GPS souřadnice a přiblížení (zoom) hlavní mapy. Souřadnice hlavní mapy je možné ovládat pomocí mini mapy. Přiblížení mapy určuje jaké dlaždice mapy se uživateli vykreslí. V případě grafů je souřadnicí čas a oddálení grafu určuje velikost zobrazovaného období a jaká agregovaná data se uživateli vykreslí (čím větší oddálení, tím jsou data agregovaná po větších intervalech; největší přiblížení tedy odpovídá syrovým datům bez agregace). Efektu „mini mapy“ je dosaženo právě zobrazením více grafů současně, kdy jsou jednotlivé grafy mezi sebou interaktivní a je možné z jednoho grafu ovládat zobrazované období jiného grafu.

Počet oddálení (a tedy i počet agregačních úrovní) je třeba dopředu dobře nadefinovat, protože v době zobrazování grafů již musí být dopředu všechna agregovaná data k dispozici, aby uživatel nemusel na nic dlouho čekat. Počet úrovní přiblížení bude vždy záviset na periodě měřených dat. Například sekundové záznamy budou potřebovat více úrovní oddálení než záznamy 30minutové. Především u sekundových záznamů je třeba při návrhu prezentování dat dělat kompromisy mezi technickými omezeními a tím, co by bylo nejlepší pro uživatele. Především je nutné:

- Nadefinovat agregování dat v nějakých rozumných intervalech, které mohou uživateli systému připadat zajímavé. Nestačí se tedy pouze zaměřit na plynulost přechodů mezi jednotlivými úrovněmi oddálení. Například je lepší použít hodinová agregovaná data, protože budou pro uživatele systému zajímavější než tříhodinová, která by ale naopak mohla lépe přispět k celkově plynulejšímu přechodu oddálení grafů od nejnižší úrovně po nejvyšší (při zachování celkového počtu úrovní oddálení).

- Určit celkový počet úrovní oddálení. Více úrovní zajistí pro uživatele plynulejší přechod mezi nimi. Současně je třeba brát na vědomí, že každá úroveň oddálení zabere místo na disku, protože pro každou úroveň oddálení budou generována agregovaná data za různě velké intervaly.
- Promyslet maximální počet zobrazovaných bodů na jeden graf. Příliš mnoho bodů zbytečně zpomalí prohlížeč na straně klienta, reakci grafu na akce uživatele, prodlouží dobu zobrazování grafu a může zpomalit i server. Příliš málo bodů nebude mít vypovídající hodnotu a bude nutit uživatele příliš zběsile aktualizovat zobrazení grafu pro různá období.
- Zvolit maximální zobrazované období pro jednotlivé úrovně oddálení. Opět by měl být volen kompromis mezi obdobími, které bude dávat uživateli největší smysl (například 5 minut, hodina, den, týden, měsíc) a obdobími, které bude nejvhodnější pro splnění již výše diskutovaných potřeb.

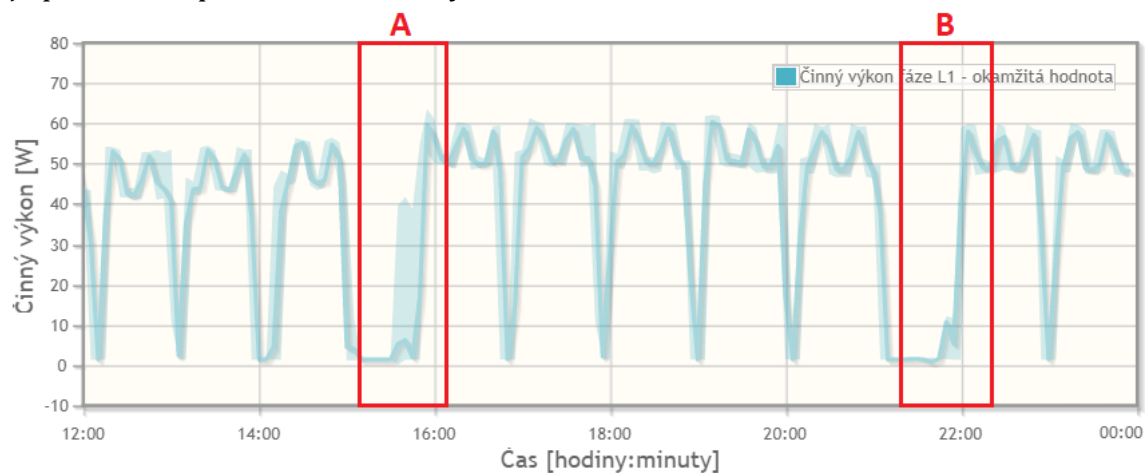
Tab. 5.1 zobrazuje návrh nastavení úrovní oddálení pro zobrazení sekundových záznamů v grafech. Při návrhu byla snaha volit smysluplné intervaly a současně se držet kolem 300 bodů na vykreslení jednoho průběhu v grafu. V případě jiných záznamů než sekundových (30sekundové, 15minutové, 30minutové, atd.), by došlo pouze k odpovídajícímu posunu a redukci některých úrovní. Pro některé periody měřených dat může být užitečné i upravit zobrazovaná období a intervaly agregace dat.

Tab. 5.1 Návrh nastavení úrovní oddálení pro zobrazení sekundových záznamů

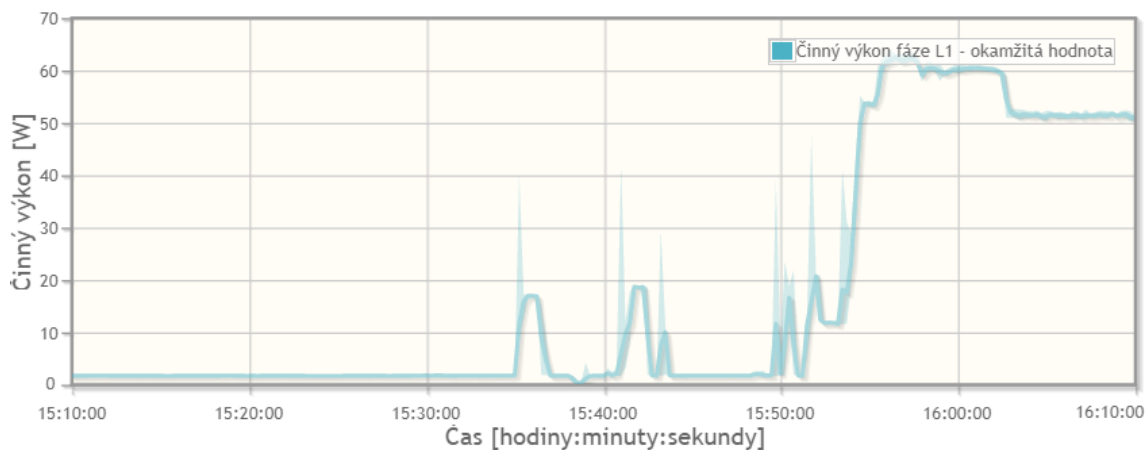
Úroveň oddálení grafu	Za jaké intervaly jsou data v grafu agregovaná	Zobrazené období	Počet bodů na 1 průběh
0 (bez oddálení)	- (původní data bez agregace)	5 minut	300
1	15 sekund	1 hodina	240
2	5 minut	1 den	288
3	1 hodina	10 dní	240
4	1 den	1 rok	365

Užitečnost použití více úrovní oddálení s různými agregovanými daty lze uvést na následujícím příkladu. Na Obr. 5.3 je zobrazen průběh okamžité hodnoty činného výkonu za období 12 hodin. Zobrazení je provedeno pomocí 5minutových agregovaných dat. V případě sekundových záznamů to znamená, že jedna hodnota v tomto grafu je dopředu vypočítaná ze 300 záznamů. Z těchto 300 záznamů je tmavší barvou zobrazena průměrná hodnota za 5 minut. Světlejší barvou je zobrazeno pásmo mezi minimální a maximální hodnotou za daných 5 minut. Čím je tento rozdíl větší, o to větší je rozsah, v jakém se pohybují hodnoty za těchto 5 minut.

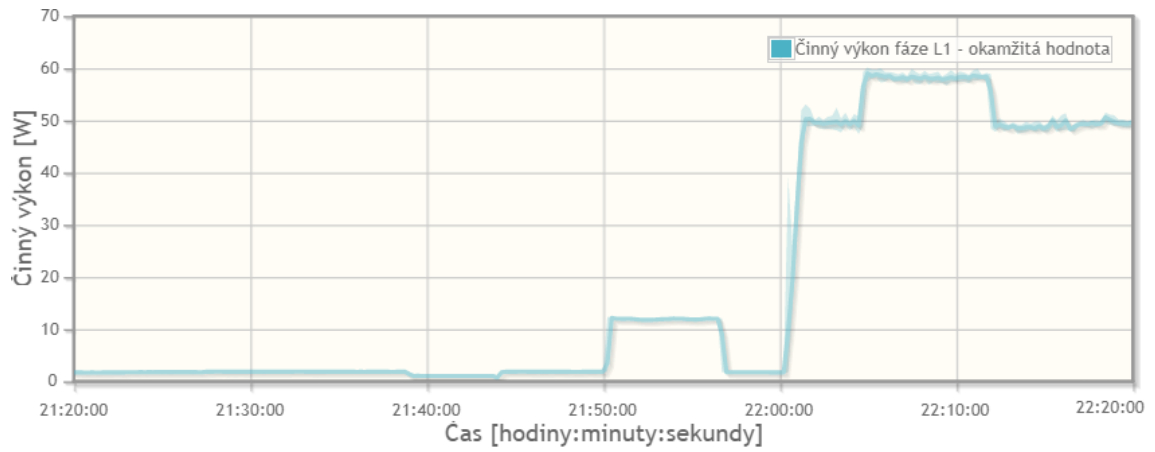
Pokud je toto pásmo širší, znamená to, že se hodnoty mění za delší časový úsek, než za 5 minut. Uživatel si tak může v krátkém čase prozkoumat průběh i za několik dní a rychle v něm vyhledat úseky, které pro něj mohou být nějakým způsobem zajímavé. Na Obr. 5.4 a Obr. 5.5 je zobrazen stejný průběh, ale pouze za období 1 hodinu a je zobrazen pomocí agregovaných dat za 15 sekund. Období odpovídá zvoleným úsekům „A“ a „B“ z Obr. 5.3. Je vidět, že už z oddáleného grafu lze vyčíst, ve kterém úseku se hodnoty mění a také odhadnout jakým způsobem. Po přiblížení je pak možné prozkoumat detaily.



Obr. 5.3 Příklad zobrazení průběhu pomocí 5minutových agregovaných dat.



Obr. 5.4 Zobrazení průběhu za vybraný úsek „A“ z Obr. 5.3. Průběh je zobrazen za hodinu pomocí 15sekundových agregovaných dat.



Obr. 5.5 Zobrazení průběhu za vybraný úsek „B“ z Obr. 5.3. Průběh je zobrazen za hodinu pomocí 15sekundových agregovaných dat.

6 IMPLEMENTACE NAVRŽENÉHO ŘEŠENÍ

V rámci optimalizací systému a implementace nových možností prezentace dat:

- Byla naimplementována obslužná aplikace modulu pro správu dat.
- Byl vytvořen instalační balíček modulu pro správu dat.
- Bylo naprogramováno přehledné prohlížení naměřených a zpracovaných dat, které je vhodné i pro velké objemy dat a dlouhé časové úseky.
- Byly vylepšeny stávající možnosti zobrazování dat.
- Bylo optimalizováno zobrazování spotřeb.

Pro implementaci navrženého řešení bylo použito několik programovacích jazyků, jak již bylo dříve v této práci diskutováno. Využití jednotlivých programovacích jazyků shrnuje tabulka Tab. 6.1.

Tab. 6.1 Porovnání použití jednotlivých programovacích jazyků

Jazyk	Počet řádků	Poměr v procentech [%]
C++	6000	55,6
SQL	1600	14,8
PHP	1600	14,8
JavaScript	800	7,4
Bash	500	4,6
HTML a CSS	300	2,8

6.1 Obslužná aplikace modulu pro správu dat

Tato aplikace obsahuje přibližně 6000 řádků kódu v jazyce C++ a 500 řádků SQL dotazů. V aplikaci jsou implementovány:

- Vybrané operace analýzy a zpracování dat:
 - Třídění nových naměřených a zpracovaných dat do patřičných tabulek a jejich označování pro další zpracování.
 - Výpočet agregovaných dat za období 15 sekund, 5 minut, 1 hodina a 1 den. Jsou počítány vždy průměrné, minimální a maximální hodnoty za daná období.
 - Výpočet energie z výkonů naměřených třífázovým vzorkovacím wattmetrem. Počítají se celkové pozitivní, negativní a absolutní činné energie ve všech fázích (L1, L2, L3) i pro všechny fáze dohromady.
- Vybrané operace pro správu a údržbu:
 - Dynamické generování tabulek pro naměřená a zpracovaná data – Automatické vytváření těchto tabulek „na míru“ pro měřicí přístroj a

registrovou mapu, ze kterých data dorazila. Pokud jsou z měřidla očekávána data s menší periodou než 1 minuta, pak dojde k rozdělení tabulky do oddílů. Všechny informace jsou zjištěny z první SOAP zprávy s naměřenými daty, která z měřicího přístroje s danou registrovou mapou dorazila.

- Správa oddílů tabulek – Týká se pouze tabulek rozdělených do oddílů. Vytváření nových oddílů podle potřeby a mazání příliš starých oddílů, jejichž data již nejsou potřeba.
- Údržba SOAP logů - Pravidelné mazání starých informací o příchozích SOAP zprávách, které se týkají naměřených dat. Tyto informace jsou ukládány jen v ladícím režimu aplikace.
- Pravidelný úklid databáze a aktualizování statistik pro plánovač (Příkaz VACUUM ANAZYLE) na vybrané tabulky nebo na vybrané oddíly rozdělených tabulek. Perioda volání je nastavena na jednotlivé tabulky (nebo jejich oddíly) zvlášť podle toho, jak často a v jakém množství je očekávána aktualizace a mazání záznamů v jednotlivých tabulkách (nebo oddílech).
- Pomocné třídy v C++ (1168 řádků) pro snadné používání oficiální knihovny „libpq“, která zajišťuje API v jazyku C pro PostgreSQL databázi.
- 3 vlákna – hlavní (řídící) vlákno, vlákno pro periodické vykonávání úloh a vlákno pro příjem dat z MT Adaptéru. Je zajištěno správné ukončení aplikace po přijetí signálu z OS (SIGINT, SIGTERM). Po těchto signálech aplikace nejdříve počká, až každé vlákno dokončí aktuálně prováděnou operaci (například dokončení zápisu všech dat z přijaté SOAP zprávy nebo dokončení výpočtu agregovaných dat) a až potom se ukončí.
- Detailní výpisy o stavu běhu aplikace na obrazovku nebo do souborů. V případě výpisů na obrazovku jsou výpisy barevně odlišeny podle vlákna. V případě výpisu do souboru má každé vlákno svůj soubor.
- Možnost konfigurace aplikace pomocí konfiguračního souboru. Cesta k souboru a jeho název je zadáván přes první argument programu. V konfiguračním souboru lze nastavit připojení k databázi, nastavení výpisů o stavu běhu aplikace a nastavení portů, na kterých/na které aplikace poslouchá/odesílá SOAP zprávy.

6.2 Instalační balíček modulu pro správu dat

Pro instalaci modulu pro správu dat byly napsány instalační skripty v jazyce Bash. Instalační skripty jsou spolu s binárním souborem obslužné aplikace zabalené do debianovského instalačního balíčku, ze kterého je pak možné modul pro správu dat nainstalovat na server a propojit se stávajícím systémem pro monitorování spotřeb

energií. Samozřejmostí je i možnost modul odinstalovat, odinstalovat pouze obslužnou aplikaci bez dat (bez databáze) nebo nainstalovat pouze aplikaci, kterou napojit na již existující databázi.

Instalační balíček vytváří i databázové uživatele s různými oprávněními přístupu k nové databázi (rozdíl mezi modulem pro správu dat a nadřazeným systémem). Nová databáze je během instalace vytvářena z SQL dotazů, které byly vygenerovány z navrženého datového modelu v aplikaci pgModeler. Po vytvoření nové databáze jsou ihned vloženy první záznamy, které pro obslužnou aplikaci definují typy operací s daty, typy tabulek s daty, jejich vzájemnou kompatibilitu a různé parametry.

6.3 Prezentace velkého objemu naměřených dat a dlouhých časových úseků

Bylo naprogramováno prezentování naměřených dat tak, aby bylo zobrazení hodnot přehledné i pro nejnáročnější případ, kterým je potřeba zobrazovat přibližně 15 miliónů naměřených dat za 6 měsíců. Toto zobrazení je realizováno skrze 5 interaktivních a různě oddálených grafů, z nichž 4 zobrazují data agregovaná za období podle úrovně oddálení a pátý graf zobrazuje původní (originální, syrová) data. V případě dat měřených s větší periodou je zobrazeno méně grafů (úrovní oddálení). Všechna agregovaná data jsou průběžně předzpracována modulem pro správu dat. Tato část, která obsahuje rozložení grafů a ovládacích prvků, vyčítání dat z databáze a jejich zobrazení, byla realizována přibližně 720řádky v jazyce JavaScript, 360řádky v PHP, 30řádky SQL dotazů, a 300řádky v HTML společně s CSS.

Na Obr. 6.1, Obr. 6.2, Obr. 6.3, Obr. 6.4, Obr. 6.5 a Obr. 6.6 je ukázka prezentace činného výkonu ve fázi L1, který byl naměřený třífázovým vzorkovacím wattmetrem. Perioda naměřených dat je 1 sekunda a data jsou k dispozici za 6 měsíců.

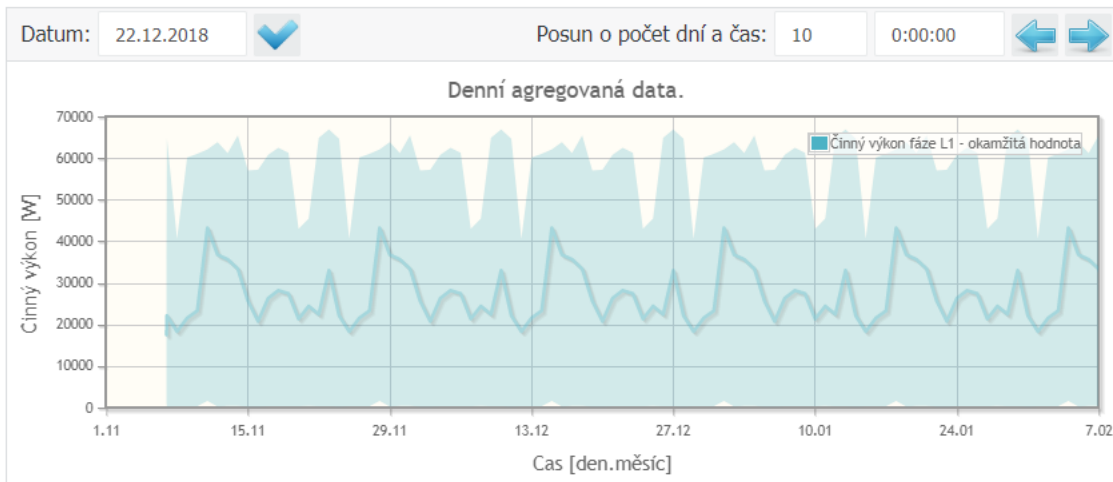
Další funkcionality zobrazení v grafech:

- Pro všechny veličiny je možné zobrazit i průběhy ze všech fází současně.
- U činného výkonu je možné zobrazit průběh součtu výkonů všech fází.
- Posun v grafech je možné přesným zadáním data a času, posunem o zadaný počet dní a čas nebo kliknutím na nadřazený oddálený graf, kdy se aktualizuje pozice grafu podřazeného.
- Je využíváno funkcionality „zoom“ pluginu jqPlot. Každý graf je možné nezávisle na ostatních grafech jakkoliv přiblížit beze změny agregace. To je vidět na Obr. 6.5 a Obr. 6.6, kde graf na Obr. 6.6 je přiblížený graf z Obr. 6.5.

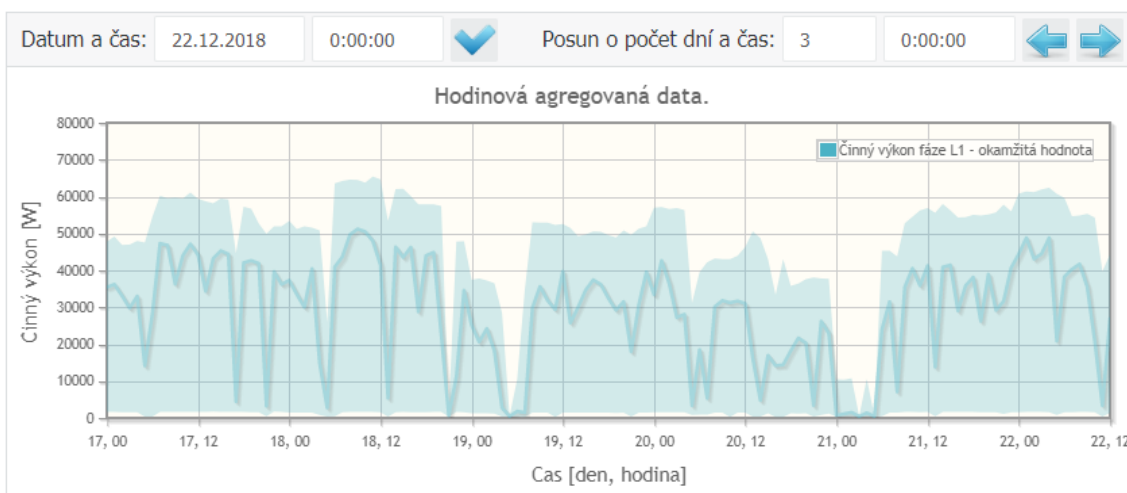
Měřicí přístroj: mt_simulator:mtx:em:001

Informace Data v tabulce Data v grafu Nápověda - jak pracovat s grafy

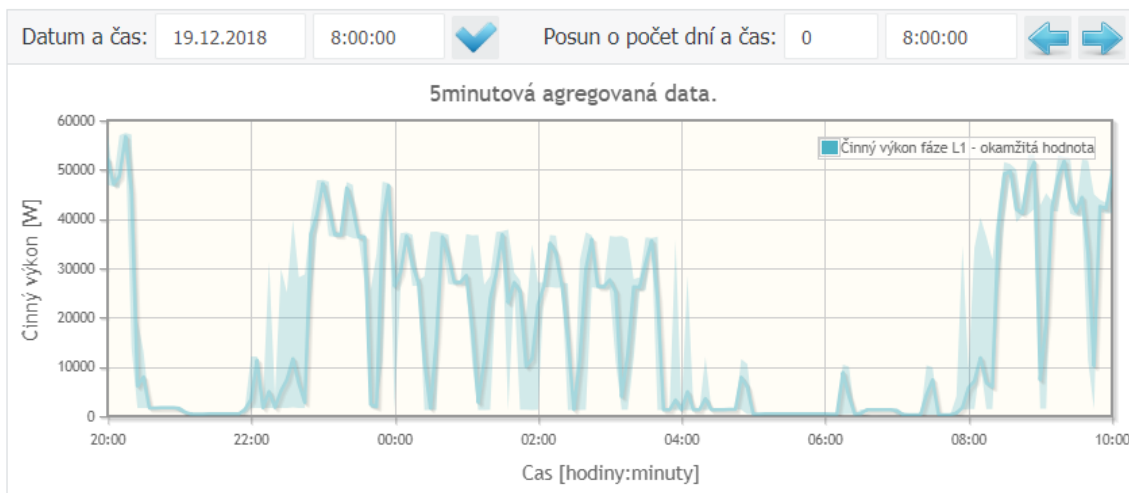
Veličina: Činný výkon [W] První fáze Druhá fáze Třetí fáze Suma



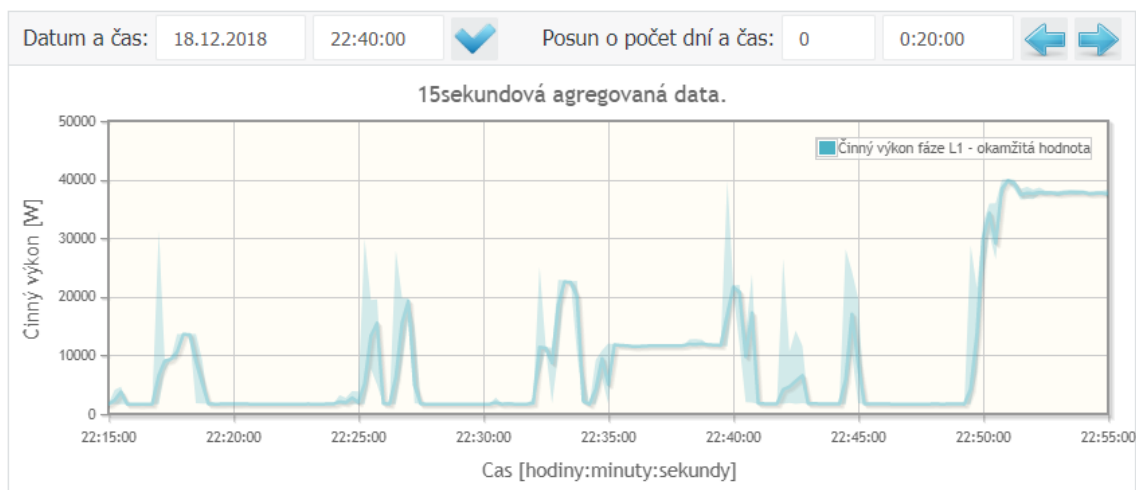
Obr. 6.1 Zobrazení průběhu zvolené veličiny pomocí denních agregovaných dat (4. úroveň oddálení)



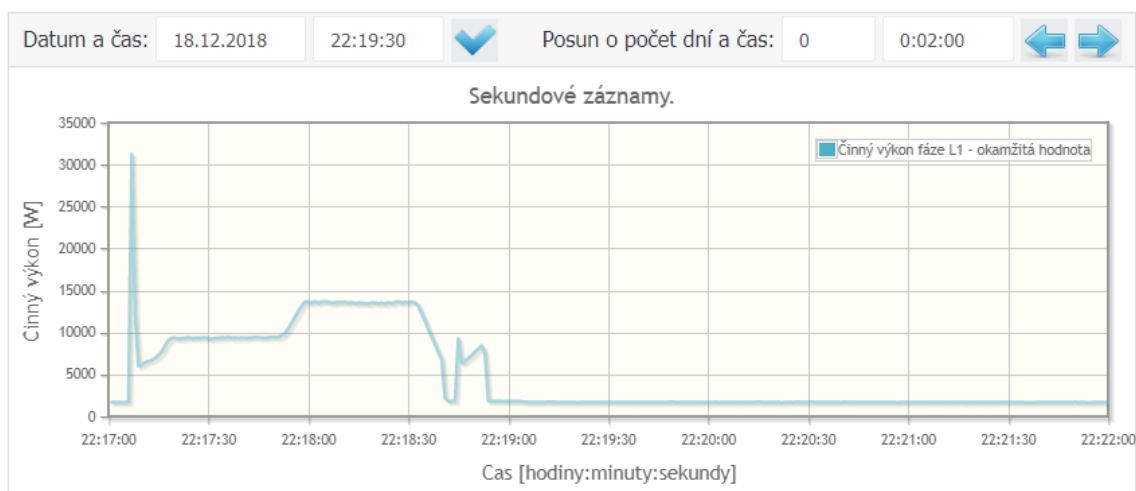
Obr. 6.2 Zobrazení průběhu zvolené veličiny pomocí hodinových agregovaných dat (3. úroveň oddálení)



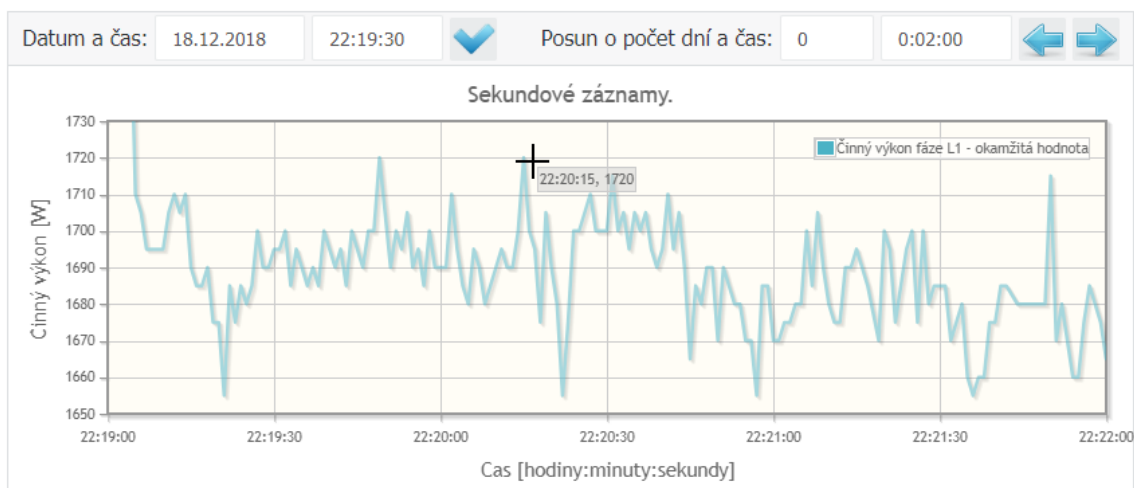
Obr. 6.3 Zobrazení průběhu zvolené veličiny pomocí 5minutových agregovaných dat (2. úroveň oddálení)



Obr. 6.4 Zobrazení průběhu zvolené veličiny pomocí dat agregovaných za 15 sekund (1. úroveň oddálení)



Obr. 6.5 Zobrazení původních (syrových) dat (bez oddálení)



Obr. 6.6 Přiblížení průběhu zobrazeném na grafu z Obr. 6.5

6.4 Vylepšení stávajících možností zobrazení dat

Na Obr. 6.7 je ukázáno nové zobrazování naměřených a zpracovaných hodnot v tabulkách, které bylo naprogramováno podle návrhu z Obr. 5.2.

Měřicí přístroj: mt_simulator:mtx:em:001

Informace Data v tabulce Data v grafu Nápověda - jak pracovat s grafy

Registrová mapa: f20da149d17e7294b552d113cac6c7a6 Datum: 09.04.2019

Unix Timestamp	Čas	Napětí fáze L1 - okamžitá hodnota [V]	Napětí fáze L2 - okamžitá hodnota [V]	Napětí fáze L3 - okamžitá hodnota [V]	Proud fáze L1 - okamžitá hodnota [A]	Proud fáze L2 - okamžitá hodnota [A]	Proud fáze L3 - okamžitá hodnota [A]
1554847199	23:59:59 09.04.2019	230.7	231	231.2	287.7	283.65	292.45
1554847198	23:59:58 09.04.2019	230.7	231	231.2	287.7	283.65	292.45
1554847197	23:59:57 09.04.2019	230.8	231.1	231.3	287.7	283.9	292.4
1554847196	23:59:56 09.04.2019	230.9	231.3	231.6	262.2	256.55	267.75
1554847195	23:59:55 09.04.2019	231	231.2	231.5	261.95	256.85	267.6

Zobrazuji 1 až 10 z celkem 83 045 záznamů Zobraz záznamů 5

První Předchozí 1 2 3 4 5 ... 8305 Další Poslední

Obr. 6.7 Ukázka nového zobrazení naměřených hodnot v tabulkách.

6.5 Optimalizace zobrazování spotřeb

Bylo optimalizováno zobrazování denních, týdenních, měsíčních a ročních spotřeb v přehledech, tabulkách, grafech i při generování spotřeb do souborů.

Jedná se o část systému, kterou bylo možné výrazně urychlit i bez použití nového modulu pro správu dat. Hlavním důvodem pro optimalizaci této části přímo v systému, nikoliv použitím nového modulu pro správu dat, byl časový tlak, kdy tato část měla být optimalizována co nejdříve a to ještě před implementací modulu pro správu dat. I s těmito spotřebami však bylo při návrhu modulu pro správu dat počítáno a bude tedy jednoduché je do modulu v budoucnu přidat.

V rámci této optimalizace byly přímo ve stávajícím systému naprogramovány nové SQL dotazy, kterými byly nahrazeny původní SQL dotazy. Současně byly naprogramovány nové metody, kterými jsou tato data upravována ještě před samotným zobrazením. Výsledkem je několikanásobně rychlejší vyčítání denních, týdenních, měsíčních a ročních spotřeb ze stávající databáze systému. Tyto změny byly v rozsahu zaokrouhleně 1200 řádků kódu v PHP (s malou částí v jazyce JavaScript) a 1100 řádků SQL dotazů.

7 OTESTOVÁNÍ NAIMPLEMENTOVANÉHO ŘEŠENÍ

Testování naimplementovaného řešení bylo provedeno ve třech fázích:

1. Testování modulu pro správu dat na simulovaných datech.
2. Testování prezentace dat na simulovaných datech.
3. Zátěžové testy a otestování celého řešení i na reálných datech.

7.1 Testování modulu pro správu dat na simulovaných datech

Pro první fázi testování byly vytvořeny testovací skripty v jazyce Bash, které:

- Umožňují sestavit SOAP zprávy s náhodně generovanými daty pro 4 měřicí přístroje a 3 různé registrové mapy. Vždy je nastavován i čas dat, který je o něco zpožděný za aktuálním časem. Skript odešle tyto SOAP zprávy na port, na kterém poslouchá modul pro správu dat. Pomocí démona Cron je možné nastavit pravidelné volání skriptů, čímž se tyto SOAP zprávy budou generovat a odesílat na daný port pravidelně. Tímto způsobem byl testován a laděn příjem SOAP zpráv a veškeré navazující operace od analýzy dat až po jejich označení pro vybraná zpracování.
- Umožňují zkopírovat reálná data z „.csv“ souborů do databáze k zadanému měřicímu přístroji. Pro testování byly k dispozici záznamy naměřené za 10 dní 3fázovým vzorkovacím wattmetrem s periodou vzorkování 1 sekunda (864 000 záznamů). Skript zvládne tato data rozkopírovat za sebe tak, aby vznikla data za 6 měsíců (přes 15 miliónů záznamů). Na těchto datech bylo pak testováno a laděno především vlákno pro periodické vykonávání úloh (různé operace pro zpracování dat, správa tabulek a údržba databáze).

7.2 Testování prezentace dat na simulovaných datech

Na datech, která byla do databáze vložena během první fáze testování, byly otestovány a vyladěny nové možnosti prezentace naměřených a zpracovaných dat. Především byl kladen důraz na:

- Rychlost zobrazování průběhů v grafech.
- Kontrola, zda je zobrazení přehledné a má vypovídající hodnotu. Tedy zda byl správně nadefinován maximální počet zobrazovaných bodů na graf, celkový počet oddálení, délka zobrazovaných období a délka intervalů, za jaká jsou data agregovaná.

7.3 Zátěžové testy a otestování celého řešení i na reálných datech

V závěrečné fázi, kdy už bylo řešení odladěné (opraveny chyby, vyřešeny nedostatky, upraveny parametry), bylo celé řešení otestováno pro případ, kdy bylo do chytré sítě zapojeno 5 třífázových vzorkovacích wattmetrů s periodou vzorkování 1 sekunda. Z každého měřidla pravidelně přicházeli do modulu pro správu dat SOAP zprávy se záznamy naměřenými v každé sekundě. Každý ze záznamů obsahuje 15 hodnot (3 napětí, 3 proudy, 3 činné, 3 zdánlivé a 3 jalové výkony). Současně bylo do databáze nakopírováno 105 miliónů simulovaných záznamů s naměřenými daty, které byly rozděleny k 7 měřicím přístrojům.

Tato simulovaná i nově příchozí data měl modul za úkol průběžně zpracovávat. Zpracování se týkalo výpočtu minimálních, maximálních a průměrných hodnot za různá období podle nadefinovaných 4 úrovní oddálení a výpočtu činných pozitivních a negativních energií pro všechny fáze.

Modul pro správu dat běžel na virtualizovaném Ubuntu Server 18.04 LTS s 8GB RAM a 4 jádrovým procesorem. Modul byl bez problémů schopen přijímat, zapisovat a zpracovávat příchozí data z měřidel a současně průběžně zpracovávat i výše zmíněné množství uložených simulovaných dat. Během tohoto stavu byla také testována rychlost zobrazování těchto dat v nově implementovaných grafech a tabulkách.

Modul byl schopen všechna simulovaná data zpracovat do jedné hodiny. Dokud bylo zpracováváno obrovské množství simulovaných dat, byl server naplno vytížen a doba zobrazování průběhů v grafech se pohybovala nejčastěji od 400ms až po 2 sekundy. Jakmile byla všechna simulovaná data zpracována a modul se tedy staral pouze o příjem a zpracování nově příchozích dat, vytížení serveru bylo nepatrné a zobrazování průběhů v grafech se pohybovalo nejčastěji od 30ms do 400ms. Vykonání příkazu VACUUM ANALYZE na celou databázi po těchto zátěžových testech trvalo přibližně 1 hodinu. Tento příkaz aplikace běžně vykonává v nočních hodinách.

8 ZÁVĚR - ZHODNOCENÍ DOSAŽENÝCH VÝSLEDKŮ A DISKUSE DALŠÍHO VÝVOJE SYSTÉMU

V práci byly analyzovány nové požadavky na aktuální systém společnosti ModemTec pro sledování spotřeb energií. Jelikož se jednalo o požadavky na systém, který je již plně v provozu, byl zhodnocen stav systému vzhledem k těmto požadavkům včetně vymezení problémů, které mohou za pomalý běh některých částí systému.

Z analýzy nových požadavků vyplynuly následující skutečnosti:

- Všechny nové požadavky na systém pro monitorování spotřeb energií se týkají komplexního problému zpracování a prezentace dat a vyhodnocování různých stavů a událostí z těchto dat.
- Aktuální možnosti systému v této oblasti musejí být v rámci optimalizací nahrazeny.

Vzhledem k těmto skutečnostem byl zvolen modulární přístup k řešení, kdy byl zcela od začátku navržen a implementován samostatný modul, jehož úkolem je řešit veškerou analýzu a zpracování dat za nadřazený systém pro monitorování spotřeb energií. Modul je pojmenován „Modul pro správu dat“ a běží optimálně i pro měřidla s periodou vzorkování 1 sekunda. Podle zadaných požadavků mají být sekundové záznamy k dispozici minimálně 6 měsíců. To je přibližně 15 milionů záznamů naměřených hodnot jedné veličiny z jednoho měřicího přístroje. I takové množství dat je modul schopen velmi rychle zpracovat. Modul tato data současně vhodně předpřipravuje pro prezentaci, aby je mohl nadřazený systém co nejjednodušeji a nejrychleji zobrazit.

Přehledná prezentace těchto dat v grafech byla také navržena a implementována v rámci této práce. Grafy fungují podobně jako zobrazení na mapách. Je k dispozici čas jako souřadnice a oddálení grafu určuje velikost zobrazovaného období a jaká agregovaná data se uživateli vykreslí (čím větší oddálení, tím jsou data agregovaná po větších intervalech; největší přiblížení tedy odpovídá syrovým datům bez agregace). Je zobrazováno více grafů současně. Jednotlivé grafy jsou mezi sebou interaktivní a je možné z jednoho grafu ovládat zobrazované období jiného grafu.

Po implementaci bylo celé řešení řádně otestováno s důrazem na zátěžové testy.

8.1 Zhodnocení dosažených výsledků

Seznam dosažených výsledků:

- Naměřená data uložená v databázi zabírají až 6x méně úložného prostoru.
- Velké zrychlení zobrazování spotřeb a naměřených dat.
- Přehledná prezentace velkého objemu naměřených dat a dlouhých časových úseků.
- Nové zpracování naměřených hodnot:
 - Výpočet agregovaných dat za období 15 sekund, 5 minut, 1 hodina a 1 den. Jsou počítány průměrné, minimální a maximální hodnoty za daná období.
 - Výpočet energie z výkonů naměřených třífázovým vzorkovacím wattmetrem. Počítají se celkové pozitivní, negativní a absolutní činné energie ve všech fázích (L1, L2, L3) i pro všechny fáze dohromady.

8.1.1 Zpracování a zobrazování naměřených hodnot

Stav nového řešení vzhledem ke zpracování a zobrazování naměřených hodnot je podrobněji popsán v kapitole 7.3, kde měl modul pro správu dat zpracovat 105 miliónů záznamů s nasimulovanými daty a současně přijímat a zpracovávat nově příchozí naměřená data z reálných měřidel s periodou vzorkování 1 sekunda. Během této akce měl nadřazený systém uživateli rychle prezentovat data.

Test byl proveden na virtualizovaném Ubuntu Server 18.04 LTS s 8GB RAM a 4 jádrovým procesorem. Modul byl schopen všechna simulovaná data zpracovat do jedné hodiny. Dokud bylo zpracováváno obrovské množství simulovaných dat, byl server naplněn vytížen a doba zobrazování průběhů v grafech se pohybovala nejčastěji od 400ms až po 2 sekundy. Jakmile byla všechna simulovaná data zpracována a modul se tedy staral pouze o příjem a zpracování nově příchozích dat, vytížení serveru bylo nepatrné a zobrazování průběhů v grafech se pohybovalo nejčastěji od 30ms do 400ms.

Na původní verzi systému (staré řešení) nebyl tento test vůbec vyzkoušený, protože zde byly testovány nové možnosti zpracování a prezentace dat, které staré řešení neobsahuje. Druhým důvodem bylo, že staré řešení mělo velké problémy i s jinými testy, které pracovali pouze s jedním miliónem záznamů.

8.1.2 Šetření úložného prostoru

Další výhodou nového řešení, kromě rychlejší práce s naměřenými daty, je úspora místa na disku, které naměřená data v databázi zabírají oproti starému řešení. Porovnání potřebného místa pro uložení naměřených hodnot v databázi shrnuje

Tab. 8.1. V přehledu nejsou započítány vyhledávací stromy, které si databáze tvoří sama. Rozdíl mezi starým a novým řešením by tak byl ještě výraznější.

Tab. 8.1 Porovnání potřebného místa pro uložení dat před a po optimalizaci

Předmět porovnání	Před	Po
1 uložená naměřená hodnota bez záznamu, k jakému náleží. Pro tento příklad je uvažována hodnota typu double (8 B).	51 B	8 B
Záznam naměřených hodnot bez samotných naměřených hodnot. Označuje, že v daném čase přišli hodnoty z daného zařízení a dané registrové mapy.	46 B	8 B
Příklad pro měřidlo s periodou vzorkování 1 sekunda, kdy je za jeden den k dispozici 86400 záznamů s naměřenými daty. Pro tento příklad jeden záznam obsahuje 15 naměřených hodnot (registrová mapa obsahuje 15 registrů) a všechny jsou typu double. (8 B).	70,1 MB	Naměřená data: 11,1 MB Agregovaná data: 2,2 MB Celkem: 13,3 MB

8.1.3 Optimalizace zobrazování spotřeb

Výsledky optimalizací zobrazování spotřeb byly testovány na serveru s parametry odpovídajícím 2 GB RAM a 2 GHz procesoru s 2 jádry. Data zvoleného projektu relevantní pro zhodnocení výsledků byla přibližně 100 odběrných míst, 70 000 událostí odběrných míst a 1 000 000 záznamů dat z přístrojů. Porovnání přibližných rychlostí načítání dat před a po optimalizaci shrnuje tabulka Tab. 8.2.

Tab. 8.2 Porovnání rychlosti načítání různých dat před a po optimalizaci.

Předmět porovnání	Před	Po
Načtení podrobných informací o aktuální spotřebě jednoho odběrného místa - spotřeba za aktuální den, předchozí den, aktuální měsíc, předchozí měsíc, porovnání těchto spotřeb a poslední naměřená hodnota hlavní sledované veličiny odběrného místa.	1 minuta a 15 sekund	100 milisekund
Načtení tabulky nebo grafu denních nebo týdenních spotřeb pro zvolené odběrné místo.	1 minuta	1 sekunda
Načtení tabulky nebo grafu měsíčních nebo ročních spotřeb pro zvolené odběrné místo.	10 sekund	500 milisekund
Načtení tabulky s přehledem spotřeb za zvolený měsíc nebo rok pro všechna odběrná místa daného projektu.	2 minuty	1 sekunda

8.2 Diskuse dalšího možného vývoje systému

8.2.1 Zpracování dat

Bylo by užitečné rozšířit systém tak, aby si sám uživatel mohl aktivovat/deaktivovat nebo různě nastavit některé operace pro zpracování dat. Nastavení by prováděl přímo z prezentační vrstvy a modul pro správu dat by se v pozadí postaral o zbytek. Návrh některých možností nastavení:

- Změna období, za která jsou data agregovaná.
- Možnost volby veličin, které mají být v systému počítány z měřených veličin.
- Vlastní nastavení automatického odstraňování dat. Úspora místa pokud se uživatel například rozhodne, že bude potřebovat pouze data agregovaná za některé intervaly, nebo že bude potřebovat detailní naměřené údaje například pouze za poslední měsíc.

8.2.2 Zobrazení naměřených a zpracovaných dat

Návrhy na rozšíření nových možností prezentace naměřených a zpracovaných dat:

- Porovnání průběhu se sebou samým, ale s časovým posunem.
- Zobrazení průběhů více veličin v jednom grafu (nutnost přidání dalších os).
- Porovnání průběhů zvolené veličiny mezi více měřidly.
- Možnosti pro změnu vzhledu grafu.

8.2.3 Spotřeby

U spotřeb (denní, týdenní, měsíční a roční) bylo dostatečné provést pouze optimalizaci jejich zobrazování přímo ve stávajícím systému (bez použití modulu pro správu dat). K dokonalosti řešení by však bylo vhodné přesunout i výpočty spotřeb do modulu pro správu dat. Datový model modulu pro správu dat je pro toto rozšíření již připravený.

8.2.4 Vyhodnocení událostí z dat

Někdy je potřeba měřenou veličinu detailně sledovat a na základě jejího průběhu vyhodnotit nějakou událost. Například detekce hodnoty mimo zvolené meze, detekce stočení měřidla, detekce úniku vody, atd. Systém by měl být rozšířen tak, aby takovou událost mohl detekovat, uložit a informovat o ní uživatele. Uživatel by měl být schopen si v systému nadefinovat co u jakého měřidla nebo odběrného místa se má sledovat a jakou formou by měl být informován o nastalé události. Datový model modulu pro správu dat je pro tuto funkčnost již připraven. Je potřeba implementovat operace pro vyhodnocení těchto událostí a navrhnout a

implementovat uživatelské rozhraní v prezentační vrstvě systému pro monitorování spotřeb energií.

Literatura

- [1] *Jak psát web* [online]. [cit. 2019-01-04]. Dostupné z: <https://www.jakpsatweb.cz/publikovani.html>
- [2] NIXON, Robin. *Learning PHP, MySQL & JavaScript*. 5. Sebastopol, United States: O'Reilly Media, Inc, USA, 2018. ISBN 9781491978917.
- [3] JavaScript Definition. *The Tech Terms Computer Dictionary* [online]. 2014 [cit. 2019-01-04]. Dostupné z: <https://techterms.com/definition/javascript>
- [4] ČÁPKA, David. Lekce 1 - Úvod do JavaScriptu. *Itnetwork.cz* [online]. [cit. 2019-01-04]. Dostupné z: <https://www.itnetwork.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>
- [5] JOSEPH, Nelson. An Introduction to Node.js, the Server Side JavaScript. *Open Source For You* [online]. 12.1.2016 [cit. 2019-01-04]. Dostupné z: <https://opensourceforu.com/2016/01/an-introduction-to-node-js-the-server-side-javascript/>
- [6] *PostgreSQL: The World's Most Advanced Open Source Relational Database* [online]. The PostgreSQL Global Development Group, 1996 [cit. 2019-01-04]. Dostupné z: <https://www.postgresql.org/docs/11/>
- [7] CronHowto. *Official Ubuntu Documentation* [online]. [cit. 2019-01-04]. Dostupné z: <https://help.ubuntu.com/community/CronHowto>
- [8] *Official Ubuntu Documentation* [online]. [cit. 2019-01-04]. Dostupné z: <https://help.ubuntu.com/>
- [9] What is Apache?. *WPBeginner* [online]. [cit. 2019-01-04]. Dostupné z: <https://www.wpbeginner.com/glossary/apache/>
- [10] Information about the Apache Web Server. *Web Hosting Services, VPS Servers and Domain Names by NTC Hosting* [online]. [cit. 2019-01-04]. Dostupné z: <https://www.ntchosting.com/encyclopedia/hosting/apache-web-server/>
- [11] Commonly Used Apache Modules: Interserver Tips. *InterServer: Affordable Unlimited Web Hosting, Cloud VPS and Dedicated Servers* [online]. [cit. 2019-01-04]. Dostupné z: <https://www.interserver.net/tips/kb/commonly-used-apache-modules/>

- [12] PITNER, T. Web Services: FI:PA165 - Vývoj programových systémů v jazyce Java. *MUNI IS* [online]. 26.9.2005 [cit. 2019-01-04]. Dostupné z: <https://is.muni.cz/el/1433/podzim2005/PA165/um/slides/03/foilgroup9.html>
- [13] SVOBODA, Vojta. Co je to Composer: Vojta Svoboda Blog. *Vojta Svoboda Blog* [online]. 2014 [cit. 2019-01-04]. Dostupné z: <http://blog.vojtasvoboda.cz/co-je-to-composer>
- [14] *PHP: Hypertext Preprocessor* [online]. The PHP Group, 2001 [cit. 2019-01-04]. Dostupné z: <http://php.net/manual/en/introduction.php>
- [15] *Nette Framework* [online]. Nette Foundation, 2008 [cit. 2019-01-04]. Dostupné z: <https://nette.org/cs/>
- [16] SYMUR: ModemTec. *ModemTec* [online]. ModemTec, s.r.o, 2006 [cit. 2019-01-04]. Dostupné z: <http://www.modemtec.cz/cz/produkty/software/symur#pro-koho-je-p%C5%99%C3%ADnosem>
- [17] *SmartGrid.gov* [online]. U.S. Department of Energy [cit. 2019-01-04]. Dostupné z: <https://www.smartgrid.gov/>
- [18] Chytré sítě. *SŠ-COPT Kroměříž* [online]. Střední škola - Centrum odborné přípravy technické Kroměříž [cit. 2019-01-04]. Dostupné z: <https://coptkm.cz/portal/reposit.php?action=0&id=4571&revision=-1&instance=1>
- [19] Advantages of Smart Grids. *CIRCUTOR: Energy efficiency technology* [online]. [cit. 2019-01-04]. Dostupné z: <http://circutor.com/en/documentation/articles/4162-advantages-of-smart-grids>
- [20] FRANEK, Lešek. *Data koncentrátor pro chytré sítě*. Brno, 2012. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Ing. Pavel Kučera, Ph.D.
- [21] [Http://www.tepco.co.jp/en/corpinfo/procure/invited/pdf/c1_basic.pdf](http://www.tepco.co.jp/en/corpinfo/procure/invited/pdf/c1_basic.pdf). *TEPCO: Tokyo Electric Power Company* [online]. Tokyo: Tokyo Electric Power Company Holdings, 21.3.2012 [cit. 2019-01-04]. Dostupné z: http://www.tepco.co.jp/en/corpinfo/procure/invited/pdf/c1_basic.pdf
- [22] O’Rielly, Kristine & Jeswiet, Jack. (2015). The Need for Better Energy Monitoring within Industry. *Procedia CIRP*. 29. 74-79. 10.1016/j.procir.2015.02.176. Dostupné z:

https://www.researchgate.net/publication/277938607_The_Need_for_Better_Energy_Monitoring_within_Industry

Seznam příloh

Příloha 1 - Vývojové diagramy modulu pro správu dat jsou uloženy na přiloženém CD