Posudek diplomové práce

Matematicko-fyzikální fakulta Univerzity Karlovy

Autor práce	Jan Buchar
Název práce	Load Balancing in Evaluation Systems for Programming Assignments
Rok odevzdání	2020
Studijní program	Informatika **Studijní obor** Softwarové systémy

Autor posudku	Petr Tůma **Role** oponent
Pracoviště	D3S MFF UK

Text posudku:

Formálním cílem práce je „prozkoumat problém efektivního automatizovaného vyhodnocování úloh z programování na velkém počtu počítačů“. Jako významný aspekt problému je uveden algoritmus pro rozvrhování hodnocených úloh mezi dostupné počítače, jako další body zadání zmiňuje zhodnocení vlivu virtualizace na měření rychlosti a průzkum možností dynamické alokace počítačů podle zátěže systému.

Řešení práce obsahuje:
- přehled kritérií pro hodnocení úloh a požadavků na systém pro hodnocení úloh,
- přehled zvažovaných konfigurací systému sloužících pro bezpečné spouštění hodnocených úloh,
- sérii experimentů vyhodnocujících stabilitu měření času vybraných úloh ve vybraných konfiguracích systému,
- přehled algoritmů pro rozvrhování hodnocených úloh mezi dostupné počítače,
- sérii experimentů vyhodnocujících vybrané algoritmy,
- diskuzi možností využití kontejnerů, a
- diskuzi možností využití dynamické alokace počítačů.

Celkově práci vnímám jako spíše různorodou sbírku samostatných pozorování pouze volně navázaných na celkové cíle. Tento rys práce je sice částečně pochopitelný vzhledem k otevřenému charakteru zadání, současně ale výrazně omezuje praktický přínos práce. Domnívám se také, že práce je částečně obětí příliš široce pojatých experimentů – nejasné zaměření vede k testování mnoha kombinací konfigurací, jejichž vzájemné porovnání lze jen těžko prezentovat dostatečně kompaktně, kvůli celkovému objemu práce pak nezbývá čas na detailní analýzu zajímavých singulárních výsledků. Po technické stránce lze experimenty charakterizovat jako průměrné – hezká je celková automatizace experimentů, slabší jsou některé detaily jako deterministické místo randomizovaného pořadí měření, stejně jako vyhodnocení vlastních měření. Text práce je rozumně čitelný, je ale znát, že některým místům by prospěla větší formulační zručnost. Práce je převážně experimentálního charakteru, s malým objemem kódu hlavně prototypové a experimentální povahy, kterému se proto v detailu nevěnuji. V následujících poznámkách vysvětluji toto celkové hodnocení ve větším rozsahu.

Přestože zadání nabízí zdánlivě jasné a ucelené téma, řešení práce se mu ve skutečnosti nevěnuje jako celku. Místo toho zadání pouze zastřešuje relativně nezávislé výsledky, kterými jsou zejména dvě poměrně rozsáhlé sady experimentů, a vedle nich pak rozsahem spíše menší příspěvky v dalších bodech práce. Tím trpí zejména celková struktura práce – ačkoliv hlavním cílem by mělo být vyhodnocování úloh na velkém počtu počítačů, práce se vůbec nevěnuje tomu, jak velký systém si má čtenář představit co do počtu a rozmanitosti úloh, uživatelů či počítačů, a jak tyto parametry ovlivňují směr experimentů.

Vedle celkového pohledu na zadání postrádám také dostatečný nadhled u zdůvodnění obou hlavních experimentů. Čekal bych, že vyhodnocení stability měření bude předcházet úvaha o tom, jak stabilní měření jsou vlastně pro hodnocení úloh potřeba, případně zda čas jako relativně málo stabilní a málo přenositelná metrika nemůže být nahrazen či doplněn jinými metrikami, například počtem iterací nebo počtem I/O operací – nakonec při hodnocení úloh může jít často o to změřit reálnou složitost spíše než jen čas.

Podobně experimenty s rozvrhováním by mělo předcházet vysvětlení, v jakém smyslu a v jakých situacích má rozvrhování zlepšit chování systému. Nezdá se, že by systém stále pracoval pod takovou zátěží, aby měl libovolný work conserving algoritmus co rozvrhovat – a pokud ano, tak by bylo vhodné nejprve blíže popsat parametry takové zátěže a ty parametry chování systému, které by se měly pod zátěží zlepšit. Také by bylo vhodné zvážit další zjevné metody úspory výpočetního výkonu, například oddělené plánování překladu a měření.

Absence jasných cílů je patrná i v detailech a vyhodnocení experimentů. Ačkoliv práce letmo zmiňuje například úlohy z paralelního programování CPU i GPU nebo úlohy externího třídění, pro vyhodnocení stability jsou místo

reálných úloh použity jednoduché programy, které vesměs používají mnohem méně rozmanitou paletu funkcí hardware. Zdůvodnění tohoto kroku je velmi stručné, jeho dopady na relevanci výsledků přitom mohou být značné. Podobně jsou použity pouze striktně symetrické konfigurace obsazení procesorů, kde měřené úlohy běží pokud možno co nejvíce současně, přitom v reálném systému je taková koordinace spíše málo pravděpodobná a zatížení by se dynamicky měnilo.

Vyhodnocení experimentů se typicky omezuje na konstatování pozorovaných efektů, které nejsou dále analyzovány – to se týká například outlierů na obrázku 2.5, nepřesného měření wall clock time ze sekce 2.4.1, nebo virtualizovaných běhů rychlejších než běhů na dedikovaném hardware z obrázku 2.8. Některé jiné efekty, například překvapivě rychlý běh v některých konfiguracích s hardwarovými thready na obrázku 2.12, jsou ponechány bez komentáře.

Práce sama také konstatuje, že ani použití hardware performance event counters nevede k vysvětlení pozorovaných efektů. Zde bylo možná na místě inspirovat se některými existujícím postupy pro analýzu výkonu pomocí hardware counters, například podle článku 10.1109/ISPASS.2014.6844459.

Vhodnější postupy by zřejmě byly k dispozici i pro statistické vyhodnocení výsledků – porovnávání intervalů spolehlivosti průměru a směrodatné odchylky nemá jasnou interpretaci a maskuje velikost pozorovaných efektů, místo něj bylo možné například určit rovnou intervaly spolehlivosti pro pozorované rozdíly nebo poměry průměrů.

Popis experimentů s algoritmy rozvrhování se věnuje řadě slepých uliček – například není jasné, proč se nejprve v sekci 3.1.2 věnuje prostor detailnímu vysvětlení komplikací spojených s preempcí měření, v podstatě se závěrem, že preempce je technicky příliš komplikovaná, ale pak se přesto v sekcích 3.3.2 a 3.5.2 diskutují algoritmy vyžadující preempci, které sekce 3.6 hned v úvodu opět kvůli potřebě preempce zavrhne.

I v těchto experimentech lze najít zvláštní detaily – například časy úloh se aproximují normální distribucí, přestože jsou k dispozici empirické distribuce z reálního provozu systému, přítom normální distribuce mimo jiné může generovat (a dodané skripty také generují) záporné časy úloh.

Celkově velmi mnoho parametrů experimentů je vybráno ad hoc. Bizarně působí vysvětlení pro experiment s více typy úloh, které v jedné větě nejprve tvrdí, že cílem experimentu je lépe napodobit reálné použití systému, ale hned ve větě následující dodává, že četnosti úloh byly vybrány bez jakéhokoliv vztahu k reálným. To opět nabádá k otázce, nakolik jsou vykonané experimety relevantní.

Z lepších aspektů experimentu lze vyzdvihnout použití historických dat k predikci času úloh při plánování. Za zmínku možná stojí otázka, zda tento mechanismus nemůže poškozovat studenty, kteří začnou s pomalejším řešením úloh – to totiž systém použije pro odhad doby běhu příštích variant řešení a u doporučeného algoritmu shortest job first pak bude hodnocení těchto variant spíše odkládat.

Části práce věnované kontejnerům a dynamické alokaci počitačů jsou velmi malého rozsahu (včetně dodaného kódu) a celkově odtržené od zbytku práce. To je problém zejména u dynamické alokace, protože ta musí hledat vhodný kompromis mezi náklady na provoz, stabilitou měření a odezvou systému.

Práci doporučuji k obhajobě.

Práci nenavrhuji na zvláštní ocenění.

V Praze dne 16.6.2020

Podpis: