



Posudek oponenta závěrečné práce

Oponent práce:	Ing. Jakub Jirůtka
Student:	Dominik Dosoudil
Název práce:	LightLog software pro zpracování logů aplikací
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Vytvořeno dne:	11. června 2021

Hodnotící kritéria

1. Splnění zadání

[1] zadání splněno

[2] zadání splněno s menšími výhradami

► [3] zadání splněno s většími výhradami

[4] zadání nesplněno

Cílem práce bylo navrhnout a implementovat software pro zpracování a následnou prezentaci logů aplikací běžících na linuxovém serveru. Tento cíl byl splněn, minimálně po stránce implementační.

Zadání stanovuje, aby byl software „navrhnout jako mikro služba“, tj. dle architektonického vzoru „microservice“. Tento vzor je definovaný dost volně, často se používá spíše jako tzv. „buzzword“. Očekával bych tedy, že student tento termín v textu definuje a popíše, jakým způsobem ho jeho řešení naplňuje. Není tomu tak. Pojem „mikro služba“ se v textu vyskytuje pouze jednou, a to ještě dost samoučelně, v kapitole č. 4 – Analýza konkurence. Navíc bych si dovilil tvrdit, že implementované řešení tomuto vzoru neodpovídá.

Zadání dále stanovuje šestibodový postup řešení:

1. Specifikujte požadavky na software – Práce neobsahuje nic, co by se dalo považovat za seriózní specifikaci a analýzu požadavků. V rámci první kapitoly (Cíl práce) je v 82 slovech uvedeno jen několik vlastností vyvinutého software. V celé práci není ani jasně specifikován model nasazení, tedy zda backend, frontend i databáze poběží na stejném serveru jako logující aplikace, na samostatném serveru určeném pro agregování logů z více aplikačních serverů nebo nějaká jiná varianta. Vůbec zde není řešeno zabezpečení komunikace mezi backendem a frontendem.

2. Na základě požadavků zvolte vhodné technologie pro implementaci. – Tímto bodem se zabývá kapitola pátá, Analýza technologií. Bez jasně stanovených požadavků lze jen stěží provést adekvátní analýzu technologií pro implementaci. Tato kapitola tedy spíše jen

uvádí zvolené technologie, povětšinou bez analýzy nebo adekvátní diskuze nad tím, z jakých důvodů byla daná technologie zvolena. Nicméně, většinu zvolených technologií považuji pro daný případ za rozumnou volbu. Zarážející je pro mne pouze nevysvětlená volba relačního databázového stroje PostgreSQL, která jde proti požadavkům na nízkou paměťovou náročnost. Očekával bych alespoň diskuzi nad použitím některé „embedded“ databáze, např. SQLite.

3. Proveďte softwarový návrh. – Něčemu, co by se možná dalo označit jako softwarový návrh, se blíží první sekce šesté kapitoly s názvem Realizace. Zde student stručně představuje vnitřní architekturu backendu založenou na návrhovém vzoru „Hexagon architecture“.

4. Implementujte prototyp aplikace. – Tento bod lze považovat za splněný.

5. Softwarové řešení podrobte vhodným testům, např. unit testy, zátěžové testy atd. – Backend obsahuje celkem 13 jednotkových testů, což není mnoho, frontend pak žádné. Přestože student v práci akcentuje procesorovou a paměťovou nenáročnost, a to až do takové míry, že se na devíti stránkách zabývá správou a alokací paměti v jazyku Rust, v celé práci nenajdeme ani náznak měření skutečné náročnosti výsledné implementace.

6. Prototyp aplikace nasadte do produkčního prostředí a zhodnoťte výsledky vaší práce. – V sedmé kapitole, Testování, student uvádí, že implementované řešení nasadili na jeden z produkčních serverů firmy, ve které pracuje. Nejsou však uvedeny žádné další detaily, žádné výsledky měření apod.

Zadání hodnotím jako průměrně náročné.

2. Písemná část práce

40 / 100 (F)

Úvod práce hovoří o potřebě přidávat k logovacím zprávám metadata a tvrdí, že textové logy toto neumožňují. V celé práci však není zmíněno, o jaké metadata se jedná.

První kapitolu (Cíl práce) čítající pouze 143 slov, považuji za informačně chudou.

Druhá kapitola (Motivace) by měla být spíše součástí Úvodu, nikoli jako samostatná kapitola.

Třetí kapitol (Logování obecně), která by měla představovat rešerši problémové domény, tvoří pouze jedna stránka textu a je naprosto nedostatečná. Obsahuje pouze informaci o existenci logovacích souborů, formátu syslog zpráv a prakticky zbytečné představení dvou notoricky známých unixových nástrojů pro práci s textovými soubory (grep a tail). Nezmiňuje dokonce ani systémové loggery nebo agregátory logů, které jsou právě tématem této práce! Logování a zpracování logů představuje poměrně rozsáhlé téma, o kterém by bylo možné napsat desítky, stovky i více stran. A minimálně deset stran by bylo potřeba k uvedení do tématu tak, aby bylo možné navázat solidní analýzou a návrhem řešení.

Čtvrtá kapitola (Analýza konkurence) představuje tři velké softwarové „stacky“ pro sběr, agregaci, indexaci, vyhledávání... logů celé infrastruktury. Zvolená „konkurence“ ve skutečnosti není tak úplně konkurencí, jde o řešení z jiné kategorie, pro jiné scénáře použití, než jakou je navržené řešení. Závěr analýzy tedy nemůže překvapit.

Pátá kapitola (Analýza technologií) je nejrozsáhlejší kapitolou této práce. Začíná poněkud samoúčelným srovnáním rychlosti a paměťové náročnosti třech programovacích jazyků: C++, Java a Rust. Následuje celých osm stran o správě a alokaci paměti v programovacích jazycích, převážně pak ve zvoleném jazyku Rust. V práci nenalezneme ani náznak analýzy či úvahy o tom, zda je navrhovaný software náročný na výpočetní výkon, jakou má paměťovou náročnost, jaký vliv na výkon má způsob alokace paměti či cokoli jiného, co by opodstatňovalo tuto analýzu.

Dále následuje definice termínu „API“ a prakticky nesouvisející představení metod protokolu HTTP. Systém NATS, na jehož integraci se práce měla zaměřit, je představen pouze ve 180 slovech, a to ještě velmi obecně. Přestože implementované řešení integruje protokol NATS, nikde se nedozvíme, jak tento protokol vypadá a na čem je postaven.

Šestá kapitola (Realizace) je v podstatě jen komentovaná procházka kódem; obsahuje příliš velké množství dlouhých ukázek kódu a popis spíše nepodstatných implementačních detailů.

V práci se vyskytují věcné chyby nebo nepřesnosti. Uvedu několik příkladů:

- V sekci 5.3 student uvádí, že „stack“ slouží pouze pro uchovávání „statických proměnných“, což je přinejmenším zavádějící.
- V podsekcích 5.7.1 a 6.2.4 se dočteme, že pro API existuje standard REST. REST ovšem není žádný standard, ale pouze architektonický styl.
- V sekci 5.11 student vysvětluje „asynchronní programování“ jako „typ paralelního programování, které nám umožňuje spustit více souběžných úloh na malém množství vláken“. U této definice je navíc reference na dokumentaci jazyku Rust, kde ovšem slovo „paralelní“ není vůbec použito. Jde o zásadní nepochopení asynchronního modelu v Rustu (a dalších jazycích) – nejde o paralelizaci, nýbrž kooperativní multitasking.

Strukturu textu, členění do kapitol, sekcí a podsekcí, považuji za velmi nepřehlednou a nelogickou. Zejména pak kapitolu 5, Analýza technologií.

Jazykovou stránku práce hodnotím spíše jako podprůměrnou. Textem se line kostrbatý a špatně čitelný slovosled, nekonzistentní (ne)skloňování některých termínů, chybný tvar předložek, chybný zápis zkratk a místy neformální formulace.

Po typografické stránce práci hodnotím jako standardní.

3. Nepísemná část, přílohy

75 / 100 (C)

Přílohu této práce tvoří implementace backend služby v jazyce Rust a frontend prohlížeče logů v jazyce TypeScript.

Kód obou implementací je přehledně strukturovaný, poměrně dobře čitelný a konzistentní. Nevšiml jsem v něm žádných zásadnějších prohřešků. Projekty správně využívají lintovací nástroje pro udržení konzistence kódu, backend využívá CI (Continuous Integration).

Backend vyžaduje pro sestavení tzv. „nightly release“ kompilátoru jazyka Rust, což není úplně standardní a v textu práce toto není zdůvodněné ani zmíněné.

Pro spuštění je nutné předem vytvořit PostgreSQL databázi a aplikovat databázové migrace – samotné SQL skripty jsou v projektu obsaženy (v adresáři ./sql), ale není zde ani skript pro jejich aplikaci, ani zmínka o tom, že je nutné je ručně nahrát.

Přestože je na mnoha místech práce akcentován požadavek na výkon, v databázovém schématu nenajdeme ani jeden index (vyjma primárního klíče). Metadata navíc ukládá do datového typu JSON namísto efektivnějšího typu JSONB.

Postup pro spuštění frontendu ve vývojovém módu nefunguje, neboť ve skriptech chybí příkaz pro kompilaci lokalizačních souborů. V práci není popsáno, jak se frontendu předává konfigurace v produkčním módu.

Zcela chybí dokumentace HTTP API, dokonce i formátu zpráv, a instalační manuál.

4. Hodnocení výsledků, jejich využitelnost

90/100 (A)

Výsledkem práce je funkční implementace služby pro agregaci, ukládání a prezentaci logů. Student vytvořil solidní základ, navrhl rozumnou vnitřní architekturu, která umožňuje další vývoj a rozšíření podporovaných protokolů i úložišť. Projekt bohužel zatím nedosahuje kvalit produktu, zejména vinou chybějící dokumentace, nicméně není od toho daleko.

Student v závěru práce uvádí, že implementovaná služba splnila požadavky firmy, ve které pracuje, a bude nasazena v reálném provozu. To je rozhodně pozitivní výsledek; zdaleka ne každá bakalářská práce se může tímto pochlubit.

Celkové hodnocení

62/100 (D)

S politováním musím konstatovat, že student neprokázal zvládnutí technik softwarového inženýrství ani schopnost napsat ucelený, dobře strukturovaný a hodnotný technický text.

Jedinou světlou stránku této práce představuje implementace, která je na dobré úrovni, pomínu-li absenci jakékoli provozní dokumentace. Velmi oceňuji, že implementovaná služba, dle slov studenta, splnila požadavky jeho zaměstnavatele a bude nasazena do reálného provozu.

Celkově tuto práci hodnotím jako velmi slabou, ale vzhledem k poslednímu bodu navrhuji známku D.

Otázky k obhajobě

1. Zvažovali jste pro agregaci logů z aplikací využití standardního loggeru typu syslog-ng, rsyslog apod? Porovnejte prosím výhody/nevýhody takového řešení oproti LightLog backendu.
2. Na základě čeho jste pro persistenci logů zvolil relační databázi? V práci zdůrazňujete požadavek na nízkou paměťovou náročnost, přesto jste pro persistenci zvolil řešení, které vyžaduje další běžící službu. Zvažoval jste použití nějaké „embedded“ databáze, např. SQLite?
3. Jak dalece jste strukturu práce a obsah kapitol konzultoval s Vaším vedoucím? Vyjadřoval se k nim před závěrečným odevzdáním práce?

Instrukce

Splnění zadání

Posudte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posudte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.

Písemná část práce

Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posudte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti.

Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře. Posudte správnost používání formálních zápisů obsažených v práci. Posudte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 26/2017, článek 3.

Posudte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.

Nepísemná část, přílohy

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů.

Hodnocení výsledků, jejich využitelnost

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

Celkové hodnocení

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.