Posudek diplomové práce

předložené na Matematicko-fyzikální fakultě
Univerzity Karlovy v Praze

Název práce:

Resource limiting and accounting facility for FreeBSD

Autor práce:

Rudolf Tomori

Posudek oponenta:

Cílem diplomové práce pana Tomoriho bylo prozkoumat přístupy používané ve vybraných operačních systémech pro evidenci spotřebovaných systémových prostředků a pro přidělování jejich částí procesům nebo jejich skupinám. Na základě analýzy bylo cílem navrhnout a implementovat podporu pro omezení procesorového času a přenosové kapacity disků v operačním systému FreeBSD.

Výsledné řešení představují dva patche do jádra FreeBSD, z nichž jeden již byl přijat do FreeBSD-CURRENT. Rozsah kódu v práci je poměrně malý, oba patche mají dohromady cca 1000 řádků, což je z části vyváženo složitostí prostředí jádra operačního systému, pro který bylo řešení implementováno. Dalším faktorem, který by měl vyvážit malý rozsah kódu práce je analýza podobných subsystémů v jiných operačních systémech, kde je opět nevyhnutelně nutné studovat netriviální množství kódu jádra.

Předložené řešení je funkční, ale z práce samotné není možné si udělat představu o tom, jak dobré/kvalitní řešení je. Vyhodnocení se omezuje na triviální testy (překlad FreeBSD ze zdrojových textů, čtení/zápis sekvenčního bloku dat na diskové zařízení pomocí utility `dd`), které postrádají znaky seriozního experimentu a statistického vyhodnocení. Z těchto testů je možné vidět, že omezení prostředků *nějak* funguje, ale nebudí dojem kvality:

- V případě CPU omezeného na 60% jsou vidět časté překmity k plnému vytížení (práce neuvádí, v jakém procentu času tento jev nastával). Zde bych očekával, že pro vyhodnocení částečného přidělování CPU budou použit benchmark, který je silně vázaný na procesor, nikoliv kompilace, která zatěžuje CPU velmi nehomogenně. Zároveň bych očekával, že autor otestuje, jak dobře a jak rychle (ve smyslu reakce na změnu zatížení generovaného procesem) jeho řešení reguluje přidělovaný čas pro různé úrovně zatížení a rychlosti změny zatížení pro jeden a více více procesů.
- V případě přenosové kapacity disku omezené na 1MiB/s (nezávisle pro čtení a zápis) systém poskytoval vyšší rychlosti, než je bylo požadováno (do 4% u čtení, kolem 16% u zápisu). Autor zvýšenou rychlost u čtení přičítá faktu, že utilita `dd` považuje data za zapsaná dříve, než jsou zapsána na disk, což jako vysvětlení nepovažuji za uspokojivé. Bez ohledu na to bych však opět očekával vyhodnocení pro různé limity při zatížení jedním a více procesy a detailnější vyhodnocení přesnosti regulace.
- Dodatek C doplňuje sekce 6.2 a 6.3 o testy, ve kterých jsou limity na CPU a přenosovou kapacitu disku vztaženy na uživatele místo na proces, ale celkově ze zavedeného schematu jednoduchých testů nevybočuje.

Text práce je anglicky, což považuji za plus i přes drobné stylistické prohřešky. Textová část práce však trpí většími problémy, které bych se nyní pokusil shrnout:

- Přestože zadání zmiňuje analýzu subsystémů pro částečné přidělování prostředků v různých systémech, autor se zabývá pouze linuxovými *cgroups*. Toto omezení pravděpodobně vzniklo po dohodě s vedoucím a není problém ho pochopit. Co však považuji za závažnější je fakt, že část, věnovaná *cgroups* není vůbec propojena se zbytkem práce v tom smyslu, že nikde není vidět, že by nějak ovlivnila design výsledného řešení. To je zarážející ze dvou důvodů: a) analýza *cgroups* je velmi detailní, jde na úroveň datových struktur v linuxovém kernelu, jejich vzájemného propojení, algoritmů pro iteraci přes členy různých skupin apod., a b) celkově zabírá v podstatě polovinu textové části, ale nakonec nic neovlivní, takže není vůbec jasné, k čemu byla potřeba.
- V kontrastu s podrobnou analýzou *cgroups* je v podstatě triviální popis FreeBSD subsystému

racct, v jehož rámci autor nakonec řešení implementoval.

- V analýze problému se autor zabývá vesměs implementačními záležitostmi, přitom např. přidělování CPU má řadu zajímavých aspektů, které by bylo dobré v analýze zvážit a případně kontrastovat s tím, co dělají současné systémy. Ať už se jedná o mechanizmy regulace, kde je možné (úspěšně) aplikovat postupy z teorie regulace, nebo zjišťování aktuálního zatížení, jeho přesnost a režie na jeho výpočet, frekvenci regulačních zásahů a jejich vliv na přesnost regulace. Ani řešení se tímto nakonec příliš nezabývá a bere jako fakt, že perioda vyhodnocování využití prostředků v *racct* je 1 sekunda. Autor píše, že např. omezení na CPU je možné implementovat v user-space, což je pravda, ale jsou s tím spojené také určité problémy spojené se zjišťováním aktuální zátěže, režií a přesností regulace např. malých zátěží. To samozřejmě nelze zjistit bez implementace, ale i tak bych očekával, že se autor v analýze bude věnovat také teoretičtějším aspektům řešení.
- Jak části věnované analýze, tak části věnované implementaci jsou velmi detailní, což by samo o sobě nevadilo, kdyby jim předcházely části, které se více zabývají koncepty. V případě implementace by se jednalo o kapitolu, která pojednává o celkovém návrhu, zvolených regulačních mechanizmech, sémantice přidělování prostředků více procesům na více procesorech. V případě analýzy by naopak stačil pouze pohled z vyšší úrovně, bez přehršle implementačních detailů, které nakonec nebyly k ničemu použity.

Celkově práce splňuje cíle vytčené zadáním, ale pouze ty explicitní. S ohledem na stručnost zadání bych očekával, že solidní experimentální vyhodnocení bude jakýmsi implicitním cílem, protože u tohoto typu práce je to jediný způsob, jak zhodnotit kvalitu řešení. Text práce by měl lépe propojit analýzu s návrhem a implementací a přidat popis na konceptuální (ne pouze implementační) úrovni.

Závěrem, I přes uvedené výhrady, konstatuji, že autor prokázal schopnost řešit netriviální problém v náročném prostředí jádra OS, a proto ji **doporučuji** k obhajobě.

V Praze, 21.05.2013 Lubomír Bulej