

Posudek bakalářské práce „On the Hardness of General Caching“ předložené Lukášem Folwarczným

Bakalářská práce pojednává o problémech z oblasti cachování souborů. Jedním z dlouho otevřených problémů v této oblasti byla otázka, zda je obecné cachování v offline případě silně NP-těžké. V nedávném článku bylo dokázáno, že tomu tak skutečně je; přesto byl tento výsledek poněkud neuspokojivý, protože důkaz vyžadoval instance používající stránky větší, než je polovina velikosti cache. Takovéto instance příliš neodpovídají skutečným vstupům. Přírozenou navazující otázkou tedy bylo, zda je offline obecné cachování NP-těžké i v případě, že jsou povoleny pouze instance s „malými“ stránkami. Hlavním přínosem práce je kladné zodpovězení právě této otázky. Dále práce přináší jednodušší důkazy již dříve známých tvrzení a několik možných aplikací.

Práce začíná úvodem, který popisuje motivaci zkoumaných problémů a poskytuje přehled práce. První kapitola je nejdelsí a také obsahuje největší přínos práce, totiž výše zmíněný důkaz NP-těžkosti obecného cachování už při omezení na stránky velikostí 1, 2 a 3. Dále je tento výsledek ještě zobecněn pro tzv. *bit model* a *fault model* a obě obvyklá pravidla načítání do cache, *optional* i *forced*. Druhá kapitola se zabývá tzv. *work functions* a *layers*, které jsou klíčovými nástroji v návrhu cachovacích algoritmů, a podává nový, jednodušší, důkaz o charakterizaci work funkce pomocí layers. Ve třetí kapitole je tento důkaz zobecněn pro cachování s proměnlivou velikostí cache; jedním z výsledků třetí kapitoly jsou např. Věty 3.1 a 3.8 o funkčnosti tzv. *Bela-dyho pravidla* pro cachování s proměnlivou velikostí. Čtvrtá kapitola předkládá dva jednoduché algoritmy pro speciální instance obecného cachování, které jsou založeny na výsledcích z předchozí kapitoly; první algoritmus řeší efektivně instance s „málo“ požadavky na abnormální (tzn. velikosti jiné než 1) stránky, druhý takové instance, v nichž jsou abnormální stránky velikostí blízké velikosti celé cache. V závěru práce jsou předloženy relevantní otevřené problémy.

Hodnocení

Obsah

Jak již bylo zmíněno, první kapitola předložené práce obsahuje netriviální nový výsledek, navíc rozpracovaný v poměrně velké šíři. Přínos druhé a třetí kapitoly nespočívá v nových výsledcích, ale zjednodušení známých důkazů, což také hodnotím kladně. Algoritmy obsažené ve čtvrté kapitole jsou též zajímavé a slouží jako demonstrace nástrojů popsaných v předchozích kapitolách. Formulace všech tvrzení a též struktura jejich důkazů jsou na vysoké úrovni. Na základě toho všeho musím konstatovat, že přínos práce je značný.

Forma

Domnívám se však, že po formální stránce by šla práce podstatně zlepšit. Obecně práce nenabízí čtenáři moc nadhledu a intuice. V první kapitole není tento nedostatek markantní a místy práce i takovéto komentáře obsahuje (např. str. 13 a komentář k významu hodnot δ_B a γ_B^c). Příkladem

místa, které by si dle mého názoru zasloužilo více vysvětlení, je popis posloupnosti požadavků na stranách 8 a 9; tento popis je technicky úplný, ale neposkytuje čtenáři žádná vodítka k pochopení otázky *proč* je posloupnost navržena právě tímto způsobem, či proč je zvoleno právě toto značení atd.

V tomto ohledu by mohla být práce zlepšena například pojmenováním nejdůležitějších tvrzení názvy vypovídajícími o jejich významu, nebo lze častěji předcházet tvrzení a důkazy jejich neformálním popisem. Zároveň si jsem vědom, že do značné míry je užitečnost těchto změn otázkou osobní preference.

Dalším významným nedostatkem formy (trochu jiného druhu) je téměř chybějící zasazení kapitol 2-4 do širšího kontextu. Pokud kapitoly 2 a 3 obsahují důkazy existujících tvrzení, nejsou tato tvrzení (a místa, kde byla původně dokázána) nijak vyznačena. Jako jiný příklad použiji úvod do kapitoly 3 (o cache proměnlivé velikosti) – z práce není zřejmé, zda je rozšíření konceptu *layers* na proměnlivou velikost zcela novou myšlenkou této práce, nebo zda vyplývá ze známých výsledků, jen zatím nebylo takto sepsáno, nebo zda jej autor uvádí jen pro úplnost. Obdobně např. podkapitola 3.3 a v ní Věta 3.8 – z práce není zřejmé, zda byla již dříve známa funkčnost Beladyho pravidla při *optional* načítacím pravidle. Taktéž v kapitole 4 by bylo užitečné popsat kontext – byla již studována složitost cachování vzhledem k jinému parametru, než je velikost vstupu? Pokud ne, jsou tyto výsledky zajímavým prvním krokem. Pokud ano, je potřeba dále zodpovědět například proč autor zvolil zrovna tyto parametry (jsou přirozené vzhledem k reálným vstupům? atd.).

Do jisté míry je tento nedostatek pochopitelný. Práce se dotýká kombinací offline/online scénáře, s pravidlem *optional* nebo *forced*, v modelech *fault* a *bit*, s proměnlivou velikostí cache či bez ní – všechny studované varianty jsou legitimní a složitost je tedy v samotném oboru cachování. Větší přehlednosti práce v tomto ohledu by snad mohlo jít dosáhnout s pomocí několika tabulek shrnujících existující a nové výsledky.

Jazyk

Práce je psána anglicky, a to na vysoké úrovni. Pokud by měl být text dále používán, doporučuji jej ještě jednou projít a soustředit se na používání anglických členů (někdy je používán určitý člen „the“, ačkoliv není myšlen žádný konkrétní objekt). Jazykově problematických míst (gramatické chyby či překlepy) ale práce obsahuje opravdu málo.

Otázky

Osobně mne zaujala čtvrtá kapitola práce o algoritmech pro speciální instance cachování. Takovýto přístup totiž přesně zapadá do optiky parametrizované složitosti, již se sám zabývám. Typickým postupem je vydělení nějaké vlastnosti instance jako parametru (v případě práce je to buď počet požadavků na abnormální stránky, nebo rozdíl velikosti cache a nejmenší abnormální stránky) a následně analýza složitosti vzhledem k tomuto parametru a velikosti vstupu. Obvykle se předpokládá, že parametr bude malý (např. logaritmický) nebo konstantní. Existence algoritmu se složitostí $f(k)poly(n)$ (kde k je parametr a n je velikost vstupu) znamená, že problém přísluší do třídy FPT; algoritmus se složitostí $n^{f(k)}$ pak implikuje příslušnost do třídy XP.

Věřím, že by autora mohly zajímat následující otázky:

- Jaké varianty cachování již byly v parametrizované složitosti zkoumány? Konkrétně jaké jsou jiné (přirozené) parametry cachování?
- Algoritmus 2 (podkapitola 4.2) má složitost $n^{O(\kappa)}$, tedy spadá do třídy XP. Lze pro tento parametr získat FPT algoritmus?

- Zajímavý algoritmus by mohlo jít získat aplikací Lenstrova algoritmu pro řešení smíšených lineárních programů (MILP) ve fixní celočíselné dimenzi. Podobný postup používá u jiného druhu problému Bansal et al. [Approximating Vector Scheduling: Almost Matching Upper and Lower Bounds, LATIN 2014]. Tento postup by byl vhodný, pokud lze sestrotit lineární program obsahující „málo“ celočíselných proměnných (pro abnormální stránky) a „mnoho“ spojitéch proměnných (pro normální stránky), z něhož lze rekonstruovat (část) řešení.

Závěr

Předkládaná práce více než splňuje požadavky kladené na bakalářskou práci. Přináší důležitý nový výsledek v oboru cachování, zjednodušuje existující výsledky a nabízí příklad aplikací. Tím rozhodně demonstruje, že pan Folwarczný se v oboru orientuje, ovládá klíčové nástroje a dovede své výsledky srozumitelně zaznamenat v písemné formě. Dalším přirozeným krokem je přitažlivě prezentovat výsledky širší vědecké komunitě – v tom vnímám prostor pro zlepšení. Přesto rád doporučuji tuto práci uznat jako práci bakalářskou.

V Praze, 20. srpna 2015