Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce	Zdeněk Tomis
Název práce	Streamlining Usability of Enterprise Data Quality Management Tools for Data Engineers
Rok odevzdání	2024
Studijní program	Informatika
Specializace	Programování a vývoj software
Autor posudku	Lubomír Bulej — Vedoucí
Pracoviště	Katedra distribuovaných a spolehlivých systémů

K celé práci

	lepší	OK	horší	nevyhovuje
Obtížnost zadání		X		
Splnění zadání		X		
Rozsah práce *... textová i implementační část, zohlednění náročnosti*	X	X		

Práce řeší problém programové dostupnosti nástrojů pro správu kvality dat v kontextu řešení společnosti Ataccama. Specificky se jedná možnost spouštět centrálně definovaná pravidla a výrazy pro kontrolu a transformaci dat jak v serverovém prostředí (na platformě Java), tak lokálně v prostředí jazyka Python, se kterým typicky pracují datoví inženýři/analytici.

Řešení v podobě transpileru (z jazyka Ataccama Expression Language do jazyka Python) se sice dá v teoretické rovině považovat za dobře prozkoumaný problém, pro praktické nasazení za daných omezení je potřeba vyřešit řadu specifických problémů. Zároveň bylo nutné ověřit, zda je řešení prakticky použitelné pro typické situace.

Výstupem práce je funkční transpiler pokrytý velkým množstvím testů, včetně základního vyhodnocení jeho výkonnosti ve dvou scénářích a několika různými objemy dat. Rozsah implementace je zhruba 16K LOC (necelých 21K včetně komentářů a prázdných řádků) v jazyce Python, z čehož necelých 13K připadá na transpiler a 3K připadá na testy. V kontextu bakalářské práce to považuji za nadprůměrný rozsah. Jako celek práce splňuje zadání v plném rozsahu.

Textová část práce

	lepší	OK	horší	nevyhovuje
Formální úprava *... jazyková úroveň, typografická úroveň, citace*		X		
Struktura textu *... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu*		X		
Analýza		X		
Vývojová dokumentace		X	X	
Uživatelská dokumentace		X	X	

Až na výjimky text práce nezabíhá do velkých technických detailů a některé formulace jsou více marketingové než technické. V daném kontextu to nepovažuji za zásadní problém. Práce řeší konceptuálně dobře známý problém (transpiler), ovšem ve specifickém kontextu a za specifických omezení a zaměřuje se na to, zde je za daných omezení typické řešení použitelné a životaschopné. To se projevuje např. tím, že místo detailního rozboru konstrukce transpileru se spíše věnuje specifickým problémům souvisejících s vyhodnocováním výrazů na zásadně odlišných platformách (staticky typovaná Java vs dynamicky typovaný Python), protože to je pro uživatele a použitelnost řešení podstatné. Toto soustředění na uživatele bych tedy spíše vyzdvihl.

Očekával bych podrobnější vyhodnocení výkonnosti. To v současné podobě zahrnuje pouze dva scénaře (s různými objemy dat v každém z nich) a není úplně jasné, zda jsou dostatečně reprezentativní. Zároveň by mi přišlo vhodné zahrnout do vyhodnocení i výkonnostní testy s intepretrem Pythonu s podporou pro JIT kompilaci (PyPy). Na druhou stranu je nutno říct, že i v současné podobě jsou výkonnostní výsledky považovány za uspokojivé a v kontextu práce se tedy řešení ukazuje jako prakticky použitelné. K metodice měření výkonnosti nemám výhrady, jen v prezentaci výsledků bych doporučil vyhnout se grafům s logaritmickou škálou pokud to není (z povahy měřených dat) opravdu nutné.

Co se dokumentace týče, vývojovou dokumentaci považuji spíše za slabší z toho důvodu, že kód není komentovaný dostatečně podrobně na to, aby bylo možné z něj vygenerovat rozumnou referenční dokumentaci. Uživatelskou dokumentací zde rozumím dokumentaci, kterou by měl používat uživatel výsledného řešení, tedy např. datový inženýr. Rozhraní pro něj je sice velmi jednoduché, ale i tak by bylo vhodné, aby metody a třídy, se kterými přijde do styku (např. `create_compiler` a `ExpressionCompiler`), byly dokumentované alespoň pomocí dokumentačních komentářů v Pythonu.

Implementační část práce

	lepší	OK	horší	nevyhovuje
Kvalita návrhu *... architektura, struktury a algoritmy, použité technologie*		X		
Kvalita zpracování *... jmenné konvence, formátování, komentáře, testování*	X	X		
Stabilita implementace		X		

Projekt používá obvyklé nástroje a vývojové postupy. V prostředí jazyka Python používá nástroj Poetry pro správu závislostí ve virtuálním prostředí, framework Pytest pro jednotkové testy a nástroj mypy pro selektivní typovou kontrolu. Zdrojové texty jsou udržovány ve verzovacím systému a testy jsou spouštěny automaticky v rámci CI.

Kód je komentovaný spíše sporadicky, je však rozumně členěn do tříd, metod a funkcí a je tedy do značné míry samodokumentující. Vyzdvihnul bych množství testů (přes 200 testových metod a přes 1300 testovaných případů), které zajišťují, že výrazy budou vracet (v co největší možné míře) stejné výsledky jako na platformě Java.

Celkové hodnocení Výborně

Práci navrhuji na zvláštní ocenění Ne

Datum 20. června 2024 **Podpis**