

**Vývoj srážko-odtokového modelu
Math1D pro spouštění na HPC**

**Reimplementation of the Math1D
Rainfall-Runoff Model for HPC
Execution**

Zadání diplomové práce

Student: **Radim Vavřík**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Vývoj srážko-odtokového modelu Math1D pro spuštění na HPC
Reimplementation of the Math1D Rainfall-Runoff Model for HPC
Execution**

Zásady pro vypracování:

Cílem této práce je prozkoumat aktuální implementaci srážko-odtokového modelu Math1D a vyvinout jej v prostředí vhodném pro spuštění na HPC. Součástí práce také bude úprava kódu pro paralelní spuštění a následná analýza možností zvýšení míry paralelizace společně s návrhem rozhraní pro změnu parametrů modelu pro provádění what-if analýz, modelování neurčitostí a inverzního modelování.

Konkrétní body zadání jsou:

1. Prozkoumat problematiku implementace srážko-odtokových modelů a aktuální implementaci modelu Math1D v prostředí Matlab.
2. Provést návrh a implementaci modelu Math1D v prostředí vhodnějším pro spuštění na HPC (Fortran/C/C++).
3. Připravit vyvinutý kód pro paralelní spuštění a analyzovat možnosti zvýšení jeho míry paralelizace.
4. Navrhnout rozhraní pro změnu parametrů modelu pro zjednodušení provádění inverzního modelování a modelování neurčitostí.
5. Spolupracovat se členy týmu Floreon+, kteří řeší modelování neurčitostí srážko-odtokových modelů a provést s nimi experimenty pro různé způsoby paralelizace (OpenMP, MPI, kombinace OpenMP/MPI).

Seznam doporučené odborné literatury:

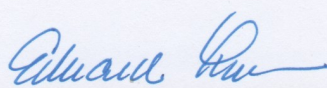
- [1] Beven K.J. 2012. Rainfall-Runoff Modelling: The Primer. John Wiley & Sons. ISBN: 9780470714591
[2] McCool M., Reinders J., Robison A. 2012. Structured Parallel Programming: Patterns for Efficient Computation. Elsevier. ISBN: 9780123914439

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

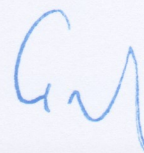
Vedoucí diplomové práce: **Ing. Štěpán Kuchař**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2014

.....
Váňk

Na tomto místě bych rád poděkoval především vedoucímu této diplomové práce Ing. Štěpánu Kuchařovi za neobyčejnou vstřícnost a ochotu při řešení dílčích úkolů a problémů po celou dobu, kdy práce vznikala. Dále také všem, se kterými jsem se mohl podílet na projektu FLOREON⁺, za přínosnou a inspirativní spolupráci.

Abstrakt

Hlavním cílem této práce je popsat novou implementaci srážko-odtokového modelu Math1D, čím se liší od té původní a z jakého důvodu a za jakým účelem byla vytvořena. Prezentovány jsou rovněž technologie použité pro zrychlení běhu modelu a jeho dalších součástí, jakým způsobem byly použity a jaké jejich užití přineslo výsledky. Představeno je také prostředí pro běh, postup spouštění a možnosti využití nově vzniklé aplikace včetně rozhraní pro parametrizaci schematizací modelu Math1D.

Klíčová slova: srážko-odtokový model Math1D, parametrizace, FLOREON⁺, superpočítání, superpočítač Anselm, paralelizace, MPI, OpenMP

Abstract

The main objective of this thesis is to describe a new implementation of the rainfall-runoff model Math1D, how it differs from the original one and why and for what purpose it was created. Presented are also technologies used to increase the operating speed of the model and its other components, how they were used and what results their use has brought. Introduced is also an environment for running, launch procedure and possibilities of using newly developed application including interface for parametrization of Math1D model schematizations.

Keywords: rainfall-runoff model Math1D, parametrization, FLOREON⁺, high-performance computing, supercomputer Anselm, parallelization, MPI, OpenMP

Seznam použitých zkratek a symbolů

| | |
|---------|---|
| ALADIN | – Aire Limitée, Adaption Dynamique, Development International |
| CSV | – Comma-Separated Values |
| CUDA | – Compute Unified Device Architecture |
| ČHMÚ | – Český hydrometeorologický ústav |
| DTO | – Data Transfer Object |
| FLOREON | – Floods Recognition on the Net |
| FD | – Finite Difference method |
| FV | – Finite Volumes method |
| GPGPU | – General-Purpose Computing on Graphics Processing Units |
| HEC-HMS | – Hydrologic Engineering Center - Hydrologic Modeling System |
| HPC | – High-Performance Computing |
| KWA | – Kinematic Wave Approximation |
| MATLAB | – Matrix Laboratory |
| MKL | – Math Kernel Library |
| MPI | – Message Passing Interface |
| NUMA | – Non-Uniform Memory Access |
| OpenCL | – Open Computing Language |
| OpenMP | – Open Multi-Processing |
| SCS-CN | – Soil Conservation Service - Curve Number |
| SV1D | – Saint-Venant 1D |
| XML | – Extensible Markup Language |
| XSD | – XML Schema Definition |

Obsah

| | | |
|----------|---|-----------|
| 1 | Úvod | 5 |
| 2 | Srážko-odtokový model Math1D a jeho součásti | 6 |
| 2.1 | FLOREON ⁺ | 6 |
| 2.2 | Srážko-odtokové modely | 6 |
| 2.3 | Model Math1D | 7 |
| 2.4 | Automatická kalibrace | 14 |
| 2.5 | Modelování neurčitostí | 14 |
| 3 | Konverze modelu Math1D | 15 |
| 3.1 | Výchozí implementace | 15 |
| 3.2 | Běhové prostředí | 16 |
| 3.3 | Vývojové prostředí | 16 |
| 3.4 | Nová implementace | 17 |
| 3.5 | Dokumentace | 25 |
| 4 | Parametrizace modelu Math1D | 26 |
| 5 | Paralelizace modelu Math1D a jeho součástí | 29 |
| 5.1 | OpenMP | 29 |
| 5.2 | MPI | 31 |
| 5.3 | Superpočítač Anselm | 31 |
| 5.4 | Paralelizace modelu Math1D | 32 |
| 5.5 | Paralelizace automatické kalibrace | 34 |
| 5.6 | Paralelizace neurčitostních simulací | 40 |
| 6 | Závěr | 42 |
| 7 | Reference | 43 |
| | Přílohy | 43 |
| A | Výpis souboru | 44 |

Seznam tabulek

| | | |
|---|---|----|
| 1 | Intervaly vybraných srážko-odtokových událostí | 11 |
| 2 | Porovnání klíčových vlastností jazyků MATLAB a C++ | 16 |
| 3 | Výsledné časy výpočtů testu škálovatelnosti modelu Math1D s OpenMP. Pro lepší názornost připojeny i časy sekvenční kalibrace využívající para- lelní model Math1D | 34 |
| 4 | Výsledné časy výpočtů testu škálovatelnosti všech implementací automa- tické kalibrace modelu Math1D pomocí MPI | 40 |

Seznam obrázků

| | | |
|---|--|----|
| 1 | Graf odpovídající kanálům v povodí Ostravice | 8 |
| 2 | Rozložení elementů schematizace | 9 |
| 3 | Třídní diagram zobrazuje strukturu modelu Math1D a její vybrané členy . | 20 |
| 4 | Princip běhu paralelního programu | 30 |
| 5 | Princip přenosu MPI zprávy | 32 |
| 6 | Barevně označeny kritické části kódu | 33 |
| 7 | Povodí se dvěma měřicími stanicemi kalibrované podle 1. implementace kalibrace | 35 |
| 8 | Povodí se dvěma měřicími stanicemi kalibrované podle 2. implementace kalibrace | 37 |
| 9 | Povodí se dvěma měřicími stanicemi kalibrované podle 3. implementace kalibrace | 39 |

Seznam výpisů zdrojového kódu

| | | |
|---|--|----|
| 1 | Ukázka vstupního CSV souboru s naměřenými srážkami. | 10 |
| 2 | Ukázka části kódu modelu Math1D paralelizovaného pomocí OpenMP . . | 30 |
| 3 | Zkrácený obsah XML souboru reprezentující vybranou schematizaci povodí | 44 |

1 Úvod

V posledních letech můžeme pozorovat stále častější výskyt různých povodňových situací, ať už jde o vyhlášení prvního stupně povodňové aktivity, tzv. stavu bdělosti, vyhlášeného již při náznaku nebezpečí povodní, nebo přímo o probíhající záplavy, včetně těch velkých, které pravidelně přináší velké škody na majetku a bohužel i četné ztráty na životech. Tyto události se nevyhýbají ani České republice, potažmo Moravskoslezskému kraji, a proto byl založen projekt FLOREON⁺ (podkapitola 2.1). Jednou z jeho součástí je srážko-odtokový model Math1D, o jehož vývoji pojednává tato práce.

Kapitoly popisují, co si pod pojmem srážko-odtokový model představit, kde se model Math1D vzal (podkapitola 2.3), jaké nad ním existují nástavby a k čemu slouží (podkapitoly 2.4 a 2.5), jakým způsobem byla jeho původní verze implementována (podkapitola 3.1), z jakého důvodu bylo potřeba provést jeho konverzi, jak probíhala a jak dopadla (kapitola 3). Zmíněny budou také nástroje, vybraný programovací jazyk a prostředí, ve kterých vývoj probíhal (podkapitola 3.3).

Aby bylo možné chování modelu nějakým způsobem ovlivňovat, je do něj zakomponována řada parametrů. Jejich nastavování se nazývá parametrizací a může probíhat více způsoby. S cílem zjednodušit a sjednotit přístup k těmto parametrům bylo navrženo rozhraní parametrizace, které je popsáno v kapitole 4.

Vzhledem k záměru provozovat aplikace využívající Math1D model na nově vzniklém superpočítači Anselm (podkapitola 5.3) bylo potřeba zajistit, aby byly jeho možnosti využívány na co nejvyšší úrovni. Za tímto účelem byla provedena paralelizace jak modelu samotného, tak jeho nástaveb. Jednotlivé techniky paralelizace (OpenMP, MPI), způsoby jejich užití a výsledný přínos jsou prezentovány v kapitole 5.

V krátké kapitole ?? bude popsáno, jakým způsobem a s jakými argumenty je možné spustit nově vzniklou aplikaci.

Závěrem je celkově zhodnocen pokrok ve vývoji nové implementace modelu Math1D a jeho přínos oproti implementaci předchozí, popsány jsou ovšem i jeho slabiny a nedostatky současné verze. Ty ale vyústí v inspiraci k alternativním řešením problémů a návrhům na další možnosti vylepšení určeným pro další etapy vývoje.

2 Srážko-odtokový model Math1D a jeho součásti

V této kapitole bude nejprve představen projekt FLOREON⁺, jehož nedílnou součástí je srážko-odtokový model Math1D, zevrubně popsany v následující podkapitole. Jeho vývoj byl hlavním praktickým úkolem této diplomové práce. Aby bylo možné zpřesnit výsledky simulací, spouštěných na modelu, byla vyvinuta automatická kalibrace (podkapitola 2.4). Zmíněno bude i modelování neurčitostí, sloužící ke zvýšení vypovídací hodnoty simulací samotného modelu (podkapitola 2.5).

2.1 FLOREON⁺

FLOREON⁺ je projekt založený za účelem vytvoření platformy, zabývající se modelováním, simulacemi, predikcí a řešením krizových situací (podpora procesů krizového řízení) způsobených např. nepříznivými povětrnostními vlivy, navíc s možností přehledné vizualizace a prezentace výsledků prostřednictvím webového rozhraní. Cílovými uživateli tak mohou být občané, starostové, představitelé obcí a především odborníci. FLOREON⁺ se zabývá několika oblastmi zájmu, stěžejním tématem je ale oblast hydrologie. Systém vyvíjený v rámci projektu je navržen modulárně a počítá s rozšířením o jednotlivé moduly řešící např. dopravu, znečištění vody a ovzduší, průběhy disperze přenosu tepla, šíření lesních požárů apod.

Jak bylo zmíněno v úvodu, ani České republice, potažmo Moravskoslezskému kraji, se nevyhýbají více či méně ničivé povodně na některém z toků. FLOREON⁺ se zaměřuje na čtyři povodí náležící právě území Moravskoslezského kraje, a to povodí Odry, Olše, Ostravice a Opavy.

Simulace jsou v současnosti prováděny automatizovaně každou hodinu, aby byly k dispozici výsledky vždy s aktuálními daty.

V systému FLOREON⁺ je implementováno několik srážko-odtokových modelů. Mezi ně patří HEC-HMS [3, 5], HYDROG a model Math1D, který je vyvíjen na Katedře aplikované matematiky za podpory Moravskoslezského kraje a o kterém pojednává tato práce.

2.2 Srážko-odtokové modely

Obecně modely slouží několika účelům. K získání rámcové představy o sledovaném systému, ke zjednodušení složitých reálných systémů nebo abstrakci podstatných a relevantních částí systému.

V hydrologii jsou modely používány pro zjednodušený popis systémů a procesů z hydrologického hlediska, ale zároveň z toho důvodu, že neexistují metody jak měřit všechny potřebné údaje v plném rozsahu. Proto se přistupuje k modelování, které by mělo chybějící data doplnit nebo nahradit.

Srážko-odtokové modely simulují pomocí různých metod transformaci srážek na odtok vody s ohledem na vlastnosti příslušného povodí. Mohou být děleny podle mnoha kritérií, např. dle:

- způsobu aplikace – návrhové a operativní,

- způsobu schematizace – celistvé, distribuované, semi-distribuované a modely 1D, 2D, 3D,
- časového a prostorového hlediska – kontinuální, diskrétní, regionální a lokální,
- způsobu výpočtu – komplexní, komponentní, 0-dimenzionální, analytické, numerické a hybridní,
- kauzality – stochastické a deterministické,
- typu parametrů – hydrodynamické a hydrologické.

Jednotlivé kategorie modelů se mohou překrývat, jejich vysvětlení nebo způsob kategorizace modelů je ale mimo téma a nad rozsah této práce.

2.3 Model Math1D

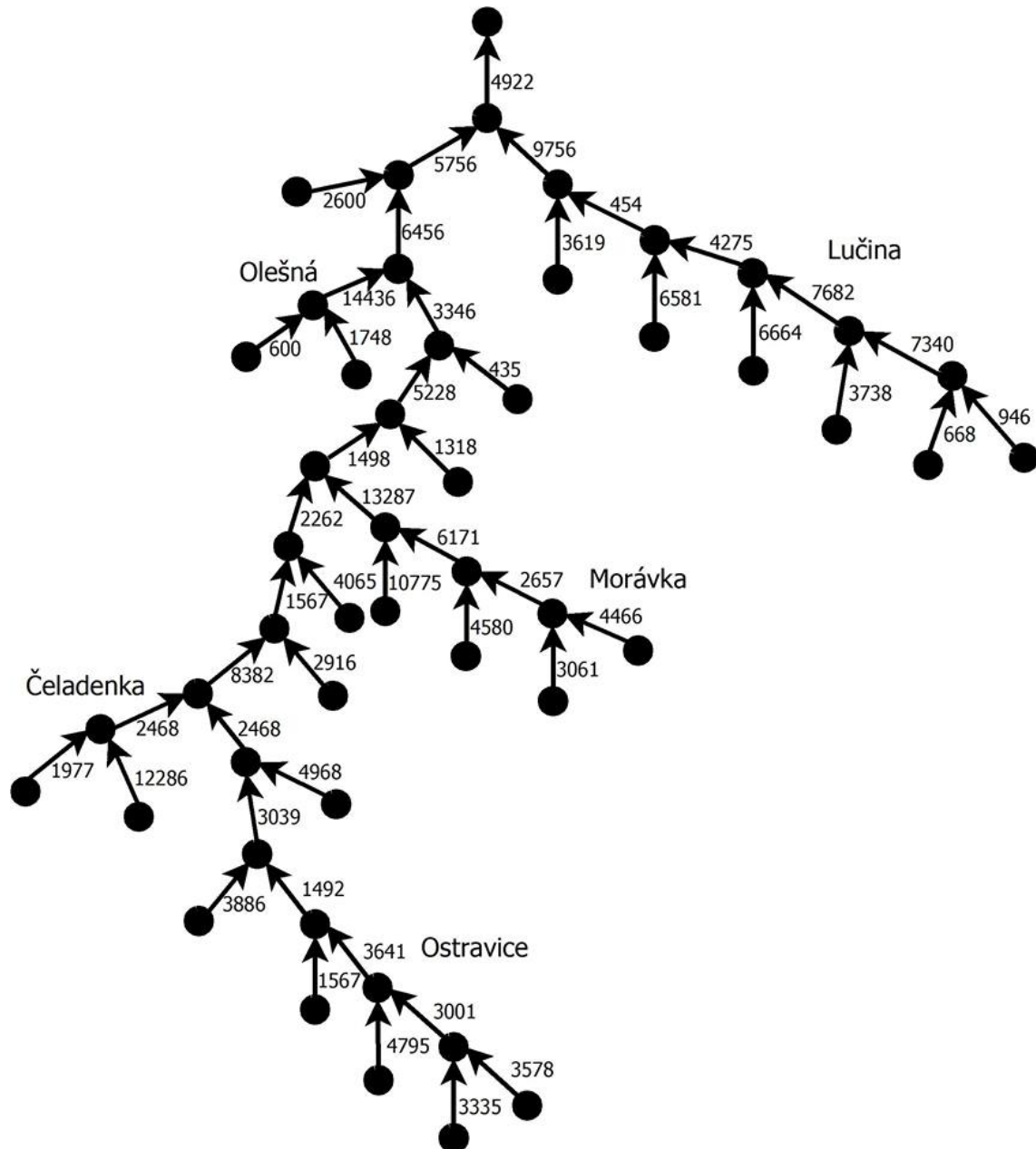
Math1D je matematický srážko-odtokový semi-distribuovaný model. Je samostatnou logickou jednotkou, jejíž rozhraní tvoří sada vstupních a výstupních souborů a spouštěcí metoda. Jeho základem jsou schematizace jednotlivých povodí, nad nimiž jsou s využitím některé z výpočetních metod prováděny simulace. Zjednodušeně řečeno, semi-distribuovaný znamená, že povodí, pro které je simulace spouštěna, může být rozděleno na menší části, tzv. subpovodí. Ta je možno během výpočtů individuálně ovlivňovat např. úpravou určitých parametrů [10].

2.3.1 Schematizace

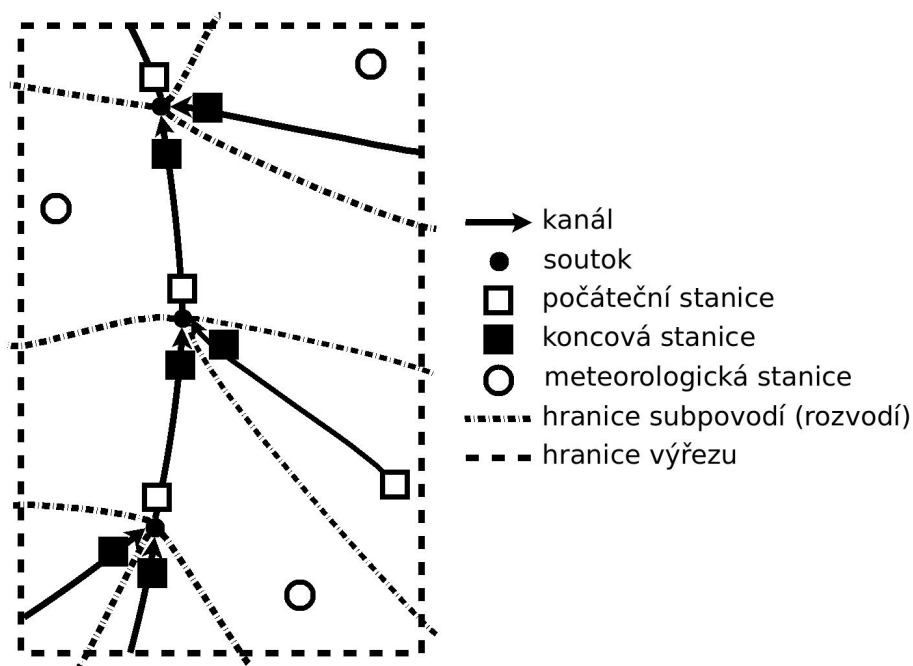
Schematizací je myšlen popis povodí realizovaný komplexní strukturou elementů tohoto povodí a jejich vlastností. Kostrou této struktury je množina kanálů, což jsou jednotlivé úseky vodních toků. Každý z nich je na svém začátku a konci ohraničen hydrologickými stanicemi, ke kterým se během výpočtů vztahují data. Pro tyto stanice platí, že mohou být buď reálné, které ve skutečnosti fyzicky existují a poskytují data o naměřeném průtoku, nebo virtuální, uměle vytvořené, které slouží pro zjednodušení výpočetního modelu. Ke každému kanálu také patří příslušné subpovodí, které reprezentuje přilehlé území, z něhož stékají případné srážky právě do odpovídajícího kanálu. Na územích subpovodí se rovněž nacházejí stanice, ale v tomto případě meteorologické, které dodávají údaje o naměřených srážkách a zároveň se k nim vztahují příslušné hodnoty srážek predikovaných.

Z matematického pohledu po zjednodušení odpovídá kostra schematizace souvislému orientovanému a ohodnocenému grafu, jaký můžete vidět na obrázku 1. Je souvislý, protože všechny kanály povodí nakonec ústí do jednoho uzlu. Orientace vyplývá z přirozeného pohybu vody povodím a ohodnocení hran pak může vyjadřovat délku jednotlivých kanálů.

Ono zmíněné zjednodušení schematizace spočívá v rozdělení souvislých toků na dílčí úseky, a to tak, že kanál vždy začíná soutokem, popř. pramenem a končí následujícím soutokem po proudu toku. Každý kanál je navíc izolován výše zmíněnými hraničními



Obrázek 1: Graf odpovídající kanálům v povodí Ostravice



Obrázek 2: Rozložení elementů schematizace

reálnými, nebo virtuálními měřicími stanicemi. Příklad rozložení jednotlivých elementů schematizace je ilustrován na obrázku 2.

Všechny elementy se skládají z množství atributů a parametrů, které budou specifikovány v podkapitole 2.5.

2.3.2 Vstupní data

Mimo schematizace povodí vstupují do modelu další dva typy vstupních dat. Jedním z nich je množina úhrnů srážek naměřených a predikovaných v daných klimatologických stanicích. Hodnoty množství srážek měřených jsou v současnosti ze zdrojů ČHMÚ a Povodí Odry k dispozici pro každou stanici po 1 hodině. Predikované srážky jsou oproti měřeným dostupné pouze každých 6 hodin s tím, že jsou předpovídány na 54 hodin dopředu. Hodnoty pro tyto 6hodinové intervaly mohou být v průběhu času dále upřesňovány. Jednotkou pro měřené i predikované srážky jsou mm/h .

Druhým typem vstupních dat jsou měřené objemy odtoků. Ty jsou získávány od Povodí Odry každou hodinu a odpovídají průtokům vody v příslušných limnigrafických stanicích. Tyto průtoky jsou získávány z naměřených vodních stavů a měrných křivek průtoků daných hlásných profilů. Jednotkou jsou m^3/s .

Oba vstupní soubory jsou realizovány ve formátu CSV s následujícím obsahem:

- Prvním řádkem je hlavička obsahující jako první položku řetězec `Time`, po němž následuje seznam ID stanic, pro které jsou data určena.

- Všechny další řádky začínají datem a časem ve formátu yyyy-MM-dd HH:mm, pro které jsou data platná. Po něm již řádek pokračuje příslušnými daty (srážky/průtoky) vztaženými k odpovídající stanici z hlavičky. Hodnoty jsou zaznamenány jako desetinná čísla s desetinnou tečkou.
- Všechny sousední prvky na řádku jsou odděleny středníkem.

Krátká ukázka vstupního CSV souboru s hodnotami naměřených srážek je ve výpisu 1.

```
Time;126;130;140;144;145;160;161;169;178;182;188;191;197;224;225;226;227
2007-09-01 05:00;0.8;0;0.1;0;0;0.1;0.2;0.1;0;0;0.2;0;0.1;0
2007-09-01 06:00;0;0;0;0.4;0.4;1.3;0.1;0;0;0.1;0;0.1;0
2007-09-01 07:00;0;0;0.2;0.9;0.3;0.2;0.3;0;0;0.9;0;0.1;0
2007-09-01 08:00;1.1;0;0.6;0.8;0.8;0.2;0.7;0;0;2.4;0;0.1;0
2007-09-01 09:00;0.5;0.7;4.2;1;1.2;0.5;0.8;0;0;1.7;0;0.1;0
2007-09-01 10:00;0.3;0.6;0.3;0.5;0.9;1;0.2;0;0;3.5;0;0.1;0
2007-09-01 11:00;0.4;0.3;4.0;0.4;0.2;5;1.5;0;0;1.9;0;0.1;0
2007-09-01 12:00;0;0;1.1;1;0.1;0.3;0.6;0;0;1.7;0;0.1;0
2007-09-01 13:00;0;0.1;0.3;3;1.4;0.3;0;0.2;0;0.1;0
2007-09-01 14:00;0;0.1;0.4;0.4;0.4;0.3;0;0.2.5;0;0.1;0
2007-09-01 15:00;0.5;0;0;0;1.9;0;0;0;0;0.1;0
2007-09-01 16:00;0;0;0;0;0;2.8;0;0.2;1;0;0.1;0
2007-09-01 17:00;0;0.1;0;0;1.2;0;0;0.7;0;0.1;0
2007-09-01 18:00;0;0.1;0.3;1.5;0.1;0;0;0;0;0.1;0
```

Výpis 1: Ukázka vstupního CSV souboru s naměřenými srážkami.

Aby bylo možné dostatečně otestovat funkčnost a rychlost běhu modelu a jeho dalších součástí, které budou představeny v pozdější části textu, a zkoumat kvalitu jejich výstupů, byly vytvořeny vstupní datové soubory s různou délkou časového intervalu, pro který jsou v souborech obsaženy záznamy. Tyto časové intervaly a data byly navíc vybrány z dostupných záznamů záměrně, vzhledem k rozličným srážko-odtokovým událostem, jako např. povodně, sucho apod. Sady těchto souborů zahrnovaly pro jednotlivá povodí intervaly uvedené v tabulce 1.

Jelikož byly z dostupných zdrojů poskytovány hodnoty jednou za hodinu, byly po identifikaci výše uvedených intervalů a získání příslušných dat z databáze pro kratší časové kroky vytvořeny sady vstupních souborů s interpolovanými hodnotami a časovými kroky záznamů po 15, 10, 5 a 1 minutě. Tato interpolace byla provedena pro všechny výše uvedené časové intervaly a povodí, takže jsou nyní dostupná veškerá možná data pro testování i s velkými vstupními soubory.

2.3.3 Výpočetní metody

Model Math1D pracuje s dvěma typy výpočetních metod. Prvním z nich je výpočet odtoků srážek z předem definované plochy (subpovodí). Pro tuto úlohu je v současnosti v modelu Math1D využívána pouze jedna metoda a tou je SCS-CN.

Metoda SCS-CN (Soil Conservation Service - Curve Number) byla původně vyvinuta v USA pro malá povodí, ukázalo se ale, že je použitelná i v prostředí ČR (pro více informací

| | Sucho | Mírné srážky, snižující se odtoky | Mírné srážky, odtoky ve špičce | Silné srážky, bez povodní | Povodně |
|-----------|--|--|--|--|--|
| Opava | 28. 11. 2011 – 4. 12. 2011 | 5. 7. 2011 – 12. 7. 2011 5. 8. 2011 – 12. 8. 2011 | 5. 8. 2010 – 12. 8. 2010 | 22. 6. 2009 – 25. 6. 2009 12. 10. 2009 – 13. 10. 2009 | 7. 9. 2007 – 13. 5. 2007 17. 5. 2010 – 24. 5. 2010 2. 6. 2010 – 4. 6. 2010 |
| Odra | 30. 5. 2012 – 3. 6. 2012 | 5. 7. 2011 – 12. 7. 2011 5. 8. 2011 – 12. 8. 2011 | 5. 8. 2010 – 12. 8. 2010 | 22. 6. 2009 – 24. 6. 2009 12. 10. 2009 – 13. 10. 2009 | 7. 9. 2007 – 10. 9. 2007 17. 5. 2010 – 19. 5. 2010 |
| Ostravice | 10. 11. 2011 – 30. 11. 2011 13. 12. 2011 – 23. 12. 2011 | 5. 8. 2011 – 12. 8. 2011 | 5. 8. 2010 – 12. 8. 2010 | 12. 10. 2009 – 13. 10. 2009 30. 6. 2011 | 16. 5. 2010 – 19. 5. 2010 |
| Olše | - | 5. 7. 2011 – 12. 7. 2011 | 5. 8. 2010 – 12. 8. 2010 5. 8. 2011 – 12. 8. 2011 | 12. 10. 2009 – 13. 10. 2009 30. 6. 2011 | 7. 9. 2007 – 8. 9. 2007 16. 5. 2010 – 19. 5. 2010 2. 6. 2010 1. 9. 2010 |

Tabulka 1: Intervaly vybraných srážko-odtokových událostí

viz [3]). Je to metoda, která modeluje odtokovou oblast a odhaduje odtok srážek. Modelování probíhá na základě rozlohy oblasti, počátečního průtoku subpovodím, počáteční abstrakce (ztráty) a hodnoty CN křivky. CN křivka je empirický parametr používaný v hydrologii a vyjadřuje vlastnosti půdy vzhledem k odtoku a vsakování srážek. Bere v úvahu např. typ půdy, její využití (les, pole, zástavba atd.) a předchozí vlhkost (míru nasáknutí). Hodnoty CN křivek jsou limitovány fyzikálními zákonitostmi v rozmezí čísel 30 - vysoká míra retence (ztrát) a 100 - nulová míra retence (beze ztrát). V metodě je využíván také tzv. jednotkový hydrogram, což je sada parametrů reprezentujících chování povodí v případě jednotkového množství srážek [2].

Druhou skupinou výpočetních metod jsou metody hydrodynamické. V modelu Math1D bylo pro experimentální účely využíváno několik těchto metod, které modelují proudění vody a pomocí nichž jsou počítány příspěvky samotného toku do výstupního průtoku (v místě spodní hydrologické stanice). Mezi ně patří např. KWA-Comsol – aproximace kinematických vln modelovaná nástrojem COMSOL Multiphysics[®], KWA-ZCU – aproximace kinematických vln Západočeské univerzity v Plzni, SV1D – metoda podle Saint-Venantova principu v 1D, KWA-FV – aproximace kinematických vln založená na metodě konečných objemů, KWA-FD – aproximace kinematických vln založená na metodě konečných diferencí a interně vyvinutá metoda Velocity.

Poslední zmiňovaná je používána v současnosti stabilně. Tato metoda pracuje s jedním ze tří dostupných profilů koryt toků, a to profilem s fixní hloubkou, obdélníkovým profilem, nebo s lichoběžníkovým profilem, který nejvíce odpovídá reálným podmínkám a je pro simulace nejpřesnější. Jednotlivé profily jsou užívány v závislosti na dostupnosti dat popisujících tvar a rozměry koryt. Vstupy této metody jsou tzv. Manningův koeficient kanálu (součinitel drsnosti koryta), sklon kanálu, jeho šířka, sklon břehů a hodnota vstupního průtoku, který odpovídá průtoku na horní hydrologické stanici. Výstupem pak je vypočítaná rychlost proudění vody kanálem, díky níž se následně dopočítá výstupní průtok kanálu.

2.3.4 Výstupní data

Na výstupu modelu Math1D, resp. simulace na něm, mohou být v závislosti na požadavcích 3 datové množiny. Nejdůležitější jsou výstupní nasimulované odtoky jednotlivých kanálů v m^3/s odpovídající jejich spodním hydrologickým stanicím. Hodnoty jsou stejně jako u vstupních dat vztaženy k určitému datu a času v krocích po 1 hodině. Analogicky vypadá výstup u množiny vodních stavů (výšek hladiny) a rychlostí průtoků vody.

Po formální stránce jde o soubory ve formátu CSV s tímto obsahem:

- Prvním řádkem je hlavička obsahující jako první položku řetězec `time/id`, po němž následuje seznam ID stanic, pro které jsou data určena.
- Všechny další řádky začínají datem a časem ve formátu `yyyy-MM-dd HH:mm`, pro které jsou data platná. Po něm již řádek pokračuje příslušnými daty (odtoky/vodními stavy/rychlostmi toku) vztaženými k odpovídající stanici z hlavičky. Hodnoty jsou zaznamenány jako desetinná čísla s desetinnou tečkou.

- Všechny sousední prvky na řádku jsou odděleny středníkem.

V současné době má smysl uvažovat pouze hodnoty simulovaných odtoků vody.

2.3.5 Průběh simulace

Nejpoužívanější časový úsek používaný v rámci systému FLOREON⁺ zahrnuje data pro 7 dní po 1 hodině. Přitom platí, že vstupní hodnoty srážek jsou pro prvních 5 dnů měřené, zatímco zbylé 2 dny patří hodnotám predikovaným. Model je samozřejmě schopný počítat simulace pro libovolně dlouhé časové úseky. Limitován je pouze dostupností vstupních dat.

Celá simulace začíná nastavením vlastností průběhu simulace, mezi které patří např. nastavení typů výpočetních metod, profilu koryta toku, použití/vynechání adjustace (viz níže) a řady dalších popisných parametrů.

Dalším krokem je načtení schematizace povodí, na kterém bude simulace probíhat, a vstupních dat z příslušných souborů. Úhrny srážek a měřené objemy odtoků jsou poskytovány ve formátu CSV. Pomocí těchto měřených objemů odtoků jsou následně vytvořeny hydrogramy pro reálné hydrologické stanice.

Následuje samotný výpočet dílčích příspěvků odtoků povodí. Během něj simulace iteruje skrze všechny kanály schematizace. Pokud se jedná o prameny – kanály, které nemají žádné přítoky, jsou užitím metody SCS-CN vypočítány pouze odtoky ze subpovodí náležící příslušnému kanálu. V případě všech ostatních kanálů jsou nejprve akumulovány příspěvky všech přítoků počítaného kanálu, pak je pomocí zvolené metody vypočítán hydrodynamický příspěvek kanálu samotného a nakonec je opět prostřednictvím metody SCS-CN připočítán příspěvek odtoků z odpovídajícího subpovodí.

V případě, že byla na začátku simulace v nastavení povolena adjustace, je vykonána. Ve své podstatě se jedná o opravu chyby modelu. Aby vypočítané hodnoty odpovídaly reálným průtokům, je použita adjustace vycházející z metody nejmenších čtverců. Při adjustaci se prochází reálné hydrologické stanice, a pokud je právě počítán kanál, jehož subpovodí spadá pod tuto stanici, je výsledný odtok upraven podle předpisu

$$\bar{Q} = \alpha Q + \beta,$$

kde Q je původní odtok a α , β jsou adjustační parametry získané řešením minimalizační úlohy zmíněnou metodou nejmenších čtverců.

Po výpočtu konečných odtoků a případné adjustaci jsou hodnoty zkonvertovány na odpovídající výšky hladiny (vodní stavy) pro dané hydrologické stanice. Přepočtení je provedeno na základě teoretického přepočtu podle Manningovy rovnice [10].

Posledním krokem simulace je sběr vypočtených hodnot a vygenerování výstupních souborů. Stejně jako ty vstupní jsou ve formátu CSV. Jde o výstupní odtoky, vodní stavy, popř. rychlosti průtoků uvedené v předchozí podkapitole 2.3.4.

2.4 Automatická kalibrace

Každý model, resp. simulace, v sobě zahrnuje určitou míru nepřesnosti a z ní pramenící chybu. Ta se u srážko-odtokového modelu Math1D projevuje odchylkou mezi simulovanými a reálně naměřenými daty za stejných podmínek, tzn. ve stejný čas, pro stejné místo toku a se stejnými vstupy (srážky, počáteční průtoky).

Cílem automatické kalibrace je nastavení parametrů schematizace tak, aby byla chyba minimalizována pod určitou předem danou toleranci. V cyklu této nastavby nad samotným modelem jsou tedy po definovaném kroku měněny hodnoty CN křivek a Manningova koeficientu drsnosti. Velikosti změny jsou limitovány pouze dříve zmíněnými fyzikálními limity parametrů (hodnoty 30–100 u CN křivek a 0,005–0,1 u Manningova koeficientu). Pro každou kalibrovanou oblast se nejprve vypočítá počáteční chyba před kalibrací. Následně jsou o daný krok změněny hodnoty parametrů a je vypočítána hodnota chyby s novými parametry. Pokud je hodnota rozdílu dvou bezprostředně po sobě následujících chyb větší než definovaná tolerance, je v cyklu proces opakován. V opačném případě je kalibrace dané oblasti ukončena, zaznamenána velikost chyby pro aktuální hodnoty parametrů a tyto hodnoty jsou označeny jako zkalibrované. V průběhu iterací jsou délky kroků (velikosti změn) parametrů a dílčí chyby vyčíslovány dynamicky pomocí parciálních derivací a vážené Euklidovské normy.

Kvůli nutnosti zjistit nové výstupy simulace modelu vždy po každé změně hodnoty parametru, a to pro hodnoty CN křivek i Manningova koeficientu, je v průběhu kalibrace spouštěna samotná simulace modelu Math1D mnohokrát v závislosti na velikosti počáteční odchylky měřených a simulovaných hodnot a nastavené velikosti kroku změny parametrů. Z tohoto důvodu je automatická kalibrace poměrně výpočetně náročná.

2.5 Modelování neurčitostí

Aby bylo možné zvýšit míru vypovídací hodnoty simulací, byly do využití Math1D modelu zaneseny neurčitosti. Byl vytvořen rámec pro neurčitostní simulace modelu, na jehož vstupu je zadána požadovaná odchylka množství srážek a kvantily hodnot, pro něž jsou neurčitostní simulace spouštěny. Výstupem jsou hydrogramy reprezentující možný rozptyl hodnot průtoků s mírou pravděpodobnosti danou zmíněnými vstupními kvantily. K výpočtu je v současné době využívána stochastická simulační metoda Monte Carlo.

Stejně jako automatická kalibrace je modelování neurčitostí jakousi nastavbou nebo rámcem nad samotným modelem Math1D, který je v průběhu modelování těchto neurčitostí mnohonásobně spouštěn. Proto se rovněž jedná o výpočetně vysoce náročnou úlohu. Řešením modelování neurčitostí s využitím modelu Math1D se podrobně zabývá paralelně vznikající práce [4].

3 Konverze modelu Math1D

Matematický srážkoodtokový model Math1D byl od svého počátku vyvíjen pomocí jazyka MATLAB. Ten je skriptovacím programovacím jazykem 4. generace, zaměřeným zejména na počítání s maticemi (z angl. matrix laboratory). Jde o bezesporu vynikající profesionální nástroj ve své kategorii. Jeho výhodou je možnost abstrakce matematických i jiných problémů na velmi vysoké úrovni.

To s sebou však oproti standardním programovacím jazykům přináší i nevýhodu v podobě nižšího potenciačního výpočetního výkonu a menší možnosti optimalizace a škálovatelnosti pro paralelní spouštění. Hlavní nevýhodou je ale komplikovaná integrace takto vyhotovených programů do složitějších logických celků, jako je tomu právě u modelu Math1D, respektive systému FLOREON⁺. Ta je zapříčiněna nutností vytvářet specifické rozhraní mezi systémem a integrovaným programem v jazyce MATLAB. Toto rozhraní je nezbytné pro vzájemné předávání dat a především samotné spouštění programů v MATLABu. Zanedbatelná není ani nutnost mít pro takový program k dispozici samotné komerční běhové prostředí MATLAB, čímž je možnost nasazení na libovolné stroje a platformy značně omezena.

Za účelem eliminace výše uvedených nedostatků spolu se záměrem rozšířit model o popsanou automatickou kalibraci a modelování neurčitostí vyplynula potřeba konverze aktuální implementace v MATLABu do některého vhodného programovacího jazyka. Na základě svých obecných vlastností, univerzálnosti, rozšířitelnosti, jednoduché integrovatelnosti, možnosti velmi dobře a přehledně reprezentovat schematizaci povodí a potažmo celý model pomocí tříd a také s ohledem na potenciační výkonnost pro plánovanou paralelizaci modelu a jeho součástí byl pro novou implementaci vybrán jazyk C++.

Pro doplnění následuje v tabulce 2 porovnání klíčových vlastností obou jazyků.

3.1 Výchozí implementace

Jak již bylo zmíněno, původní implementace modelu Math1D byla vytvořena na Katedře aplikované matematiky. Z toho také vyplývá použití programového prostředí a programovacího jazyka MATLAB, který je vhodný pro rychlý vývoj a prototypování. Výhodou je také možnost jednoduché a efektivní práce s maticemi, což bylo v původní implementaci modelu hojně využito.

Matlabovský kód obsahuje kromě implementace modelu tak, jak je popsán v podkapitole 2.3, i mnoho dalších experimentálních, testovacích nebo pomocných funkcí, které již ale při běžném spouštění simulací nejsou využívány. Jde např. o obousměrné převody mezi průtoky a výškami hladiny, převody mezi různými datovými typy, třídění kanálů, převody časových formátů, práci s webovými službami nebo vykreslování grafů.

Dalším specifikem implementace modelu v MATLABu je, že nezpracovává a neposkytuje výstupní hodnoty rychlosti pohybu vody v toku.

| Vlastnost | MATLAB | C++ |
|-----------------|----------------------|-------------------|
| Zpracování kódu | interpretovaný | kompilovaný |
| Typová kontrola | dynamicky typovaný | silně typovaný |
| Silná stránka | rychlé prototypování | výkonnost |
| Zaměření | maticové operace | programové smyčky |

Tabulka 2: Porovnání klíčových vlastností jazyků MATLAB a C++

3.2 Běhové prostředí

Od svého začátku byl model Math1D provozován pod operačním systémem Microsoft Windows, ať už šlo o jeho vývoj v MATLABu, nebo jeho nasazení na výpočetní server. Rovněž databáze obsahující data pro schematizace a všechna vstupní data pro model pochází z dílny Microsoftu (MS SQL Server). Z důvodu plánovaného nahrazení nové implementace modelu místo původní a její integrace do systému FLOREON⁺ byla konverze matlabovského kódu započata právě pod systémem Windows. Druhým aspektem konverze byl požadavek na budoucí paralelizaci modelu, automatickou kalibraci i modelování neurčitostí a jejich spouštění na nově vzniklém superpočítači Anselm (detailnější popis v kapitole 5.3). Operačním systémem na Anselmu je naopak Linux, proto bylo již od začátku nutné vyhnout se platformově závislým knihovnám a funkcím jazyka C++.

3.3 Vývojové prostředí

Většina vývoje (konverze) byla realizována v integrovaném vývojovém prostředí Microsoft Visual Studio 2010 Ultimate¹ s interním C++ překladačem. Toto prostředí bylo zvoleno díky komfortnímu procesu vývoje, ladění, debugování, možnosti jednoduché integrované výkonové analýzy kódu nebo základnímu integrovanému klientu pro přístup do databáze MS SQL Server, který pro potřeby konverze plně dostačoval. V pravidelných intervalech byla také testována kompatibilita kódu na Linuxu (Ubuntu 12.04 LTS) pomocí kompilátoru g++ ve verzi 4.8.1 s podporou nového standardu C++11 [9].

Po dokončení konverze byla započata paralelizace kódu modelu, automatické kalibrace i modelování neurčitostí, která už ale probíhala čistě v prostředí Linuxu. Protože již nebylo potřeba masivního vývoje, jednalo se o rozsahově méně náročné úpravy kódu, nebyl kód portován do žádného dalšího integrovaného vývojového prostředí (nabízelo se např. prostředí CodeBlocks²). Pro editaci kódu byl dostatečný pokročilý textový editor Sublime Text³. K sestavení a kompilaci byla využita klasická utilita make v kombinaci se zmíněným kompilátorem g++, později pro experimentální účely i kompilátorem společnosti Intel icpc verze 13.1.3 a nakonec pro kompilaci kódu s využitím MPI, o kterém pojednává samostatná kapitola 5.2, i tzv. wrapper kompilátorů mpicc. Pro potřeby debugování a ladění kódu pak byl použit klasický debugger gdb verze 7.4 a 7.2.

¹<http://www.visualstudio.com/>

²<http://www.codeblocks.org>

³<http://www.sublimetext.com/>

3.4 Nová implementace

Prvním krokem konverze modelu Math1D z původní matlabovské implementace do jazyka C++ bylo prostudování a pochopení modelu a procesu simulace. Samotná konverze pak spočívala ve třech krocích:

1. vytvoření datové struktury reprezentující schematizaci povodí,
2. přenesení jednotlivých výpočetních algoritmů,
3. návrh a implementace vhodného rozhraní.

Samotný jazyk C++ ze své podstaty neposkytuje vysokoúrovňové funkce, operátory a jiné nástroje pro práci s daty, zejména s maticemi. Původní implementace modelu v MATLABu na nich byla naopak založená, proto bylo nutné všechny tyto prvky implementovat pomocí jednodušších stavebních kamenů, které jazyk C++ nabízí. Šlo např. o násobení matic, násobení po prvcích, logické maticové operace, vytváření aritmetických posloupností nebo čtení a zápis matic do souborů. Další z řady odlišností MATLABu je také jeho nedeklarativní práce s proměnnými, kterou bylo ve spojitosti s interpretací kódu oproti deklaracím proměnných a kompilaci C++ nutno zohlednit.

3.4.1 Návrh a realizace schematizace

Pro potřeby perzistence byla navržena a vytvořena datová struktura ve formátu XML souboru a jeho odpovídající XSD schéma. Obsah XML souboru a rozsah dat odpovídá původní struktuře používané v prostředí MATLABu, včetně názvů elementů a atributů. Byly pouze vypuštěny některé nevyužívané atributy daných entit povodí a naopak byly doplněny jednoznačné identifikátory a indexace, na jejíž korektnosti je závislý úspěšný průběh simulace. Zkrácená ukázka obsahu XML souboru reprezentující jednu ze schematizací je v příloze (výpis 3). Obsah byl zkrácen tak, aby každá kolekce obsahovala pouze jeden svůj element.

Pro generování XML souborů se schematizacemi jednotlivých povodí byla vytvořena utilita `Math1DXMLSchematizationWriter`. Na základě zadaného ID povodí se utilita připojí k databázi projektu FLOREON⁺. Pomocí sdílených tříd pro práci s databází jsou stažena data o povodí a inicializovány příslušné datové objekty (DTO). Ty jsou následně již čistě v režii utility zpracovány, setříděny a zaindexovány, načež je vytvořen XML soubor s požadovanou schematizací. Jelikož utilita využívá některé společné části kódu z jiných modulů systému FLOREON⁺, je stejně jako ony naprogramována v jazyce C#.

Reprezentace schematizací pomocí XML souborů byla zvolena pro svou jednoduchost a možnost univerzálního použití i v budoucnu. Schematizace jsou poměrně statické popisné celky, které se téměř nemění, jelikož popisují geomorfologické vlastnosti koryt vodních toků a povodí. Z toho důvodu se předpokládá, že pro simulace budou využívány předgenerované XML soubory jednotlivých povodí. Tímto se ušetří čas při simulacích jinak potřebný pro přístup do databáze a získání dat schematizací. Oproti původní implementaci v MATLABu byla eliminována také všechna komunikace prostřednictvím webových služeb, která je značně časově náročná.

Pro potřeby generování schematizací do XML souborů a pro jejich validaci vzniklo rovněž odpovídající XSD schéma, které definuje povolené elementy, jejich pořadí, datové typy, četnost výskytu nebo rozmezí přípustných hodnot. Utilita `Math1DXMLSchematizationWriter` nabízí možnost po vygenerování XML souboru schematizace provést validaci právě oproti tomuto XSD schématu. To může být užitečné např. po importu nových dat do databáze.

3.4.2 Konfigurace

Jako rozhraní pro nastavení cest ke vstupům a pro výstupy modelu `Math1D` slouží konfigurační soubor. Ten je ve formátu XML a obsahuje následující elementy:

- `matDataXmlFilePath` – obsahuje cestu k XML souboru s vygenerovanou schematizací povodí určeného k simulaci,
- `measuredDischargeVolumesCsvFilePath` – obsahuje cestu k CSV souboru, ve kterém jsou informace o odtocích měřených v daných hydrologických stanicích ve specifikovaný den a čas,
- `precipitationsCsvFilePath` – obsahuje cestu k CSV souboru, ve kterém jsou informace o srážkách naměřených, popř. predikovaných pro dané klimatologické stanice na specifikovaný den a čas,
- `qCsvFilePath` – obsahuje cestu k CSV souboru, do něhož budou zapsány (přepsány) výsledné nasimulované hodnoty průtoků pro dané hydrologické stanice ve specifikovaný den a čas,
- `hCsvFilePath` – obsahuje cestu k CSV souboru, do něhož budou zapsány (přepsány) výsledné nasimulované hodnoty vodních stavů (výšek hladin) pro dané hydrologické stanice ve specifikovaný den a čas,
- `vCsvFilePath` – obsahuje cestu k CSV souboru, do něhož budou zapsány (přepsány) výsledné nasimulované hodnoty rychlostí pohybu vody pro dané hydrologické stanice ve specifikovaný den a čas,
- `logTarget` – nastavení cíle pro logovací výpisy modelu. 0 znamená výpis do standardní konzole, 1 znamená výpis do textového logovacího souboru a 2 znamená, že model nebude produkovat žádný textový výstup o průběhu simulace kromě kritických a chybových hlášení,
- `logFilePath` – obsahuje cestu k textovému souboru, do kterého je v případě nastavení hodnoty 1 v elementu `logTarget` zaznamenáván průběh simulace modelu.

Cesta k samotnému konfiguračnímu souboru je předávána formou argumentu při spuštění aplikace, viz kapitola ??.

3.4.3 Inicializace modelu

Po spuštění aplikace je provedeno parsování argumentů, se kterými byla spuštěna. Na jejich základě se běh rozvětňuje a je spuštěna samostatná simulace srážko-odtokového modelu, automatická kalibrace, nebo modelování neurčitostí. Pokud byla zvolena první možnost, je vytvořena instance modelu `Math1D`.

Ta je v kódu tvořena třídou `MatData`, která je hlavní a výchozí třídou implementace modelu. Obsahuje všechny důležité a pro simulaci nepostradatelné metody a členy. Název `MatData` je převzat stejně jako většina ostatních identifikátorů z původní implementace v `MATLABu` z důvodu zachování přehlednosti a zpětné dohledatelnosti funkcionality. Konstruktor třídy `MatData` přijímá jako parametr cestu ke konfiguračnímu souboru. Ta může být, jak bylo uvedeno výše, zadána jako argument při spuštění, a pokud v argumentu dodána není, je použita výchozí relativní cesta `../Resources/Config.xml`.

V rámci vykonávání těla konstrukturu je parsován konfigurační soubor a inicializován objekt třídy `Logger`, který se v průběhu simulace stará o vypisování informací do požadovaného logovacího kanálu.

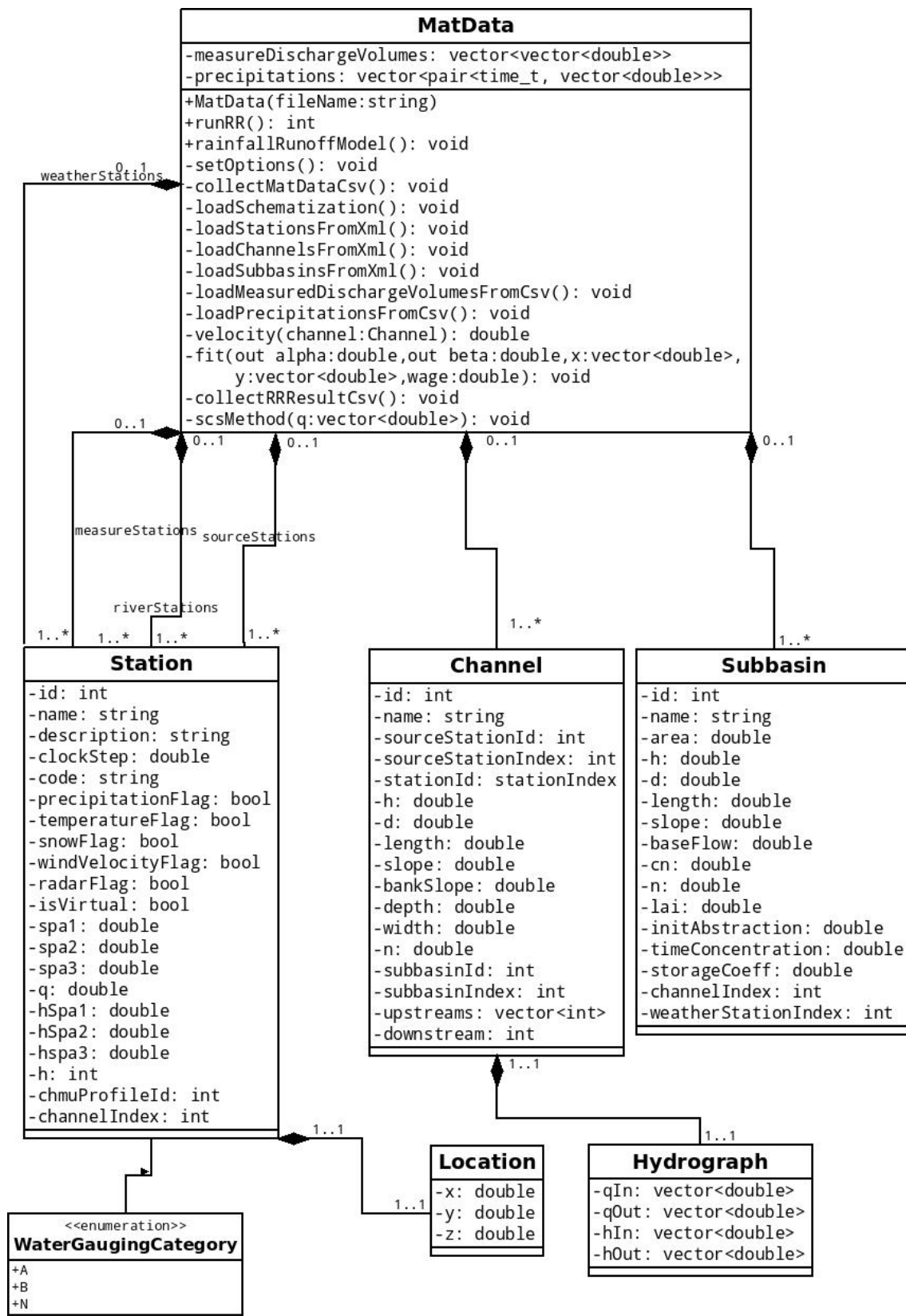
Následuje volání metody `runRR`, která zahrnuje vše, co je potřeba pro úspěšnou simulaci provést. Implementována je jako sekvence několika kroků:

1. Volání metody `setOptions`, která přednastaví sadu atributů modelu jejich výchozími hodnotami, např. minutový krok simulace, data a časy začátku/konce simulace, názvy zvolených výpočetních metod, typ profilu koryt toků nebo zapnutí/vypnutí adjustace.
2. Volání metody `collectMatDataCsv`, která zaobaluje načítání všech vstupních dat modelu z jednotlivých vstupních souborů.
3. Výpis počátečních hodnot vybraných položek nastavení.
4. Volání metody `rainfallRunoffModel` – tato metoda spustí samotný výpočet srážko-odtokové simulace nad danými vstupními daty.
5. Volání metody `collectRRResultCsv` – metoda z modelu vyextrahuje vypočtená simulovaná data pro výstup a vygeneruje z nich požadované výstupní CSV soubory.

3.4.4 Objektová reprezentace schematizace

Objektová datová struktura reprezentující celé povodí je tvořena jednotlivými třídami. Jejich objekty pak představují konkrétní kanály, subpovodí, hydrometeorologické měřicí stanice a další údaje nezbytné pro simulace. Na obrázku 3 je znázorněn třídní diagram popisující vybrané třídy, jejich metody a členy, které představují základ pro objektovou reprezentaci schematizací.

Po spuštění aplikace je schematizace povodí v rámci inicializace načítána do vnitřní objektové struktury. Z metody `collectMatDataCsv` jsou postupně hierarchicky volány



Obrázek 3: Třídní diagram zobrazuje strukturu modelu Math1D a její vybrané členy

další specializované metody. Nejprve `loadSchematization`, která načítání schematizace zastřešuje a provádí parsování vstupního XML souboru se schematizací. Poté následují metody `loadStationsFromXml`, `loadChannelsFromXml` a `loadSubbasinsFromXml`. Každá z nich se stará o načítání odpovídajícího typu elementů z XML souboru, vytváření objektů, jejich plnění načtenými daty a nakonec předávání těchto objektů do kolekcí, které jsou členy kořenové třídy `MatData` a nad kterými se pak provádí simulace.

Všechny třídní členy jsou z důvodu dodržení paradigmatu zapouzdření privátní a pro získávání či nastavování jejich hodnoty slouží pomocné veřejné metody tzv. `getter`y a `setter`y. Nyní budou popsány hlavní členy jednotlivých tříd.

MatData

- `sourceStations` – kolekce hydrologických stanic, které se nachází na začátku kanálu,
- `riverStations` – kolekce hydrologických stanic, které se nachází na konci kanálu,
- `weatherStations` – kolekce meteorologických stanic, které se nachází v oblasti spadající pod dílčí subpovodí,
- `channels` – kolekce kanálů, což jsou úseky vodních toků mezi jednotlivými soutoky,
- `subbasins` – kolekce subpovodí, což jsou oblasti v okolí kanálů, z jejichž povrchu stékají srážky do těchto kanálů,
- `measuredDischargeVolumes` – matice velikosti $m \times n$, kde m je počet časových kroků pro záznamy hodnot a n je počet měřících stanic ve schematizaci. Hodnoty jsou typu `double` a reprezentují naměřené průtoky pro dané stanice a časy,
- `precipitations` – kolekce párů typu klíč-hodnota, kde klíč je typu `time_t` vyjadřující datum a čas, pro který je hodnota určena, a hodnotou je kolekce typu `double` obsahující úhrny srážek pro odpovídající dobu. Uložení časů tímto způsobem bylo zvoleno kvůli úspory paměti a z důvodu zamezení duplicity hodnot,
- `options` – objekt obsahující nastavení modelu `Math1D`, průběhu simulace, datum a čas začátku a konce časového úseku, pro který se simulace provádí, zvolené výpočetní metody, použití/vynechání `adjustace` (viz podkapitola 2.3.5),
- `logger` – objekt udržující informace o logovacím kanálu. Voláním jeho metody `log` je do tohoto kanálu vložen požadovaný záznam,
- `nTimeSteps` – hodnota určující počet časových kroků, resp. řádků záznamů ve vstupních, potažmo výstupních datech,
- `minuteStep` – hodnota určující délku jednoho časového kroku. Pro současné zdroje vstupních dat je tato hodnota rovna 60,

Station

- `id` – jednoznačný číselný identifikátor stanice daného typu,
- `description` – popis stanice, její umístění apod.,
- `name` – název stanice, např. podle názvu obce,
- `location` – objekt obsahující zeměpisné souřadnice polohy `x`, `y` a `z` (výška),
- `clockStep` – hodnota určující délku jednoho časového kroku,
- `code` – kódové označení stanice,
- `precipitationFlag` – příznak určující, zda stanice poskytuje údaje o srážkách,
- `temperatureFlag` – příznak určující, zda stanice poskytuje údaje o teplotě,
- `snowFlag` – příznak určující, zda stanice poskytuje údaje o množství sněhu,
- `windVelocityFlag` – příznak určující, zda stanice poskytuje údaje o rychlosti větru,
- `radarFlag` – příznak určující, zda stanice poskytuje radarovou předpověď počasí,
- `isVirtual` – příznak určující, zda je stanice virtuální, nebo reálná,
- `spa1` – mezní hodnota průtoku, po jejímž překročení je vyhlášen 1. stupeň povodňové aktivity,
- `spa2` – mezní hodnota průtoku, po jejímž překročení je vyhlášen 2. stupeň povodňové aktivity,
- `spa3` – mezní hodnota průtoku, po jejímž překročení je vyhlášen 3. stupeň povodňové aktivity,
- `q` – průměrná roční hodnota průtoku,
- `waterGaugingCategory` – značí kategorii hlásných profilů, může nabývat hodnot `A`, `B`, nebo `N`,
- `hSpa1` – mezní hodnota vodního stavu, po jejímž překročení je vyhlášen 1. stupeň povodňové aktivity,
- `hSpa2` – mezní hodnota vodního stavu, po jejímž překročení je vyhlášen 2. stupeň povodňové aktivity,
- `hSpa3` – mezní hodnota vodního stavu, po jejímž překročení je vyhlášen 3. stupeň povodňové aktivity,
- `h` – průměrná roční hodnota vodního stavu,

- `chmuProfileId` – identifikátor hlásného profilu stanice,
- `channelIndex` – pořadí kanálu v kolekci kanálů.

Channel

- `id` – jednoznačný číselný identifikátor kanálu,
- `name` – označení kanálu,
- `sourceStationId` – ID hydrologické stanice na začátku kanálu,
- `sourceStationIndex` – pořadí hydrologické stanice na začátku kanálu v kolekci počátečních stanic,
- `stationId` – ID hydrologické stanice na konci kanálu,
- `stationIndex` – pořadí hydrologické stanice na konci kanálu v kolekci koncových stanic,
- `h` – nadmořská výška nejvýše položeného bodu kanálu,
- `d` – nadmořská výška nejnižší položeného bodu kanálu,
- `length` – délka kanálu,
- `slope` – sklon kanálu,
- `bankSlope` – sklon břehů,
- `depth` – hloubka koryta,
- `width` – šířka koryta,
- `n` – Manningův koeficient drsnosti,
- `subbasinId` – ID subpovodí, z něhož stékají srážky do daného kanálu,
- `subbasinIndex` – pořadí subpovodí, z něhož stékají srážky do daného kanálu v kolekci subpovodí,
- `upstreams` – kolekce pořadí přítoků v kolekci kanálů,
- `downstream` – pořadí následujícího kanálu v kolekci kanálů,
- `hydrograph` – objekt obsahující kolekce počátečních a koncových průtoků a kolekce počátečních a koncových vodních stavů kanálu.

Subbasin

- `id` – jednoznačný číselný identifikátor subpovodí,
- `name` – označení subpovodí,
- `area` – plocha subpovodí,
- `h` – nadmořská výška nejvýše položeného bodu subpovodí,
- `d` – nadmořská výška nejniže položeného bodu subpovodí,
- `length` – délka nejdelší trasy, kudy mohou srážky z povodí do vodního kanálu stékat,
- `slope` – sklon subpovodí,
- `baseFlow` – počáteční průtok,
- `cn` – hodnota CN křivky,
- `n` – Manningův koeficient drsnosti,
- `lai` – míra plochy subpovodí kryté stromovou klenbou,
- `initAbstraction` – počáteční ztráta (retence) na subpovodí,
- `timeConcentration` – čas, za který stečou srážky z nejvzdálenějšího místa subpovodí do vodního kanálu,
- `storageCoeff` – koeficient popisující, kolik vody dokáže subpovodí pojmout (vsáknout),
- `channelId` – ID kanálu, do něhož stékají srážky z daného subpovodí,
- `channelIndex` – pořadí kanálu, do něhož stékají srážky z daného subpovodí v kolekci kanálů,
- `weatherStationIndex` – pořadí meteorologické stanice v kolekci těchto stanic.

Options

- `startDate` – datum a čas posledního záznamu vstupních dat,
- `endDate` – datum a čas posledního záznamu vstupních dat,
- `minuteStep` – hodnota určující délku jednoho časového kroku vstupních dat,
- `meteoModelId` – ID meteorologického modelu,

- `meteoModelName` – název meteorologického modelu,
- `rr` – příznak, zda provádět srážko-odtokovou simulaci,
- `hd` – příznak, zda provádět hydrodynamickou simulaci,
- `rrSchemeId` – ID srážko-odtokové schematizace,
- `rrSchemeName` – název srážko-odtokové schematizace,
- `rrModelId` – ID srážko-odtokového modelu,
- `rrModelName` – název srážko-odtokového modelu,
- `rrType` – použitá srážko-odtoková metoda,
- `rrHdType` – použitá hydrodynamická metoda pro srážko-odtoky,
- `rrProfileShape` – tvar srážko-odtokového profilu,
- `hdModelId` – ID hydrodynamického modelu,
- `hdType` – použitá hydrodynamická metoda,
- `hdProfileShape` – tvar hydrodynamického profilu,
- `simulationId` – ID simulace,
- `fitting` – použití/vynechání adjustace.

3.5 Dokumentace

Pro zdokumentování kódu byl použit nástroj Doxygen⁴. Ten umožňuje automaticky generovat dokumentaci z komentářů, které jsou v podporovaném formátu. Výsledná dokumentace může být ve formátu HTML stránek, \LaTeX , RTF, XML, stránky pro program `man`, popř. DocBook⁵. Adresář s aktuální vygenerovanou dokumentací se nachází v kořenovém adresáři projektu. Obsahuje seznam tříd a všech jejich veřejných i privátních metod a členů. U vybraných metod je obsažen také slovní popis funkcionality.

⁴<http://www.doxygen.org>

⁵<http://www.docbook.org>

4 Parametrizace modelu Math1D

Parametrizace modelu představuje způsob, jak ovlivnit chování modelu v průběhu simulace. Aby toho bylo možné dosáhnout, obsahuje model Math1D řadu parametrů a atributů. K těmto parametrům je nutné nějakým způsobem přistupovat za účelem jejich nastavení nebo modifikace. V původní matlabovské implementaci byly všechny tyto parametry nastavovány manuálně, editací zdrojových kódů. To bylo dostatečné, jelikož šlo o proces návrhu a vývoje algoritmů a samotného modelu. Nyní, když jsou hlavní funkcionální a datové prvky modelu hotovy, je nejlepší příležitost abstrahovat parametry, které mají potenciál být modifikovány mimo samotný kód, a vytvořit tak určité rozhraní pro přístup k těmto parametrům.

Způsobů, jak toto rozhraní realizovat, existuje více. Možností by byly formáty pro serializaci dat, jako jsou např. JSON, YAML, popř. vytvoření API, které by umožnilo modifikovat parametry programově. Vzhledem k již zavedenému způsobu komunikace s okolím byl pro tento účel vybrán parametrizační soubor. Konkrétně se jedná o XML soubor, jakožto standardní univerzální formát pro výměnu informací. Jeho editace a správa může probíhat jednoduše manuálně, ale stejně tak i programově z jiné aplikace nebo externí utility k tomu určené. Vzhledem k nynějšímu malému počtu parametrů, které je potřeba modifikovat, je tento způsob plně dostatečný.

Robustnějším řešením by bylo vytvoření tzv. wrapperu, který by tvořil obecné rozhraní pro přístup k parametrům a zároveň adaptér pro ostatní způsoby přístupu. Těmi by mohly být už zmíněné souborové formáty, ale také webové služby nebo MPI zprávy pro využití na výpočetním clusteru. Toto řešení je náročnější na implementaci a především v současné době, kdy nejsou zcela zřejmé způsoby využití, není aktuální. Do budoucna ale jde o vhodnou volbu.

Modifikace parametrů je aktuálně využívána především pro potřeby automatické kalibrace a modelování neurčitostí. Při výpočtech simulací je používána metoda SCS-CN, jejímiž důležitými parametry jsou hodnoty CN křivek a Manningův koeficient drsnosti. Pro ně jsou nastavovány velikosti kroku, horní a dolní limity na základě jejich fyzikálních mezí a možné odchylky. Pro potřeby modelování neurčitostí se přidávají další možnosti nastavení uvedené v seznamu níže.

Parametry, které je aktuálně možné modifikovat pomocí parametrizačního souboru, jsou:

Model Math1D

- `minuteStep` – hodnota určující délku jednoho časového kroku.

Automatická kalibrace

- `cnLimitLower` – spodní hranice hodnoty CN křivky,
- `cnLimitUpper` – horní hranice hodnoty CN křivky,

- `nLimitLower` – spodní hranice hodnoty Manningova koeficientu drsnosti,
- `nLimitUpper` – horní hranice hodnoty Manningova koeficientu drsnosti,
- `stepCn` – velikost kroku změny hodnoty CN pro minimalizaci chyby,
- `stepN` – velikost kroku změny hodnoty N pro minimalizaci chyby,
- `pCn` – velikost změny hodnoty CN pro výpočet parciálních derivací podle CN,
- `pN` – velikost změny hodnoty N pro výpočet parciálních derivací podle N.

Modelování neurčitostí

- `quantiles` – kolekce kvantilů, pro které je modelování prováděno,
- `precRandType` – zvolená metoda randomizace,
- `daysMeasured` – počet dnů, pro které jsou srážky měřené,
- `daysPredicted` – počet dnů, pro které jsou srážky predikované,
- `valuesPerDay` – počet položek za den,
- `resultsDir` – cesta k adresáři pro výsledky,
- `precEnabled` – příznak určující, zda budou modelovány neurčitosti srážek,
- `precLimits` – kolekce limitních hodnot srážek,
- `cnEnabled` – příznak určující, zda budou modelovány neurčitosti CN křivek,
- `precDeviation` – odchylka od hodnoty srážek,
- `cnLimitLower` – spodní hranice hodnoty CN křivky,
- `cnLimitUpper` – horní hranice hodnoty CN křivky,
- `cnDeviation` – odchylka od hodnoty CN křivky,
- `nEnabled` – příznak určující, zda budou modelovány neurčitosti Manningova koeficientu drsnosti,
- `nLimitLower` – spodní hranice hodnoty Manningova koeficientu drsnosti,
- `nLimitUpper` – horní hranice hodnoty Manningova koeficientu drsnosti,
- `nDeviation` – odchylka od hodnoty Manningova koeficientu drsnosti,
- `stepN` – velikost kroku změny hodnoty N pro minimalizaci chyby,

- p_{CN} – velikost změny hodnoty CN pro výpočet parciálních derivací podle CN,
- p_N – velikost změny hodnoty N pro výpočet parciálních derivací podle N.

Stejným principem je možné parametrizovat libovolné další parametry a nastavení modelu nebo jeho součástí a tím umožnit rozšíření modelu, popř. přidání dalších modulů nebo provádění tzv. what-if analýz srážko-odtokových simulací.

5 Paralelizace modelu Math1D a jeho součástí

Výpočet samotné simulace na modelu Math1D je poměrně rychlý. I na průměrném osobním počítači nebo laptopu trvá při standardním rozsahu vstupních dat (5 + 2 dny) řádově jednotky sekund. V praxi je ale vyžadováno, aby byla s příchodem nových hydrologických a meteorologických dat automaticky aktualizována všechna povodí. Pokud se k tomu přidají ostatní rámce, jako automatická kalibrace nebo modelování neurčitostí, které mnohonásobně spouští simulaci modelu a jsou tedy výpočetně velmi náročné, mohl by čas potřebný k sekvenčnímu zpracování na výše zmíněných strojích vzrůst na desítky minut, nebo dokonce hodiny.

V budoucnu by rovněž mělo být dostupné manuální spouštění simulací pro koncové uživatele. Ten by zadal požadované parametry a odeslal požadavek na příslušnou simulaci. Výsledky by mu měly být zobrazeny v rozumně krátkém čase, proto by nebylo vhodné ani praktické, aby simulace trvala příliš dlouho. Tento případ užití se předpokládá např. pro pracovníky řešící krizové řízení, kteří provádí tzv. what-if analýzy a variantní řešení pro různé situace.

Za účelem splnění těchto vytyčených požadavků bylo potřeba analyzovat možnosti a učinit potřebné kroky pro urychlení běhu, a tím zkrácení času potřebného k provedení samotné simulace na modelu Math1D i automatické kalibrace a modelování neurčitostí. Řešením tohoto problému se ukázala být paralelizace vhodných částí jednotlivých algoritmů.

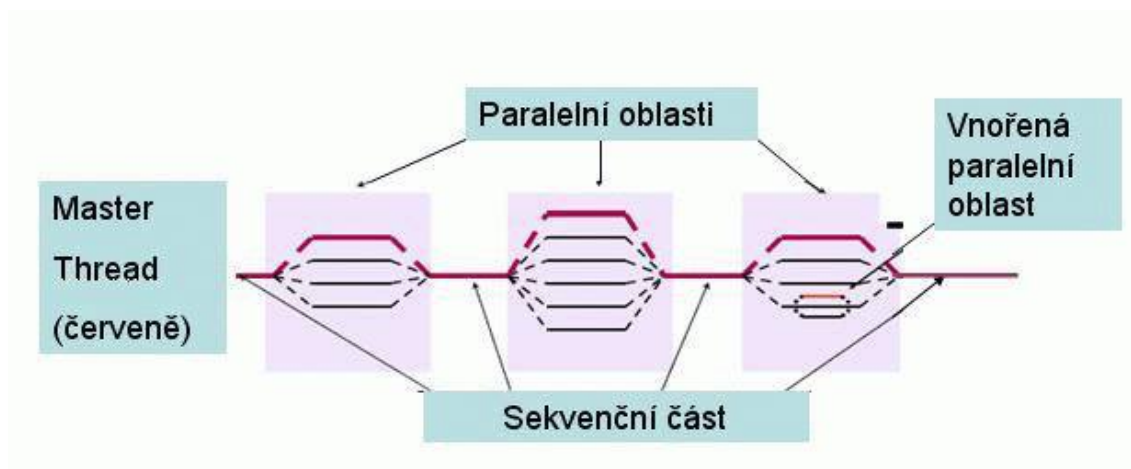
Paralelizace spočívá v takové úpravě programu, kdy je jeho průběh rozvětven do několika (mnoha) částečně, nebo plně nezávislých částí (vláken/procesů), které mohou být zpracovány a vykonávány vícejádrovým procesorem současně. Tímto se zkrátí celkový čas běhu programu o příslušný podíl odpovídající míře paralelizace. Princip běhu paralelního programu je patrný z obrázku 4.

V následujících dvou podkapitolách jsou stručně popsány dva velmi rozšířené modely pro paralelní programování – se sdílenou pamětí (OpenMP) a zasíláním zpráv (MPI). Každý z nich funguje na jiném principu a každý z nich je proto vhodný pro jinou situaci. Prvním z nich je OpenMP.

5.1 OpenMP

Jedná se o programovou knihovnu a sadu direktiv pro překladače jazyků C, C++ a Fortran, které umožňují rozdělení kódu do programových vláken⁶. Ta běží v rámci jednoho procesu, není zde tedy režie nutná pro vytvoření a správu požadovaného počtu procesů, ale pouze režie pro správu vláken, která je mnohem nižší. Další a zároveň klíčovou vlastností OpenMP je sdílení společné paměti mezi vlákny. Díky ní jsou vlákna schopna komunikovat, resp. sdílet data, se kterými pracují. Okomentovaná ukázka paralelního kódu s využitím OpenMP je ve výpisu 2.

⁶<http://openmp.org/>



Obrázek 4: Princip běhu paralelního programu

```

int nthreads, tid ;
// Je vytvořena množina vláken, z nichž každé má svou privátní proměnnou tid
#pragma omp parallel private(tid)
{
    // Každé vlákno zjistí své ID
    tid = omp_get_thread_num();

    // Pouze master vlákno zjistí aktuální počet vláken a hodnotu uloží
    if (tid == 0)
    {
        nthreads = omp_get_num_threads();
    }
} // Vlákná se připojí k master vláknu a jsou ukončeny

// Algoritmus konvoluce srazek a jednotkového hydrogramu
int j, j1, k;
// Vypočet velikosti dávky dat pro 1 vlákno
size_t chunk = m / nthreads;
// Je vytvořena množina vláken, se sdílenými i privátními proměnnými
#pragma omp parallel shared(Q,pef,q1,chunk) private(j, j1, k)
{
    // paralelní for – iterace jsou staticky rozděleny mezi vlákna po davce
#pragma omp for schedule(static, chunk)
for(j = 0; j < ((int)(m + n) - 1); j++)
    {
        j1 = j;
        Q[j]=0;

        for( k = 0; k < (int)n; k++)
        {
            if (j1 >= 0 && j1 < (int)m)
            {
                Q[j] += (pef[j1]*q1[k]);
            }
        }
    }
}

```



```

        }
        j1--;
    }
}

```

Výpis 2: Ukázka části kódu modelu Math1D paralelizovaného pomocí OpenMP

5.2 MPI

Druhým modelem paralelního programování je MPI. Jedná se o technologii určenou pro paralelní výpočty na clusterech. Po inicializaci je spuštěn požadovaný počet plnohodnotných procesů, což s sebou přináší nezanedbatelnou režii. V procesech je vykonáván totožný program, přičemž jeho vnitřní průběh je řízen identifikátory procesů a každý proces tak zpracovává určitou část kódu. Komunikace a předávání dat jsou založeny na zasílání zpráv mezi jednotlivými procesy, což je naznačeno na obrázku 5. Toto předávání zpráv je ale poměrně časově náročná operace, proto by vždy měla být snaha o minimalizaci množství komunikace mezi procesy. K té přispívají i tzv. kolektivní operace⁷. Klíčovou vlastností MPI je to, že jediný program, ač spuštěný ve více procesech, může běžet na několika nezávislých počítačích a využívat jejich výpočetní zdroje.

5.3 Superpočítač Anselm

Aby mělo smysl spouštět paralelní aplikace a především bylo možné plně využít paralelizovaných algoritmů, je nutné mít k dispozici hardware, který to umožňuje. Tuto podmínku v dnešní době splňují takřka všechny novější osobní počítače a laptopy. Ve většině případů jsou totiž tyto stroje vybaveny vícejádrovými procesory, nebo dokonce grafickými procesory s podporou technologií pro spouštění programů, tzv. GPGPU (např. CUDA⁸, OpenCL⁹). Oproti sekvenčním aplikacím tak může i běžný uživatel využít výrazné zrychlení úloh.

Pro výpočetně velmi náročné a rozsáhlé úlohy je ale potřeba strojů s mnohem vyšším tzv. masivním výpočetním výkonem – superpočítačů. Jedná se o cluster samostatných navzájem propojených výkonných počítačů.

To je případ clusteru Anselm spravovaného Národním superpočítačovým centrem IT4Innovations¹⁰. Anselm (malý cluster) byl uveden do provozu v květnu 2013 a byl dodán společností BULL.

Anselm se skládá z 209 výpočetních uzlů, celkově s 3 344 jádry a 15 TB paměti RAM, poskytujících ve špičce více než 94 Tflop/s výpočetního výkonu. To odpovídá výpočetní síle přibližně 3 000 průměrných domácích počítačů¹¹. Každý z uzlů je tvořen velmi výkonným počítačem se standardní architekturou x86-64 a disponuje 16 jádry (2

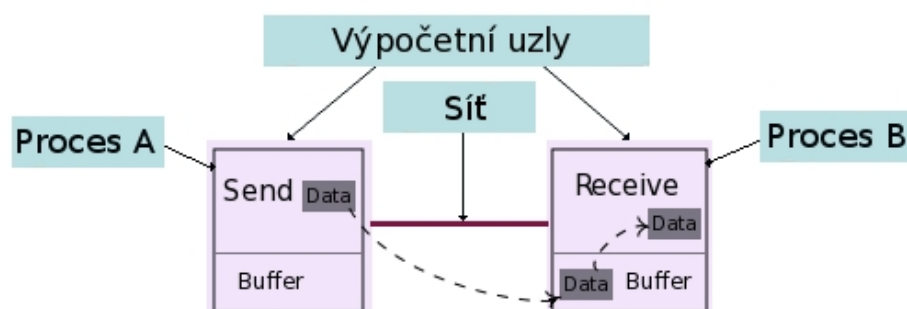
⁷<http://www.open-mpi.org>

⁸<http://docs.nvidia.com/cuda>

⁹<https://developer.nvidia.com/opencv>

¹⁰<http://www.it4i.cz>

¹¹Založeno na expertním odhadu



Obrázek 5: Princip přenosu MPI zprávy

osmijádrovými procesory s frekvencí 2.4GHz a mikroarchitekturou Intel Sandy Bridge), 64 GB paměti RAM a 500 GB lokálního úložiště. Některé uzly jsou navíc akcelerovány¹². Všechny uzly jsou mezi sebou propojeny pomocí speciální sítě, tzv. Infiniband¹³.

Operačním systémem na Anselmu je Linux, konkrétně jde o bullx Linux Server release 6.3, který je založen na linuxové distribuci Red Hat Enterprise Linux. Mimo to Anselm nabízí velké množství softwarového vybavení pro vývoj, analýzu, modelování a simulace v mnoha rozličných oblastech výzkumu a průmyslu.

5.4 Paralelizace modelu Math1D

Záměrem a jedním z cílů této práce bylo analyzovat možnosti paralelizace modelu Math1D a připravit jej pro spouštění v prostředí HPC. Uvažovány byly především programovací modely se sdílenou pamětí (OpenMP) a zasláním zpráv (MPI).

Průběh simulace byl podroben výkonové analýze (tzv. profilování) pomocí profilovacího nástroje. Ta měla změřit, kolik procesorového času trvá vykonání té či oné metody, nebo bloku kódu. Je zřejmé, že pokud se jedná o opakované volání, např. v cyklu, konečný čas je násobkem času potřebného pro jeden průchod metodou v závislosti na počtu iterací cyklu, ze kterého je volána. Výsledkem této analýzy je seznam metod (částí kódu) a příslušných relativních časů spotřebovaných v dané metodě vyjádřených v procentech. Podle této procentuální hodnoty pak lze snadno identifikovat metody, ve kterých tráví procesor nejvíce času. Takové metody jsou nazývány kritickými úseky kódu a jsou vhodnými kandidáty pro paralelizaci. Pokud totiž zkrátíme čas potřebný pro vykonání časově nejnáročnější metody programu, na výsledný celkový čas běhu programu to bude mít největší možný vliv. Dvě časově nejnáročnější operace během výpočtu simulace před paralelizací jsou identifikovány na obrázku 6. Z obrázku je patrné, že tyto dvě operace

¹²<https://docs.it4i.cz>

¹³<http://www.infinibandta.org>

| | | |
|---------|-----|--|
| < 0.1 % | 226 | <code>for(int j = 0; j < m + n - 1; j++)</code> |
| | 227 | <code>{</code> |
| | 228 | <code>int j1 = j;</code> |
| | 229 | <code>double tmp = 0;</code> |
| 3.3 % | 230 | <code>for(int k = 0; k < n; k++)</code> |
| | 231 | <code>{</code> |
| 1.6 % | 232 | <code>if(j1 >= 0 && j1 < m)</code> |
| | 233 | <code>{</code> |
| 51.9 % | 234 | <code>tmp += (pef[j1]*q1[k]);</code> |
| | 235 | <code>}</code> |
| 0.8 % | 236 | <code>j1 -= 1;</code> |
| | 237 | <code>}</code> |
| | 238 | <code>//Q[j] += pef[k] * q1[j - k - 1];</code> |
| 26.8 % | 239 | <code>Q[j] = tmp;</code> |
| 4.2 % | 240 | <code>}</code> |
| | 241 | <code>}</code> |

Obrázek 6: Barevně označeny kritické části kódu

zabraly dohromady téměř 80% času celé simulace. Nabízelo se tedy přetransformovat a paralelizovat cyklus, ve kterém se tyto kritické operace nacházely.

Jak bylo zmíněno na úvod této kapitoly, průběh jedné simulace se standardní velikostí vstupních dat je poměrně krátký, a proto se jeví jako nejvhodnější použití technologie OpenMP, která má na rozdíl od MPI mnohem menší režii a v kontextu použití simulací modelu Math1D dává větší smysl.

Po provedení vhodných úprav kódu a začlenění direktiv OpenMP (viz výpis 2), převážně v metodě `scsMethod` modelu, bylo potřeba otestovat míru škálovatelnosti a rychlost provádění simulací. Škálovatelnost je u OpenMP řízena především nastavením počtu vláken, která se na výpočtu podílí. Test probíhal na jednom z výpočetních uzlů superpočítače Anselm. Tento uzel disponuje 16 výkonnými výpočetními jádry, proto byl test prováděn jako několik sad sekvencí simulací vždy se stejnými vstupními daty v rámci sady, ale různým počtem přiřazených vláken pro každé spuštění až do konečného počtu 16 vláken. Sady byly tvořeny sekvencemi simulací se vstupními daty s délkou 7 dní a 2 měsíce, následovaly sekvence automatických kalibrací s daty na 7 dní bez optimalizací překladače a poté s optimalizacemi (přepínač překladače `-O3`). Výsledné časy výpočtů jsou uvedeny v tabulce 3. Pro porovnání je uveden také čas simulace původní implementace v MATLABu a časy kalibrací zkompileovaných pomocí Intel C++ překladače.

Z výsledků vyplývá, že pro standardní interval vstupních dat na 7 dní je nejvýhodnější spouštět simulace s použitím 8 vláken. Nejlepší časy této konfigurace jsou důsledkem vnitřní NUMA architektury výpočetního uzlu Anselmu, který se skládá ze 2 osmijádrových procesorů. Projevuje se zde větší časová prodleva při komunikaci mezi procesory při použití všech 16 jader uzlu, která se však při delším běhu s větším množstvím dat rozprostře a v celkovém čase již není tak patrná.

| Počet vláken | 7 dní | 2 měsíce | 7 dní s kalibrací | 7 dní s kalibrací a optimalizacemi | 7 dní s kalibrací a optimalizacemi, Intel kompilátor |
|--------------|------------|-----------|-------------------|------------------------------------|--|
| 1 | 0.531623 s | 32.3380 s | 264.429 s | 63.7680 s | 62.4006 s |
| 2 | 0.385408 s | 18.5771 s | 157.743 s | 43.4596 s | 43.4310 s |
| 4 | 0.27465 s | 11.5134 s | 96.2565 s | 31.8298 s | 30.4321 s |
| 8 | 0.260878 s | 8.19633 s | 72.1598 s | 31.0404 s | 24.4763 s |
| 16 | 0.374548 s | 6.97017 s | 88.1866 s | 52.9296 s | 30.2631 s |
| MATLAB | 1.28 s | - | - | - | - |

Tabulka 3: Výsledné časy výpočtů testu škálovatelnosti modelu Math1D s OpenMP. Pro lepší názornost připojeny i časy sekvenční kalibrace využívající paralelní model Math1D

5.5 Paralelizace automatické kalibrace

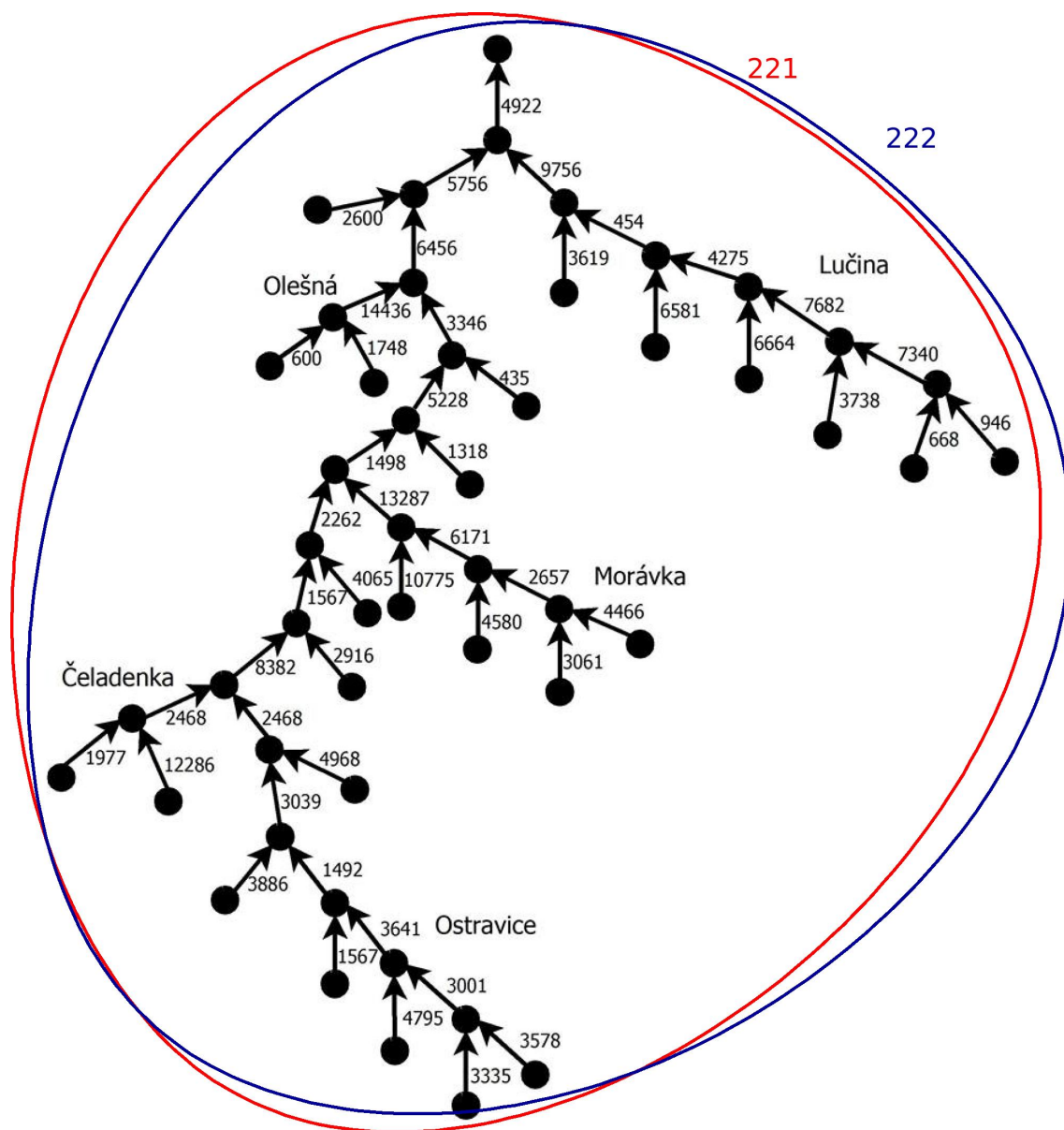
Při paralelizaci automatické kalibrace bylo vycházeno ze sekvenční implementace, která byla vytvořena dřívější konverzí z matlabovského kódu. Podstatnou částí algoritmu je programová smyčka, v níž dochází k opakovanému mnohonásobnému spouštění simulací modelu Math1D. Jelikož model už je paralelizován pomocí OpenMP, nemělo by již další použití této technologie velký přínos. Za účelem dalšího zrychlení běhu systému proto bylo potřeba využít jiný přístup. V případě kalibrace modelu se vybízí použití MPI.

Aktuálně existují 3 implementace algoritmu automatické kalibrace, zejména pro experimentální a testovací účely:

1. kalibrace je spuštěna zcela nezávisle pro všechny hydrologické měřicí stanice na celém povodí (5.5.1). Na obrázku 9 je povodí se dvěma měřicími stanicemi kalibrované podle implementace 1.
2. kalibrace je spuštěna pouze pro závěrový profil povodí (nejspodnější hydrologická měřicí stanice) a bude kalibrovat přes všechny kanály povodí (5.5.2). Na obrázku ?? je povodí se dvěma měřicími stanicemi kalibrované podle implementace 2.
3. kalibrace je spuštěna nejprve pro první měřicí hydrologickou stanicí nejvýše na povodí a kalibruje přes kanály, které leží na toku proti proudu nad ní. Výsledné hodnoty jsou předány do další iterace kalibrující nižší stanicí (po proudu) skrze kanály, které jsou proti proudu nad touto stanicí a zároveň ještě doposud nebyly kalibrovány v žádném z předchozích kroků. Toto se opakuje pro celé povodí (5.5.3). Na obrázku ?? je povodí se dvěma měřicími stanicemi kalibrované podle implementace 3.

5.5.1 Implementace 1

Průběh kalibrace modelu je následující. Po úvodní inicializaci, kdy probíhá vytvoření a příprava procesů, je jeden z procesů určen jako tzv. master – řídicí proces, ten pokračuje



Obrázek 7: Povodí se dvěma měřicími stanicemi kalibrované podle 1. implementace kalibrace

v parsování vstupních argumentů. Master následně rozešle ostatním procesům vstupní parametry kalibrace MPI zprávami.

Každý proces si vytvoří vlastní lokální instanci modelu Math1D a spustí první referenční simulaci. Po jejím úspěšném dokončení spustí samotnou kalibraci s dodanými počátečními parametry a vypočítá počáteční chybu schematizace. Podle dostupných informací o celkovém počtu procesů a vlastním pořadí mezi procesy každý z nich provede příslušný počet iterací vnější smyčky kalibrace. Tyto iterace opakované pro každou měřicí stanici, resp. měřený úsek toku, obsahují několik kroků:

1. Připraví se aktuální vstupní data a parametry pro danou iteraci,
2. provede se výpočet chyby schematizace a verifikuje, zda je vypočtená chyba v mezích přednastavené požadované tolerance,
3. pokud je chyba větší, než dovoluje tolerance, počítají se derivace modifikovaných parametrů modelu, tzn. hodnot CN křivky a Manningova koeficientu drsnosti,
4. na základě těchto derivací jsou pomocí minimalizační funkce vypočítány nové hodnoty parametrů a je opět spuštěna simulace a verifikována chyba,
5. tato smyčka se opakuje, dokud není rozdíl aktuální chyby a chyby z minulé iterace v dané toleranci,
6. pokud ano, je naposled spuštěna simulace s již kalibrovanými parametry a vypočten rozdíl mezi chybami před a po kalibraci.

Poté se celý postup opakuje pro následující měřený úsek. Jakmile všechny procesy dokončí jim přiřazené iterace, vypočítají rozdíl chyby schematizace před a po kalibraci a algoritmus kalibrace končí.

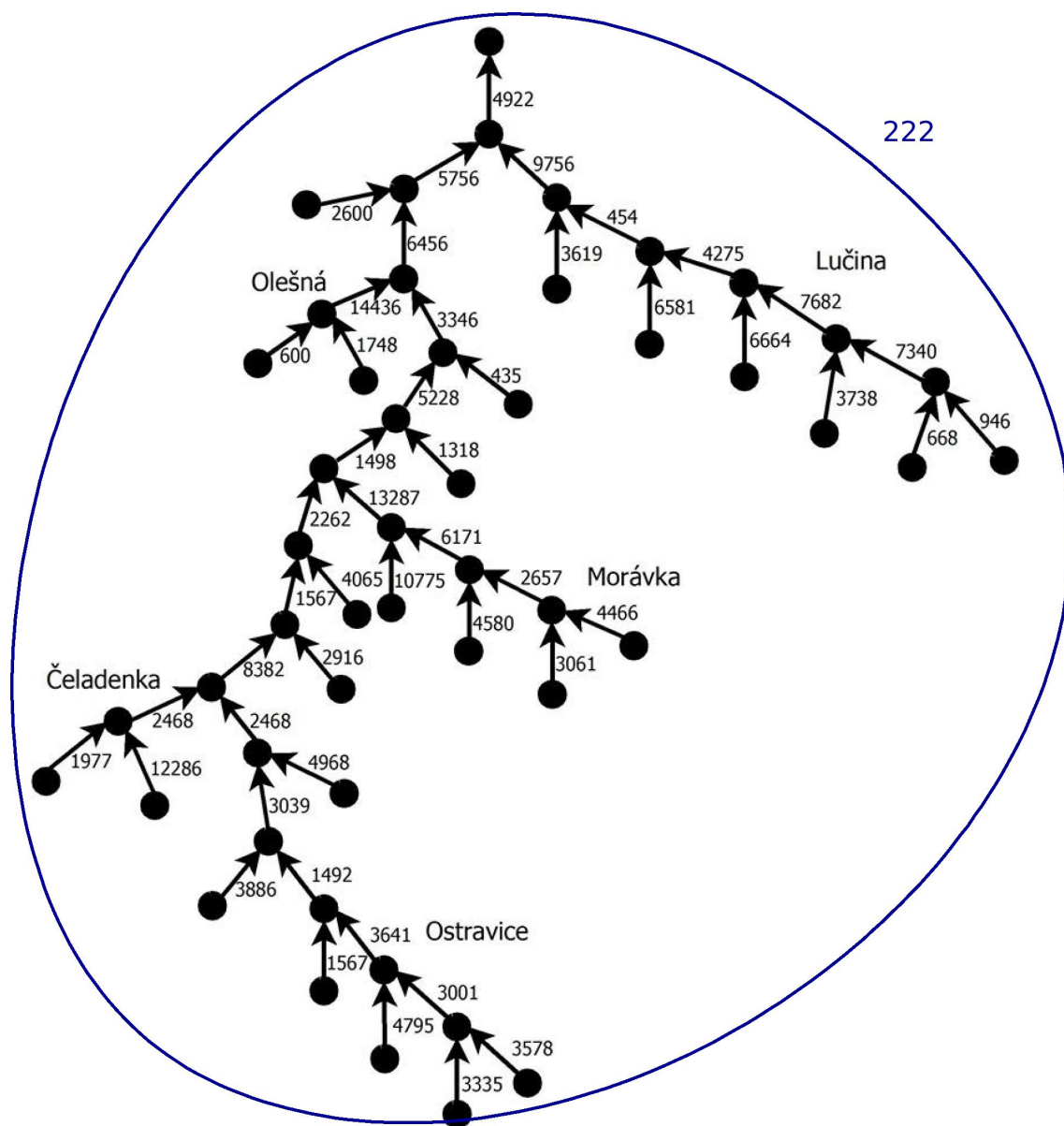
Test škálovatelnosti kalibrace byl prováděn s testovacími daty pro 2 měřicí stanice na klasickou dobu 5 + 2 dny, tzn., nemělo smysl používat více než 2 výpočetní uzly. Naměřené časy pro jednotlivé kombinace počtu MPI procesů kalibrace a OpenMP vláken modelu Math1D jsou uvedeny v tabulce 4.

Z naměřených časů vyplývá, že se doba výpočtu za použití výše popsaného algoritmu kalibrace s pomocí MPI s každým začleněným uzlem zkrátí na příslušný podíl času sekvenčního běhu a celkového počtu použitých uzlů. Limitem je z principu kalibrace pouze počet měřených úseků – rozdělení výpočtů na více uzlů, než je počet úseků, již nemá smysl.

Tato implementace má výhodu v tom, že se pro každou měřicí stanici započítává celé povodí, tzn. dílčí kalibrace pro část povodí příslušící dané měřicí stanici je velmi přesná. Nevýhodou je naopak komplikované kombinování těchto částečných kalibrací a získávání jediné nejlepší pro spuštění simulací na celém povodí.

5.5.2 Implementace 2

Druhá implementace se liší tím, že délka běhu nezávisí na počtu měřících stanic. Kalibrace je prováděna pouze pro poslední měřicí stanici povodí - závěrový profil. V tomto případě



Obrázek 8: Povodí se dvěma měřicími stanicemi kalibrované podle 2. implementace kalibrace

bylo tedy nutné řešit paralelizaci jiným způsobem. Jako nejvhodnější se ukázal následující postup:

1. Po počáteční inicializaci a nastavení vstupních hodnot parametrů jsou podle počtu procesů a počtu kanálů, pro něž se budou modifikovat parametry, dynamicky vypočítány intervaly prvků, které budou jednotlivé procesy zpracovávat,
2. master proces se potom stará o ukončovací podmínku, výpočet minimalizační funkce, shromažďuje od ostatních procesů vypočtené parciální derivace a spolu s ostatními procesy počítá také vlastní příděl derivací,
3. dělení práce spočívá v tom, že v každé iteraci si každý z procesů vypočítá parciální derivace pro svůj příslušný rozsah kanálů, výsledky odešle master procesu, čeká, dokud master proces nevypočítá chybu a nové modifikované parametry, které následně rozešle,
4. ukončovací podmínka je stejná jako v první implementaci, tzn., smyčka se opakuje, dokud není rozdíl posledních dvou chyb dostatečně malý.

Nevýhodou této implementace je fakt, že do výsledku kalibrace nejsou zohledněna data měřicích stanic uprostřed povodí.

V této implementaci jsou využívány kolektivní MPI metody `MPI_Bcast` a `MPI_Gatherv`. První umí rozeslat z master procesu všem ostatním procesům požadovaná data. Druhá zmíněná sbírá data odesílaná ze všech procesů směrem k master procesu a před doručením z nich seskládá pole. Velikost dílčích posílaných zpráv navíc může být různá.

Počet MPI procesů, které lze u této implementace využít, je větší, než u předchozí, protože není limitován poměrně malým počtem měřicích stanic, ale počtem kanálů a subpovodí v rámci celého povodí.

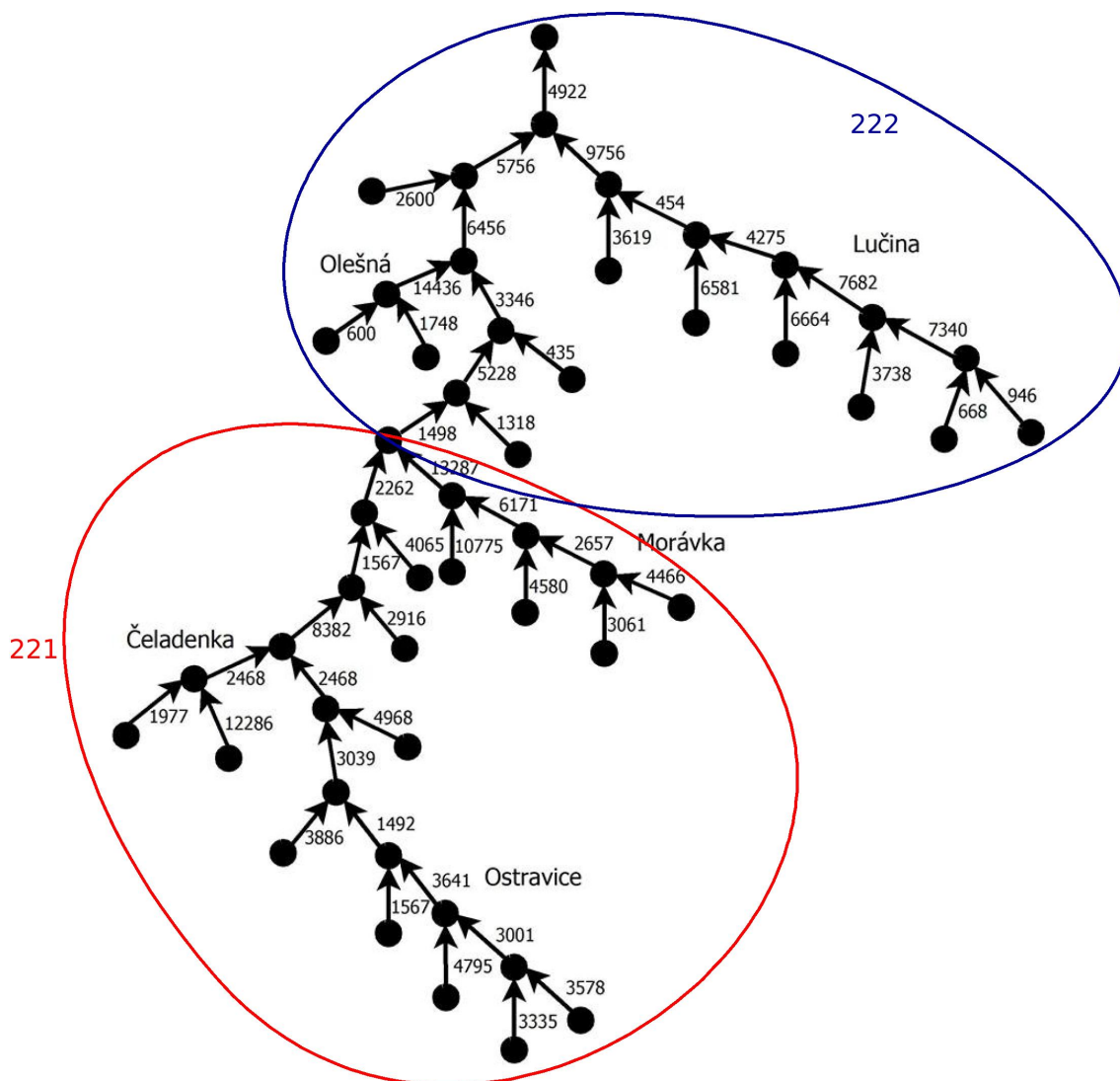
Naměřené výsledné časy jednotlivých konfigurací jsou v tabulce 4. Nejlepších časů kalibrace bylo dosahováno s maximálním počtem testovaných MPI procesů, nicméně použití více procesů, než je počet kanálů, resp. subpovodí, by bylo opět zbytečné.

5.5.3 Implementace 3

Třetí implementace se liší od předchozí tím, že opět obsahuje smyčku iterující přes měřicí hydrometeorologické stanice. Další změnou je selekce kanálů. V každé ze zmíněných iterací jsou pro kalibraci vybírány pouze kanály, které jsou proti proudu nad stanicí, pro kterou probíhá kalibrace, a navíc pouze ty, které se ještě kalibrace neúčastnily.

Iteraci přes měřicí stanice nelze paralelizovat jako v implementaci 1. Než může být prováděna kalibrace oblasti pro stanici níže v povodí, musí být zkalibrovány oblasti proti proudu. Jejich dílčí výsledky jsou předány do následující iterace a teprve na jejich základě je kalibrována oblast po proudu.

Výhodou této implementace je jednoduché určení výsledné kalibrace, kladem je i zohlednění všech měřicích stanic povodí.



Obrázek 9: Povodí se dvěma měřícími stanicemi kalibrované podle 3. implementace kalibrace

| MPI procesů | OMP vláken | Implementace 1 | Implementace 2 | Implementace 3 |
|-------------|------------|-----------------|-----------------|-----------------|
| 1 | 1 | 22 min 42.96 s | 16 min 15.757 s | 11 min 31.489 s |
| 1 | 2 | 16 min 10.103 s | 12 min 22.591 s | 8 min 12.372 s |
| 1 | 4 | 11 min 10.027 s | 8 min 33.291 s | 5 min 39.493 s |
| 1 | 8 | 10 min 22.677 s | 7 min 46.767 s | 5 min 21.055 s |
| 1 | 10 | 11 min 8.234 s | 8 min 3.204 s | 5 min 39.553 s |
| 1 | 12 | 11 min 54.745 s | 8 min 36.977 s | 6 min 6.553 s |
| 1 | 14 | 12 min 39.243 s | 9 min 20.835 s | 6 min 31.827 s |
| 1 | 16 | 17 min 3.62 s | 23 min 40.36 s | 8 min 8.238 s |
| 2 | 1 | 17 min 53.18 s | 8 min 43.428 s | 6 min 44.498 s |
| 2 | 2 | 12 min 15.331 s | 5 min 56.744 s | 4 min 48.585 s |
| 2 | 4 | 9 min 12.368 s | 4 min 12.103 s | 3 min 20.207 s |
| 2 | 8 | 7 min 48.045 s | 4 min 33.358 s | 3 min 38.805 s |
| 2 | 10 | 8 min 49.213 s | 4 min 23.027 s | 6 min 47.491 s |
| 2 | 12 | 8 min 58.589 s | 4 min 45.731 s | 3 min 27.569 s |
| 2 | 14 | 10 min 47.118 s | 4 min 48.53 s | 3 min 44.067 s |
| 2 | 16 | 14 min 53.662 s | 12 min 13.529 s | 5 min 7.157 s |
| 3 | 8 | - | 3 min 13.585 s | 2 min 12.329 s |
| 4 | 8 | - | 3 min 25.91 s | 2 min 14.887 s |
| 8 | 8 | - | 1 min 45.871 s | - |
| 16 | 8 | - | 1 min 6.6989 s | - |

Tabulka 4: Výsledné časy výpočtů testu škálovatelnosti všech implementací automatické kalibrace modelu Math1D pomocí MPI

Způsob paralelizace je shodný s druhou implementací. Naměřené časy jsou opět v tabulce 4.

Z hlediska výpočetního výkonu je třeba zdůraznit, že u všech implementací je při každém jednotlivém výpočtu derivace každého parametru i jejich minimalizací opakovaně spouštěna a vypočítávána simulace na Math1D modelu. V průběhu kalibrace je tedy simulace spouštěna mnohokrát, v závislosti na počtu měřených úseků toků, na velikosti počáteční chyby parametrů schematizace a také nastavení jednotlivých kroků a limitů modifikovaných parametrů.

Z toho vyplývá, že paralelizace kalibrace je zejména pro budoucí využití, kdy bude s velkou pravděpodobností docházet k modifikacím většího počtu parametrů, tím i zvýšení počtu simulací, opravdu vhodným předmětem zájmu.

5.6 Paralelizace neurčitostních simulací

Jelikož byla pro modelování neurčitostí vybrána metoda Monte Carlo, v rámci které jsou během velkého množství iterací prováděny simulace, byla pro paralelizaci zvolena

technologie MPI. Podrobně je paralelizace neurčitostí popsána v paralelně vznikající práci [4].

6 Závěr

Cílem práce bylo navrhnout a implementovat stávající verzi modelu do jazyka vhodnějšího pro další vývoj a nasazení. To bylo splněno hned během prvotních fází této práce. Bylo tomu tak zejména proto, že nová implementace byla základním stavebním kamenem pro vývoj dalších probíraných komponent, automatické kalibrace a modelování neurčitostí. Nová implementace je zcela funkční, přijímá totožné vstupy a poskytuje totožné výstupy jako implementace minulé. Aby ale mohl být vývoj modelu prohlášen za ukončený, chybí ještě mnoho práce, zejména ladění, optimalizace a refaktORIZACE.

Druhým úkolem bylo připravit nový model pro spouštění v prostředí HPC. Tento cíl se také povedl splnit, nicméně v kódu jsou stále značné výkonové rezervy, které by se daly přeměnit na výsledné zrychlení aplikace. Jde především o ještě hlubší pochopení principů paralelního programování v rámci jednotlivých paralelních technologií.

Během práce také probíhala plodná a inspirativní spolupráce v rámci celého týmu výzkumné skupiny i s mezinárodní účastí.

Následuje několik postřehů a podnětů pro budoucí práci v rámci vývoje modelu Math1D a jeho součástí.

V současné době se pokračuje ve vývoji a optimalizaci kódu modelu Math1D i automatické kalibrace a modelování neurčitostí, tudíž lze očekávat další zrychlení běhu výpočtů jednotlivých komponent.

Aktuálně jde o intenzivní práci na paralelním zpracování a optimalizaci procesu načítání a inicializace vstupních dat modelu Math1D. To by mělo do budoucna eliminovat, nebo alespoň omezit problém se specifiky NUMA architektury procesorů, jimiž jsou vybaveny výpočetní uzly superpočítače Anselm.

Tuto práci lze považovat za výchozí referenci k nové implementaci srážko-odtokového modelu Math1D a výchozí bod dalšího vývoje, který by si model jistě zasloužil.

Nová implementace byla kvůli přehlednosti během konverze tvořena s ohledem na zpětnou kompatibilitu jednotlivých identifikátorů. V aktuální fázi by byla velmi vhodná také refaktORIZACE kódu, zpřehlednění názvů identifikátorů, odstranění nevyužívaných částí, popř. odladění a optimalizace těch využívaných, nebo plánovaných k využívání. Zváženo by do budoucna mohlo být kompletní oddělení verzí modelu pro Windows a pro Linux za účelem jednodušší distribuce a správy obou verzí, popř. vytvoření platformově nezávislého projektu pro některé vývojové prostředí, např. zmíněný CodeBlocks, apod.

Dalším významným krokem ve vývoji by v rámci optimalizace mohlo být použití pokročilých knihoven pro práci s matematickými strukturami, např. Intel MKL nebo IT++, což by mělo přinést značné zrychlení výpočtů.

7 Reference

- [1] BEVEN, Keith. J. *Rainfall-Runoff Modelling: The Primer*. 2nd edition. New Jersey: John Wiley & Sons, únor 2012. ISBN 978-0-470-71459-1.
- [2] BRUTSAERT, Wilfried. *Hydrology: An Introduction*. Ilustrované vydání, dotisk. Cambridge: Cambridge University Press, 2005.
- [3] FELDMAN, Arlen D. *Hydrologic Modeling System HEC-HMS: Technical Reference Manual*. Davis, CA: U. S. Army Corps of Engineers - Hydrologic Engineering Center, březen 2000.
- [4] GOLASOWSKI, Martin. *Modelování neurčitostí ve srážko-odtokových modelech*. Ostrava, 2014. Diplomová práce na Fakultě elektrotechniky a informatiky Vysoké školy báňské - Technické univerzity Ostrava na Katedře informatiky. Vedoucí diplomové práce Ing. Štěpán Kuchař.
- [5] HEC-USACE. *Hydrologic Engineering Center - US Army Corps of Engineers* [online]. c2012 [cit. 2014-04-24]. <<http://www.hec.usace.army.mil>>.
- [6] MARTINOVIČ, Jan, KUCHAR, Štěpán, VONDRÁK, Ivo, VONDRÁK, Vít, ŠÍR, Boris a UNUCKA, Jan. *Multiple scenarios computing in the flood prediction system FLOREON +*. In *Proceedings 24th European Conference on Modelling and Simulation. ECMS 2010*, 2010, pp. 182–188.
- [7] MARTINOVIČ, Jan, ŠTOLFA, Svatopluk, KOŽUSZNIK, Jan, UNUCKA, Jan a VONDRÁK, Ivo. *FLOREON—the system for an emergent flood prediction*. In *ECECFUBUTEC-EUROMEDIA*. Porto, Portugal, 2008.
- [8] MCCOOL, Michael, ROBISON, Arch D. a REINDERS, James. *Structured Parallel Programming: Patterns for Efficient Computation*. USA: Elsevier Science, 31. července 2012. ISBN 9780123914439.
- [9] STROUSTRUP, Bjarne. *The C++ Programming Language*. 4th edition. Boston: Addison-Wesley Professional, c2013. ISBN 0-321-56384-0.
- [10] VIEUX, Baxter E. *Distributed Hydrologic Modeling using GIS*. 4th edition. Addison-Wesley Professional, květen 2013. ISBN 978-0321563842.
- [11] VACHOVÁ, Jitka. *Model ALADIN* [online]. Plzeň: Západočeská univerzita v Plzni, 2009. [cit. 2014-04-24]. <[http://num.kma.zcu.cz/galerie/MM-prace/Galerie MM 2009/Vachova-Aladin-predpoved_pocasi.pdf](http://num.kma.zcu.cz/galerie/MM-prace/Galerie_MM_2009/Vachova-Aladin-predpoved_pocasi.pdf)>.

A Výpis souboru

```

<MatData>
  <SourceStations>
    <SourceStation>
      <Id>120</Id>
      <Description>virtual source for R10</Description>
      <Name>virtual</Name>
      <Location>
        <X>-456239.89079848</X>
        <Y>-456239.89079848</Y>
        <Z>0</Z>
      </Location>
      <ClockStep>3.40282347E+38</ClockStep>
      <Code>V_SRS_R10</Code>
      <PrecipitationFlag>>false</PrecipitationFlag>
      <TemperatureFlag>>false</TemperatureFlag>
      <SnowFlag>>false</SnowFlag>
      <WindVelocityFlag>>false</WindVelocityFlag>
      <RadarFlag>>false</RadarFlag>
      <Virtual>>true</Virtual>
      <Spa1>0</Spa1>
      <Spa2>0</Spa2>
      <Spa3>0</Spa3>
      <Q>0</Q>
      <WaterGaugingCategory>N</WaterGaugingCategory>
      <HSpa1>0</HSpa1>
      <HSpa2>0</HSpa2>
      <HSpa3>0</HSpa3>
      <H>0</H>
      <ChannelId>724</ChannelId>
      <ChannelIndex>49</ChannelIndex>
    </SourceStation>
  </SourceStations>
  <RiverStations>
    <RiverStation>
      <Id>222</Id>
      <Description>Slezska Ostrava</Description>
      <Name>Ostrava</Name>
      <Location>
        <X>-469549.35963</X>
        <Y>-469549.35963</Y>
        <Z>201.9</Z>
      </Location>
      <ClockStep>60</ClockStep>
      <Code>O4293000</Code>
      <PrecipitationFlag>>true</PrecipitationFlag>
      <TemperatureFlag>>false</TemperatureFlag>
      <SnowFlag>>false</SnowFlag>
      <WindVelocityFlag>>false</WindVelocityFlag>
      <RadarFlag>>false</RadarFlag>
      <Virtual>>false</Virtual>
      <Spa1>182</Spa1>

```

```
<Spa2>378</Spa2>
<Spa3>665</Spa3>
<Q>1</Q>
<WaterGaugingCategory>N</WaterGaugingCategory>
<HSpa1>290</HSpa1>
<HSpa2>400</HSpa2>
<HSpa3>530</HSpa3>
<H>0</H>
<ChannelId>724</ChannelId>
<ChannelIndex>49</ChannelIndex>
</RiverStation>
</RiverStations>
<BasinId>23</BasinId>
<WeatherStations>
  <WeatherStation>
    <Id>126</Id>
    <Name>Lucina</Name>
    <Location>
      <X>-460234.80054</X>
      <Y>-460234.80054</Y>
      <Z>0</Z>
    </Location>
    <Code>O1LUCI01</Code>
  </WeatherStation>
</WeatherStations>
<Channels>
  <Channel>
    <Id>748</Id>
    <Name>R480</Name>
    <SourceStationId>233</SourceStationId>
    <SourceStationIndex>25</SourceStationIndex>
    <StationId>234</StationId>
    <StationIndex>25</StationIndex>
    <H>603.4692</H>
    <D>494.8335</D>
    <Length>3578</Length>
    <Slope>0.0304</Slope>
    <BankSlope>-999.999</BankSlope>
    <Depth>-999.999</Depth>
    <Width>-999.999</Width>
    <N>-999.999</N>
    <IsJunction>>false</IsJunction>
    <SubbasinId>955</SubbasinId>
    <SubbasinIndex>1</SubbasinIndex>
    <Upstreams />
    <DownstreamId>747</DownstreamId>
    <DownstreamIndex>3</DownstreamIndex>
  </Channel>
</Channels>
<Subbasins>
  <Subbasin>
    <Id>955</Id>
    <Name>R480W480</Name>
    <Area>21.419</Area>
```

```
<H>960</H>
<D>494.8335</D>
<Length>3578</Length>
<Slope>0.25771</Slope>
<BaseFlow>0.024</BaseFlow>
<Cn>69.5</Cn>
<N>-999.999</N>
<Lai>-999.999</Lai>
<InitAbstraction>49.06</InitAbstraction>
<TimeConcentration>2.11</TimeConcentration>
<StorageCoeff>2.14</StorageCoeff>
<ChannelId>748</ChannelId>
<ChannelIndex>1</ChannelIndex>
<WeatherStationId>144</WeatherStationId>
<WeatherStationIndex>6</WeatherStationIndex>
</Subbasin>
</Subbasins>
</MatData>
```

Výpis 3: Zkrácený obsah XML souboru reprezentující vybranou schematizaci povodí