M U N I

Popis softwarového výsledku

ANALYZA – Orchestrační a výpočetní subsystém

vytvořeného v rámci řešení projektu
Komplexní analýza a vizualizace heterogenních dat velkého rozsahu („ANALYZA“)

Období řešení projektu: 1. ledna 2017 – 31. prosince 2020

Výzkumný program: Program bezpečnostního výzkumu České republiky 2015-2020

Hlavní řešitel: RNDr. Tomáš Rebok, Ph.D.

Řešitelský tým:

RNDr. Tomáš Rebok, Ph.D. (*2017-2020*)
RNDr. Michal Batko, Ph.D. (*2017-2020*)
RNDr. Milan Čermák (*2017-2020*)
RNDr. Martin Drašar, Ph.D. (*2017-2020*)
Mgr. Miroslava Jarešová (*2017*)
doc. RNDr. Barbora Kozlíková, Ph.D. (*2017-2020)*
RNDr. Vladimír Míč, Ph.D. (*2017-2018*)
RNDr. Filip Nálepa, Ph.D. (*2017-2018*)
RNDr. David Novák, Ph.D. (*2017-2018*)
RNDr. Daniel Tovarňák, Ph.D. (*2017-2018*)
Mgr. Kristína Zákopčanová (*2017-2020*)
prof. Ing. Pavel Zezula, CSc. (*2017-2020*)
Mgr. Martin Macák (*2018*)
Mgr. Matúš Guoth (*2018*)
RNDr. Jakub Peschel (*2019-2020*)
RNDr. Katarína Furmanová, Ph.D. (*2019*)
Bc. Denis Kasanič (*2019-2020*)
Bc. Marko Řeháček (*2019-2020*)
Bc. Denisa Šrámková (*2019-2020*)
Matej Babej (*2019-2020*)
Bc. Dávid Brilla (*2019-2020*)
Bc. Martin Kažimír (*2019-2020*)
Erik Hricišák (*2019-2020*)
Bc. Jozef Bátrna (*2019-2020*)
Daniel Plakinger (*2020*)
Vladimír Lazárik (*2020*)
Kristián Gutič (*2020*)
Ing. Helena Mojdlová (*2017*)
Ing. Jitka Ročková (*2018-2020*)

V Brně, dne 30. ledna 2021

Masarykova univerzita

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

Obsah

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

1 Kontext SW komponenty „Orchestrační a výpočetní subsystém“ ve vyvinutém systému

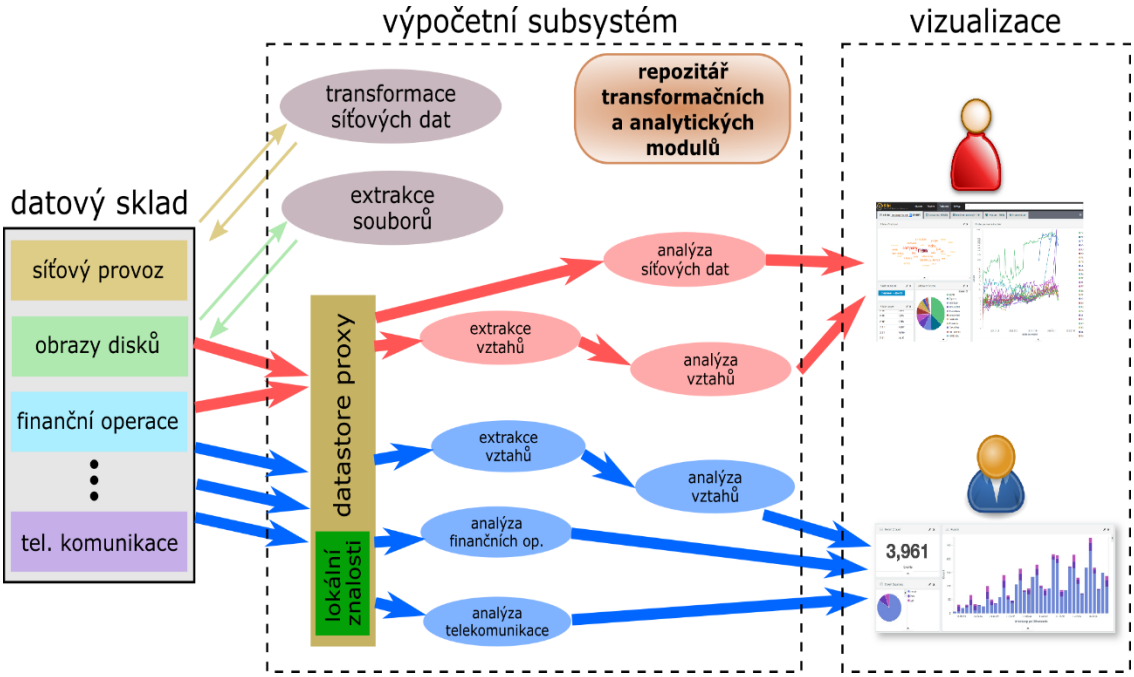
Hlavním cílem námi řešeného projektu byl návrh a realizace distribuovaného systému umožňujícího komplexní analýzu heterogenních dat velkého rozsahu. Navržený systém je koncipován jako nástroj umožňující jednotné uložení netriviálně velkých datových sad (např. záchytů síťového provozu, obrazů disků, komunikačních informací či znalostí z terénu), nad nimiž jsou vyvolávány nejrůznější mezidoménové analýzy (např. identifikace skupin, které jsou v kontaktu nezávisle na tom, jaký komunikační prostředek volí, případně identifikace po telekomunikační síti komunikujících stran, které si současně vyměňují netriviální finanční prostředky atp.). Takto navržená koncepce má za cíl prozkoumat možnosti a limity podobných mezidoménových analýz, jejichž ambicemi je omezit izolované analýzy různých datových domén a nezbytnost budování mezidoménového kontextu jen v režii analytika. Navržená koncepce vypomůže jak s budováním a udržováním tohoto kontextu přímo nad analyzovanými datovými sadami, tak i s odhalováním nových nebo složitě zaznamenatelných datových souvislostí a kontextů, které by jinak byly buď stěží nebo zcela vůbec odhalitelné.

Od tohoto cíle se odvíjely základní požadavky na vyvinutý systém, z nichž ty nejzásadnější lze v krátkosti sumarizovat následovně:

- *Škálovatelnost* – základní požadavek jakéhokoli systému, jehož ambicemi je práce s velkými daty či náročnými výpočty. Nejinak je tomu i v případě námi vyvinutého systému, kdy jak návrh celé architektury, tak i návrh a implementace dílčích komponent podporuje principy distribuovaného či paralelizovaného zpracování a práce s velkými daty.
- *Flexibilita* – neméně zásadní požadavek na systém, jehož ambicemi je práce s různými datovými doménami a různorodými analýzami. Námi vyvinutý systém je vyvinut s předpokladem, že nelze specifikovat konkrétní datové sady a informace, které má systém udržovat, ale naopak musí být připraven v budoucnu jednoduše integrovat různé, i nyní neznámé datové sady. Podobně i všechny analytické funkce systému nebylo možno specifikovat v době jeho vývoje, ale poskytnutím obecného frameworku s dostatečnou flexibilitou je umožněno jejich budoucí doplňování a adaptace novým potřebám.
- *Interaktivita* – požadavek úzce související se škálovatelností systému, jehož ambicí je poskytnout co nejinteraktivnější zpracování a reakce na požadavky uživatele – v závislosti na množství analyzovaných dat – v co nejkratším čase, což je nezbytně důležité zejména pro účely tzv. explorativních analýz (postupného odhalování znalostí).
- *Spolehlivost a důvěryhodnost* – neméně důležité požadavky, posilující důvěru uživatelů v odhalené skutečnosti a schopnosti systému.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

- *Bezpečnost* – jelikož tato netriviální oblast byla nad časový i personální rámec aktivit tohoto projektu, systém jsme vyvinuli tak, aby byl připraven pro dodatečnou implementaci rozsáhlých bezpečnostních mechanismů, včetně práce principů AAI (autentizace, autorizace, identity).

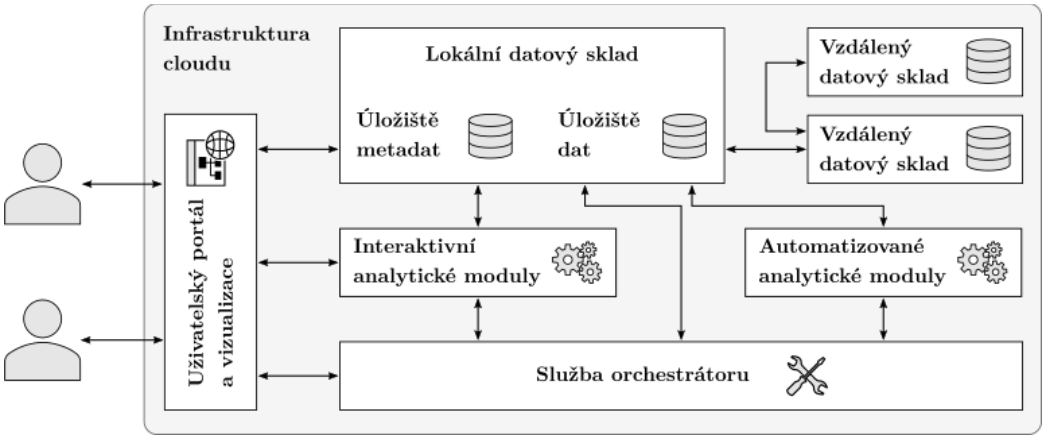


Obrázek 1: Ilustrace myšlenky sjednocených datových analýz s využitím analytických operací sestavených do datového workflow

Pro demonstraci této myšlenky sjednocených analýz (Obrázek 1) jsme navrhli a implementovali architekturu systému, jehož základní komponenty jsou ilustrovány na následujícím obrázku (Obrázek 2). Těmito základními komponentami jsou:

- ***Datový sklad*** v hierarchické sestavě distribuovaných datových úložišť udržujících veškerá analyzovaná data,
- ***Výpočetní subsystém***, poskytující sadu transformačních a analytických modulů a ***Orchestrační službu***, která tyto moduly vhodně skládá do datových workflow realizujících transformace dat v Datovém skladu či vlastní datové analýzy (z technického, programátorského a uživatelského pohledu jsou popsány v následujících kapitolách),
- ***Vizualizační komponentu*** poskytující analytické možnosti a prezentující výsledky jednotlivých analýz uživateli.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.



Obrázek 2: Architektura vyvinutého distribuovaného systému pro analýzu rozsáhlých heterogenních dat

Jádro celého implementovaného systému tvoří *analytické moduly*, které vhodně sestaveny do datového workflow (tzv. *analytické operace*) reprezentují analytické možnosti celého modulárního systému. Veškerá vstupní data – včetně poznatků, pozorování a výsledků provedených analýz – udržuje komponenta Datového skladu, ze které jsou potřebná data vyzvedávána analytickými moduly (a následně tamtéž ukládány zpracované mezivýsledky). Analytické moduly jsou ovládány Orchestrátorem, který zajišťuje jejich správné spuštění, včetně předání parametrů specifických pro daný modul. Současně zajišťuje řetězení jednotlivých modulů do analytických operací tak, aby se analytik mohl soustředit pouze na cílový analytický modul, který mu zpřístupní požadovaná data bez toho, aby je musel zpracovávat a připravovat odděleně. Různé analytické operace a moduly jsou spouštěny skrze Vizualizační komponentu, která umožnuje zachycení jednotlivých poznatků a případně výsledků analýz ve formě vztahového diagramu. Na základě výběru jednotlivých uzlů a hran daného grafu pak může analytik vybrat data, se kterými chce zacházet, a předat je Orchestrátoru a jednotlivým analytickým modulům. Jednotlivé komponenty analytického systému tvoří efektivně propojený celek tak, aby se plně využil potenciál pokročilých analýz a nových nástrojů pro zpracování dat spojených s policejním vyšetřováním.

Analytické operace jsou typicky složeny ze specializovaných automatizovaných a interaktivních analytických modulů. Automatizované moduly poskytují funkce transformace daných dat a různé formy detekce na základě předaných parametrů. Výsledky těchto modulů jsou uloženy v rámci Datového skladu, kde je možné s nimi dále pracovat ať už v navazujících analytických modulech nebo ve vztahové vizualizaci. Interaktivní analytické moduly zpřístupňují analytikovi daná data v rámci uživatelského rozhraní, kde je možné je analyzovat pomocí pokročilých funkcí daného analytického nástroje. S využitím pokročilých funkcí Orchestrátoru je možné analytické moduly spouštět samostatně nebo jako klastr distribuovaných výpočtů, pokud to daný modul podporuje.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

2 SW komponenta „Výpočetní a orchestrační subsystém“

Komponenta Výpočetního a orchestračního subsystému (podle své hlavní činnosti také označována jako „orchestrační služba“ nebo také „Orchestrátor“) slouží pro vlastní zpracování a analýzu vybraných dat. Komponenta zajišťuje dostupnost infrastruktury umožňující běh analytických modulů realizujících toto zpracování. Mezi její hlavní úkoly pak patří správa a řízení (tzv. orchestrace) této infrastruktury, a integrované propojení jednotlivých analytických částí vyvíjeného systému. Komponenta si udržuje přehled o všech běžících analytických projektech a analýzách v rámci nich realizovaných, včetně informací k nim přiřazeným, a připravuje tak prostředí pro import (tj. nahrávání z datového skladu) či výměnu dat a řídících informací mezi analytickými moduly. Analytické moduly jsou touto orchestrační službou spouštěny, po dobu běhu spravovány a v definovaných případech i ukončovány. Důležitou součástí této komponenty je i repositář analytických modulů, tj. služba, která udržuje informace o všech dostupných analytických modulech a dostupných analytických operacích, které lze s jejich pomocí realizovat, včetně identifikace konkrétních modulů, které mají být v případě požadavku na danou analytickou činnost vyvolány.

Jednotlivé analytické moduly vyvíjeného systému jsou samostatnými, jednoúčelovými miniaplikacemi, které jsou buď spuštěny po celou dobu běhu systému (například transformační moduly provádějící transformace dat nad datovým skladem) nebo jsou spouštěny na žádost podle vyžádaných analýz. Komponenta výpočetního subsystému a orchestrační služby tak musí zajistit zejména:

- spouštění potřebných modulů pro analýzy či transformace dat, včetně řešení závislostí mezi nimi;
- komunikaci mezi moduly a datovým skladem, mezi moduly vzájemně, a mezi moduly a vizualizační komponentou;
- udržování informací o jednotlivých analytických projektech a v rámci nich spuštěných analytických modulech.

Vyvinutou komponentu Orchestrátoru jsme tedy pro podporu této funkcionality navrhli zejména s ohledem na:

- *spolehlivost* – robustnost proti výpadku hardware;
- *bezpečnost* – vzájemná izolace modulů, bezpečná komunikace pouze skrze jimi definované programové rozhraní (API);
- *flexibilitu* – možnost spouštění modulů jak v datovém centru, tak případně i na lokálním výpočetním stroji;
- *škálovatelnost* – schopnost jednoduchého doplňování nových modulů do systému.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

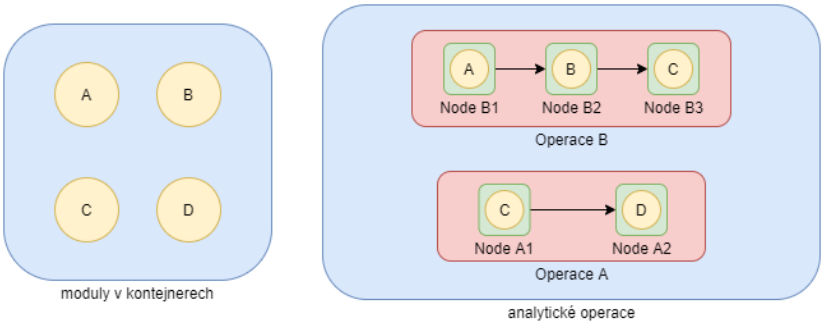
2.1 Architektura

Komponenta Výpočetního a orchestračního subsystému (podrobněji popsaná v ročních zprávách projektu a popisu výsledku projektu) sestává ze dvou vzájemně propojených a úzce spolupracujících služeb: vlastního výpočetního subsystému, sloužícímu pro běh analytických a transformačních modulů (uložených v centrálním repositáři modulů) a orchestračních služeb, spravujících celý jejich životní cyklus (od správného pořadí jejich spuštění, vlastní běh v průběhu zpracování dat až po jejich korektní ukončení). Vlastní analytické a transformační moduly jsou pak reprezentovány jako virtualizované kontejnery, poskytující vždy přesně definovanou a znovu použitelnou funkcionalitu. Jejich vhodným sestavením do tzv. analytických operací je pak dosaženo požadované analytické funkce, vyvolané datovým analytikem v uživatelském rozhraní systému, které je blíže popsáno v následující sekci. Obecně tak musí orchestrační služba zajistit dvě základní funkcionality: (1) spouštění potřebných modulů s řešením závislostí mezi nimi a (2) komunikaci mezi moduly, Datovým skladem a uživatelským rozhraním.

Analytický modul představuje jeden výpočetní program (izolovanou a přesně definovanou funkcionalitu), spouštěný v jediném (izolovaném) kontejneru. K provedení většiny, ve vyvinutém systému dostupných, analytických operací je však často i pro jedinou analytickou operaci potřeba spustit několik takových modulů, které až společně pokrývají požadovanou funkcionalitu. Jednotlivé analytické moduly nejsou striktně vázány k jediné analytické operaci, ale naopak mohou být využity a nezávisle spuštěny i ve zcela odlišných operacích, v jejichž zpracovávacím řetězci (datovém workflow) nachází své uplatnění.

Na úrovni výpočetního modelu jsou jednotlivé analytické moduly uzavřeny do takzvaných *uzlů* (*Node*). V těchto uzlech je možné – pro každé jednotlivé spuštění daného modulu v rámci konkrétní operace – upravovat definice vstupních parametrů modulu, a tím je pro konkrétní spouštěnou instanci vhodně konfigurovat. Cílem tohoto konceptu je právě snaha o poskytnutí žádoucí funkcionality, kdy nám tento přístup dovoluje využít jedinou definici analytického modulu ve více operacích. Ve speciálních případech je pak možné použít i již běžící kontejner analytického modulu, který je v případě identické konfigurace využit z jiné běžící analytické operace (což umožňuje efektivnější využívání dostupných výpočetních prostředků). Následující Obrázek 3 demonstruje hlavní myšlenku popsaného konceptu.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.



Obrázek 3: Ilustrace analytických operací a zapouzdření analytických modulů do uzlů (Node)

Požadavky kladené na funkcionalitu výpočetní komponenty a orchestračních služeb vyžadují sestavení distribuované infrastruktury, která umožní běh jednotlivých analytických modulů a jejich správu. Zde jsme využili řešení využívající virtualizační infrastrukturu, konkrétně řešení virtualizace na úrovni aplikací s minimalizovanými režijními nároky (technologie Docker[1] kontejnerů). Pro zajištění izolovanosti modulů, efektivního spouštění nebo zpracování datových workflow a spolehlivosti celého systému jsme nad tímto základním infrastrukturním řešením implementovali vlastní Orchestrátor, který s využitím nástroje Kubernetes[2] spravuje běh a komunikaci jednotlivých virtualizovaných kontejnerů.

2.2 Typy podporovaných modulů

Pro potřeby flexibilnějšího zpracovávaní operací orchestrační službou jsme definovali dva typy analytických modulů:

- *Job (automatizovaný)* – představuje jednu výpočetní úlohu, zpravidla krátce trvající, s definovaným koncem. Orchestrační služba čeká na ukončení tohoto modulu před spuštěním modulu navazujícího. Příkladem takového modulu je např. modul pro vytvoření indexu pro potřeby podobnostního vyhledávání.
- *Service (interaktivní)* – tento typ modulu slouží pro spuštění dlouhodobě běžící služby. Orchestrační služba po spuštění takového modulu nečeká na jeho ukončení, ale rovnou přistoupí k vykonávaní dalších úkonů. Takto spuštěná služba se pak ukončí výhradně na podnět ukončení celé běžící operace (resp. všech operací, které tento modul obsahují). Vhodným příkladem modulu tohoto typu je např. běh prostředí pro analýzu a vizualizace uložených dat v prostředí Elastic Stacku[3] (nástrojů ElasticSearch a Kibana).

[1] https://www.docker.com/
[2] https://kubernetes.io/
[3] https://www.elastic.co/

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

3 Instalační manuál a programátorská dokumentace

3.1 Obsah archívu

Přiložený archív obsahuje následující přílohy:

- *swagger.yaml* – dokumentace API vytvořená nástrojem Swagger
- *Dockerfile* – manifest pro vytvoření Docker obrazu Orchestrační služby
- *config/config.yaml* – konfigurační soubor Orchestrační služby
- *test* – adresář obsahující demonstrační příklady vstupních dat pro API volání
- *scripts* – adresář obsahující skripty pro nasazení Orchestrátoru

Ostatní přiložené adresáře představují zdrojový kód vyvinuté Orchestrační služby.

3.2 Návod pro nasazení Orchestrátoru

Teno návod popisuje nasazení Výpočetního a orchestračního subsystému na systému s operačním systémem Ubuntu 18.04 pro uživatele *ubuntu*.

Vlastní instalace Orchestrátoru sestává z následujících kroků:

1. Vytvoření databází
2. Vytvoření Docker repositáře
3. Vytvoření Kubernetes clusteru
4. Sestavení (*build*) Orchestrační služby
5. Spuštění Orchestrační služby

Vytvoření databází

Pro svůj běh vyžaduje Orchestrační služba 2 databáze:

- Databáze definic
- Databáze metadat

Tyto databáze je možno vytvořit v libovolném SQL systému (např. PostgreSQL, MySQL či Sqlite) – Orchestrační služba skrze izolaci volání umožňuje (pomocí definice vhodného konektoru) využít libovolný z nich. V tomto návodu si připravíme databázi typu PostgreSQL. Pro vytvoření obou zmíněných databází je možno využít přiložený skript *Postgresql.sh*.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

Instalace Docker subsystému

Pro možnost spouštění kontejnerů, které představují základní složku fungování Orchestrační služby, je zapotřebí nainstalovat Docker. Doporučený návod k instalaci Docker v operačním systému Ubuntu je k nalezení na následujícím odkazu: https://docs.docker.com/engine/install/ubuntu/

Vytvoření Docker repositáře

Pro podporu infrastruktury více systémů je potřeba vytvořit Docker repositář, který se bude používat pro ukládání všech Docker obrazů dodaných jednotlivými SW komponentami výsledků projektu. Popis vytvoření Docker repositáře je součástí oficiální dokumentace Docker, kde je popsán na následujícím odkazu: https://docs.docker.com/registry/

Vytvoření Kubernetes clusteru

Kubernetes cluster slouží jako výpočetní platforma pro Orchestrační službu, skrze kterou jsou spouštěny všechny analytické moduly, a která spravuje všechny kontejnery, komunikaci mezi nimi, a také komunikaci se specifickými kontejnery z Vizualizační komponenty. Pro svou vysokou komplexnost odkazujeme na podrobně popsaný oficiální dokumentační návod, který je dostupný na následujícím odkazu: https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/

Sestavení/instalace Orchestrační služby

Pro úspěšné sestavení Orchestrátoru je zapotřebí mít v systému nainstalovaný balík *golang* verze minimálně 1.14. V balíku je přiložen skript *golang.sh,* který vhodnou verzi balíku *golang* do systému automaticky stáhne a nainstaluje.

Sestavení Orchestrátoru je pak možno vykonat následujícím příkazem spuštěným v základním adresáři přiloženého archívu:

```
go build -o orchestrator-server ./server
```

Po úspěšném sestavení bude v adresáři vyroben spustitelný binární soubor s názvem *orchestrator-server.*

Spuštění Orchestrační služby

Pro úspěšný start služby je nezbytný vhodně vyplněný konfigurační soubor. Jeho příklad i s komentáři je přiložen v souboru *config/config.yaml*. Rovněž je nezbytný soubor obsahující přístup pro práci s Kubernetes a zastřešující příkaz *kubeconfig* (https://kubernetes.io/docs/concepts/configuration/organize-cluster-access-kubeconfig/).

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

Po adaptaci a přípravě konfiguračního souboru a *kubeconfig* je možno Orchestrátor spustit následujícím příkazem:

```
./orchestrator-server -c <cesta k config.yaml>
```

Podpůrné programy příkazové řádky

Pro jednodušší přípravu definic a testovaní obsahuje Orchestrační služba i následující podpůrné programy:

- *Definition Designer CLI* – správa definic uložených v databázi definic
  - sestavení: `go build -o orch-designer ./designer`
  - spuštění: `./orch-designer --help`
- *Orchestrator Client CLI* – program příkazové řádky pro testování komunikace s API Orchestrátoru
  - sestavení: `go build -o orch-client ./client`
  - spuštění: `./orch-client --help`

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

4 Uživatelská dokumentace

4.1 Analytické operace a moduly

Pojem analytická operace představuje proces provedení specifické analýzy nad objekty Datového skladu systému ANALYZA. Příkladem takové analýzy může být vyhledání obrázků s konkrétním vzorem nebo hledání komunit v síťových datech. K provedení analýz je obvykle potřeba provést několik kroků (příprava dat, výpočet, spuštění služby grafického prostředí pro shlédnutí výsledků analýzy, ...). Tyto jednotlivé kroky budeme nazývat analytické moduly, jejich sestavení do komplexního analytického zpracování pak jako analytické operace.

Modulem označujeme jeden kontejner (program, skript), který má obvykle jediný úkol. Bere vstupní parametry (ve formě proměnných prostředí, resp. *environment vars*), má přístup k objektům Datového skladu a svá vstupní data zpracovává v běžící instanci na Kubernetes clusteru, a následně výsledky ukládá zpět do Datového skladu. Pro lepší modularitu není modul svázaný s konkrétní analytickou operaci. Může se tedy použít v kterékoliv operaci.

Abychom byli schopni vytvořit operaci jako provedení kroků v definovaném pořadí, potřebujeme v ní definovat pravidla, která budou pořadí vynucovat. Operace se proto skládá z uzlů (kroky) a hran (pravidla). Operace je tedy směrový graf. Uzel obsahuje jméno definice modulu, který chceme v kroku použít. Hrana (A, B) pak definuje, že modul v uzlu B se provede až po úspěšném provedení modulu v uzlu A.

4.2 Příprava modulu

Příprava modulů sestává z následujících kroků:

1. Vytvoření obrazu (*image*) modulu
2. Nahrání vytvořeného obrazu do repositáře
3. Vytvoření definice modulu
4. Vložení definice modulu do databáze definic

4.2.1 Vytvoření obrazu modulu

Popis vytvoření Docker obrazu (image) je nejvhodněji popsán v oficiální Docker dokumentaci popisující tvorbu jeho předpisu (tzv. *Dockerfile*), dostupné na https://docs.docker.com/engine/reference/builder/ (tvůrce modulu se může inspirovat přiloženým předpisem *Orchestrator*). Platí však jedno pravidlo: všechny vstupní parametry definované pro modul budou při vytvoření kontejneru vloženy jako proměnné prostředí (tzv. *environment variables*). K nim Or-

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

chestrátor navíc vloží i další užitečné parametry, jako například odkaz na Datový sklad nebo adresy koncových bodů (tzv. *endpoints*) dalších modulů spuštěných pod stejnou operací. Tabulka s názvy a popisy těchto parametrů se nachází v sekci *Proměnné prostředí*.

4.2.2 Nahrání obrazu do repositáře modulů

Před pokračováním v této části potřebujeme mít připravený běžící Docker repositář, ze kterého může naše platforma (Kubernetes) stahovat požadované obrazy (image). Repositář běží pod svým DNS jménem použitým v certifikátu. V tomto návodu budeme jako adresu repositáře používat *"orchestrator:443"*.

Na našem stroji se nejprve přihlásíme na repositář:

```
docker login orchestrator:443
```

Nahrajeme obraz do lokální Docker cache:

```
docker load -i <image>.tar
```

Následně přejmenujeme nahraný obraz tak, aby obsahoval i cestu k repositáři:

```
docker tag <názov_image> orchestrator:443/<názov_image>:<tag>
```

Nyní jsme připraveni k nahrání obrazu do repositáře:

```
docker push orchestrator:443/<názov>:<tag>
```

4.3 Vytvoření definice modulu

Aby byl Orchestrátor schopen spustit požadovaný modul, musí mít k dispozici údaje tento modul popisující – například název obrazu představujícího daný modul a další jeho vlastnosti (možné vstupní parametry či použité porty). Všechny tyto údaje jsou obsaženy v jeho definici, která bude uložena v databázi definic.

Pro ukládání definic modulů se používá JSON formát, příklad souboru s definicí modulu ilustruje Obrázek 4.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

```
{
   "modules": [
      {
         "name": "modul-test-1",
         "type": "service",
         "description": "testing service",
         "image": "test_1",
         "tag": "1.0",
         "isBlocking": true,
         "healthPort": 6060,
         "ports": [
            {
               "name": "port1",
               "number": 8080,
               "isPublic": true
            }
         ],
         "parameters": [
            {
               "name": "param1",
               "defaultValue": "value1",
               "description": "param1 description",
               "isStrict": true
            }
         ],
         "resources": {
            "cpus": 1,
            "ram": 250
         }
      }
   ]
}
```


Obrázek 4: Příklad souboru s definicí analytického modulu ve formátu JSON

V definici modulu může jeho tvůrce použít následující parametry:

- `name` – název modulu (unikátní, povinný);
- `description` – popis modulu (volitelný);
- `type` – typ modulu, který může nabývat hodnoty *job* nebo *service*. Typ *job* představuje úlohu, která eventuálně někdy sama skončí a vyhodnotí se její úspěšné/neúspěšné ukončení. Typ *service* slouží pro spuštění dlouhodobějších procesů, které bývají ukončeny až manuálně uživatelem skrze některý z definovaných portů nebo zastavením celé operace.
- `image` – název obrazu (bez tagu) v Docker repositáři; Orchestrátor při vytváření kontejneru přidává prefix označující konkrétní repositář.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

- `tag` – tag obrazu;
- `isblocking` – udává, zda se má čekat na dokončení modulu nebo zda se po jeho úspěšném startu může rovnou přejít na vykonání dalších kroků v pořadí (výchozí hodnota je *false*);
- `ports` – síťové porty ve formě seznamu, na kterých modul poslouchá. Každý z uvedených portů sestává z:
  - `name` – unikátní název portu,
  - `number` – unikátní číslo portu,
  - `isPublic` – udává viditelnost portu. V případě hodnoty *false* je port dosažitelný pouze z ostatních modulů na Orchestrátoru. Pro otevření portu uživatelům je zapotřebí nastavit hodnotu na *true* (výchozí hodnota je *false*);
- `healthPort` – volitelně označuje číslo portu, na kterém Orchestrátor testuje živost modulu; modul nebude označený jako *running* dokud požadavek GET na "/" nevrátí na tomto portu stavový kód 200;
- `parameters` – seznam možných vstupních parametrů modulu. Definice každého takového parametru sestává z:
  - `name` – název parametru (unikátní pro daný analytický modul),
  - `defaultValue` – defaultní hodnota,
  - `description` – popis parametru (volitelný),
  - `isstrict` – pokud chceme, aby se tento parametr bral do úvahy při kontrole parametrů v případě možnosti znovupoužití již běžícího modulu, nastavujeme hodnotu *true* ;
- `resources` – požadavky na minimální množství výpočetních prostředků pro daný modul
  - `cpus` – počet požadovaných jader procesoru,
  - `ram` – požadované množství paměti RAM (v MiB) pro úspěšné vykonání modulu.

Vložení definice analytického modulu do databáze

Připravený JSON soubor popisující nový analytický modul je do databáze definic možno vložit s využitím připraveného a přiloženého nástroje *Definition Designer CLI*.

4.4 Příprava analytické operace

Příprava analytické operace sestává z následujících kroků:

1. Vytvoření definice operace
2. Vložení vytvořené definice do databáze definic

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

Vytvoření definice operace

Aby byl Orchestrátor schopen spustit požadovanou operaci, musí mít k dispozici její definici, která bude uložena v databázi definic. Pro ukládání definic operací se rovněž používá JSON formát. Příklad takto zapsané definice operace zobrazuje Obrázek 5.

```
{
   "operations": [
      {
         "name": "test-operation-1",
         "description": "operation description",
         "nodes": [
            {
               "id": 1,
               "description": "description of node",
               "moduleName": "m1",
               "parameters": [
                  {
                     "name": "param1",
                     "DefaultValue": "overriden",
                     "isConstant": true
                  }
               ]
            },
            {
               "id": 2,
               "moduleName": "m2"
            }
         ],
         "edges": [
            {
               "fromNode": 1,
               "toNode": 2
            }
         ]
      }
   ]
}
```

Obrázek 5: Příklad souboru s definicí analytické operace ve formátu JSON

V rámci definice analytické operace může její tvůrce využít následující parametry:

- `name` – unikátní název analytické operace;
- `description` – volný popis analytické operace;

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

- `nodes` – seznam uzlů operace (představujících zpracovávající body grafu)
  - `id` – ID uzlu, které je unikátní v rámci operace (používá se pro referenci v hranách);
  - `description` – popis uzlu;
  - `modulename` – jméno modulu, který se použije pro vykonání uzlu;
  - `parameters` – seznam parametrů, které chceme upravit pro kontext této analytické operace;
    - `name` – jméno parametru, který chceme upravit;
    - `defaultValue` – nová výchozí hodnota parametru modulu, která se použije při spuštění modulu v rámci tohoto uzlu;
    - `isConstant` – umožňuje pro tuto operaci nastavit parametr jako *read-only*, což znamená, že uživateli nebude dovoleno při volání této analytické operace tento parametr přepsat jinou hodnotou;
- `edges` – seznam pravidel popisujících pořadí vykonávaní uzlů (představuje hrany grafu)
  - `fromNode` – uzel, který má být spuštěný před uzlem uvedeným v *toNode;*
  - `toNode` – uzel, který má být spuštěný až po vykonání uzlu uvedeném jako *fromNode*.

Vložení definice do databáze

Připravený JSON soubor popisující novou analytickou operaci je do databáze definic možno vložit s využitím připraveného a přiloženého nástroje *Definition Designer CLI*.

4.5 Poznámky k definicím modulů

Definice v jediném souboru

Definice operací a modulů lze společně umístit i do jediného souboru. V tomto případě bude jejich JSON definice takového souboru vypadat následovně:

```
{   "operations": [...   ],   "modules": [... ]}
```


Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

Proměnné prostředí

Následující proměnné prostředí (tzv. *environment variables*) Orchestrační služba vkládá do každého jí spuštěného kontejneru:

Jméno proměnné	Popis
DATAWAREHOUSE	URL na Datový sklad
MODULE_ID_SELF	ID běžícího modulu, které je zároveň použité jako *hostname*
MODULE_NAME	Název definice běžícího modulu
PARAMETER_<MODULE_NAME>_<PARAM_NAME>	Hodnota parametru s názvem <PARAM_NAME> modulu s názvem definice <MODULE_NAME>
ENDPOINT_<MODULE_NAME>_<PORT_NAME>	Číslo portu uložené v databázi pod názvem <PORT_NAME>, které náleží běžícímu modulu s názvem své definice <MODULE_NAME>
MODULE_ID_<NAME>	ID jiného běžícího modulu v rámci operace, s názvem definice <NAME>, které zároveň slouží jako jeho DNS jméno

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

4.6 Komunikační rozhraní (API) Orchestrátoru

Požadovaná metoda	URL	Popis volání
GET	/config	Získání informací prostředí (IP adresa Datového skladu)
GET	/operation/{id}	Získání běžící operace (specifikovaného ID)
POST	/operation/	Spuštění nové operace
DELETE	/operation/{id}	Smazání běžící operace se specifikovaným ID
GET	/operation/	Získání všech běžících operací
GET	/definition/operation/	Získání všech definic operací, které je možné spustit
GET	/definition/operation/{názov}	Získání definice operace se specifikovaným názvem
GET	/definition/module/	Získání všech definicí možných modulů
GET	/definition/module/{name}	Získání definice modulu se specifikovaným názvem
GET	/module/	Získání všech běžících modulů
GET	/module/{id}	Získání běžícího modulu se specifikovaným ID
GET	/node/{operation_id}	Získání všech uzlů běžící operace se specifikovaným ID
GET	/definition/node/{operation}	Získání definice uzlu operace se specifikovaným názvem

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.

5 Poděkování

Vytvořeno v rámci projektu *„Komplexní analýza a vizualizace heterogenních dat velkého rozsahu (ANALYZA)“* (kód projektu VI20172020096, období řešení 1.1.2017 – 31.12.2020) v rámci Programu bezpečnostního výzkumu České republiky v letech 2015-2020 Ministerstva vnitra ČR.

Žerotínovo nám. 617/9, 601 77 Brno, Česká republika
T: +420 549 49 1111, E: info@muni.cz, www.muni.cz
Bankovní spojení: KB Brno-město, ČÚ: 85636621/0100, IČ: 00216224, DIČ: CZ00216224
V odpovědi prosím uvádějte naše číslo jednací.