Аб выкарыстанні мабільных платформаў у добраахвотных вылічэннях

Каваленка У.Ю., Касцюк Д.А., Чацвёркіна Г.А.

Брэсцкі дзяржаўны тэхнічны універсітэт, volodik666@gmail.com

An overview of approaches to use mobile devices in volunteer computing related to free/libre projects is presented. Analysis of mobile platform aplicability is done from the point of view of processing power and typical use cases, with brief discussion of possible directions to develop mobile-oriented GRID-like compiting.

«Добраахвотныя вылічэнні» (ангельск. volunteer computing) гэта выкарыстанне асабістых рэсурсаў карыстальнікаў Internet нейкай GRID-сеткай (SETI@home, праекты distributed.net і інш.). Сёння карыстальніцкія GRID дэманструюць вылічальную магутнасць, параўнальную з топавымі суперкамп'ютэрамі, ахопліваюць задачы матэматыкі, крыптаграфіі, біялогіі, медыцыны, фізікі і інш. Праекты завабліваюць удзельнікаў важнасцю вырашаемай задачы, упорам на пазіцыі ў рэйтынгах, электроннымі бэджамі, прывабнымі кліенцкімі праграмамі (напрыклад, экраннай застаўкай, якая падвышае сацыяльны статус ўладальніка кампутара).

У апошні час з'яўляюцца мабільныя кліенты volunteer computing, у першую чаргу для Android - дзякуючы як Linux-аснове, так і асаблівасцям самой платформы, якая змушае канцэнтраваць ў мабільнай прыладзе шмат'ядравы працэсар, які мэтанакіравана прастойвае вялікую частку часу. Кліенты ствараюцца альбо мэтавым партаваннем (Android-кліент BOINC), альбо зборкай Linux-версіі: напрыклад, першакрасавіцкі жарт аб вылічальным кластары Google на Nexus One прывёў да з'яўлення некалькіх рэальных аналагаў [1].

Аднак ўключэнне ў GRID мабільных прылад, пры ўсёй іх шматлікасці і шмат'ядравасці, абмяжоўваюць асаблівасці архітэктуры і іх выкарыстання. Архітэктура ARM, якая ўзнікла ў 80-я гады для персанальных кампутараў пад кіраваннем RiscOS, у 90-я гады пераарыентавалася на наладонные кампутары дзякуючы прастаце, эфектыўнаму выкарыстанню блокаў працэсара і меншаму, у параўнанні з x86, энергаспажыванню і цеплавыдзяленню. Далейшая аптымізацыя ARM праходзіла з упорам на гэтыя два крытэрыі, і сёння карыстальнікі ведаюць, што «ў ARM і x86 розныя мегагерцы». Праілюстраваць гэтую розніцу можна наглядна, калі параўнаць працягласць выканання тыповых каманд у тактах працэсара (мал. 1-а, светлы фон адпавядае больш высокай прадукцыйнасці). Паказчыкі сямейства ARM знізяцца яшчэ больш, калі ўлічыць меншыя тактавыя частоты і разраднасць некаторых інструкцый, і сітуацыю не выправіць нават ўлік вялікага ліку ядраў. Безумоўна, калі дадаткова пранармавать прадукцыйнасць па рассейванай магутнасці (мал. 2-б), ARM апынецца безумоўным лідэрам, але гэтыя мела б значэнне, калі б нізкаватныя GRID-сеткі былі хоць колькі-

небудзь запатрабаваныя.

	Data transfer	Logical	Integer +/	Integer *	SSE/NEON(VFP)	Special math
ARM Cortex A9	2,66	1	4,16	5,66	10,5	
ARM Cortex A7	3,66	1	5,16	6,66	16,5	
x86 Atom 330	2,66	1	4	19	7	
x86 AMD A10 (Trinity)	2,33	1	3,5	5	5,5	113,33
x86 I7 3xxx	2	1	3,5	3	3	73,33

а)

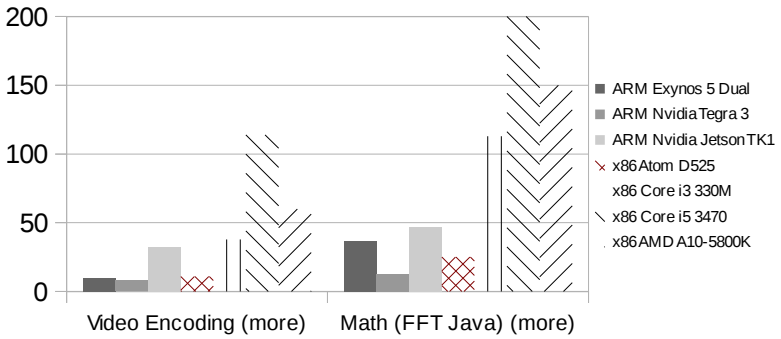
	Data transfer	Logical	Integer +/	Integer *	SSE/NEON(VFP)	Special math
ARM Cortex A9	98 932	263 158	63 259	46 494	25 063	
ARM Cortex A7	136 612	500 000	96 899	75 075	30 303	
x86 Atom 330	46 992	125 000	31 250	6 579	17 857	
x86 AMD A10 (Trinity)	4 292	10 000	2 857	2 000	1 818	88
x86 I7 3xxx	3 846	7 692	2 198	2 564	2 564	105

б)

Мал. 1. Працягласць выканання інструкцый у цыклах працэсара (а) і вынік нармавання па рассейванай магутнасці (б)

Варта згадаць, што рэальны разрыў паміж ARM і x86 яшчэ больш з-за аптымізацыі выканання патокаў каманд ў x86 (спекулятыўнае выкананне, прадказанне пераходаў, зліццё і падзел інструкцый і г.д.), але гэта дрэнна паддаецца разлікам, і таму прадукцыйнасць даводзіцца ацэньваць на канкрэтных задачах, якія ўлічваюць, праўда, яшчэ і розніцу ў хуткадзейнасці памяці і дыскавай падсістэмы, як, напрыклад, бэнчмаркі, якія перыядычна публікуюцца рэсурсам phoronix.com (мал. 2).

Працоўны працэс мабільнай прылады, з другога боку, мяркуе як мага больш доўгае функцыянаванне ад батарэі, і як вынік - мінімальнае выкарыстанне наяўных вылічальных магутнасцяў. Распрацоўнікі Andoid-кліентаў гэта ўлічваюць: так, кліент BOINC працуе толькі падчас зарадкі і пазбягае перадачы дадзеных праз мабільную сетку.



Мал. 2. Параўнанне архітэктур на рэальных задачах (сямейства x86 адзначана штрыхоўкай; найбольш высокія значэнні адпавядаюць большай прадукцыйнасці)

Такім чынам, апроч меншых вылічальных магутнасцяў, мабільны вузел

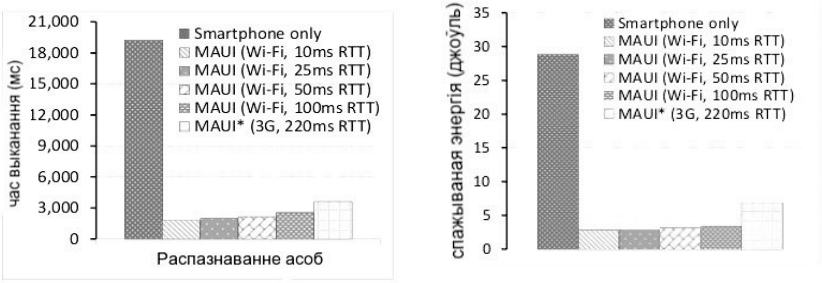
хутчэй за ўсё будзе заставацца ў валанцёрскай GRID-сетцы толькі пакуль сетка практычна не нагружае яго вылічэннямі, і таму прымяненне мабільных прылад у складзе GRID носіць эпізадычны характар.

Найбольш падыходныя для мабільных вылічэнняў задачы павінны альбо выкарыстаць вылічальную магутнасць ў выглядзе кароткачасовых пікавых нагрузак (1), альбо здабываць карысць не столькі з вылічальнага рэсурсу, колькі з тэрытарыяльнага роскіду (2) і гетэрагеннасці вузлоў сеткі (3).

Да задач катэгорыі 1 ставіцца самадыягностыка GRID з мэтай вымярэння дасягальных пікавых нагрузак; аднак гэта ператварае пабудову сістэмы у спартовае мерапрыемства. Задачы тыпу 2 перыядычна абмяркоўваюцца ў прэсе (ужыванне смартфонаў для стварэння размеркаванай сеткі датчыкаў). На жаль, рэальныя прылады не маюць датчыкаў, сетка якіх апынулася б карыснай у рэальных даследаваннях або маніторынгу.

Ёсць прыклады «ня матэматычных» валанцёрскіх вылічэнняў. Так, праект Majestic-12 выкарыстоўвае інтэрнэт-канал ўладальніка прылады для пабудовы размеркаванай сістэмы сеткавага пошуку, праект Surveill@Home ацэньвае прадукцыйнасць і адмоваўстойлівасць вэб-сайтаў, а адзін з чатырох метадаў узаемадзеяння, закладзеных у праекце SmartLab, тычыцца выдаленай адладкі праграм на мабільнай прыладзе [1].

Больш перспектыўна ў мабільных добраахвотных вылічэннях зваротная схема - напрыклад, апісаныя ў [3] і [4] спробы дэлегавання частцы нагрузкі мабільнай маламагутнай прылады выдаленаму серверу для памяншэння энергаспажывання і павышэння прадукцыйнасці ў некаторых тыпах задач. Прыклады канкрэтных абласцей – гульні, складаная матэматыка з багаццем спецыяльных функцый, або задачы, якія цяжка дзеляцца на паралельныя патокі і ў якіх прадукцыйнасць аднаго ядра вельмі крытычна. Мал. 3 паказвае графік прадукцыйнасці пры распазнаванні асоб сіламі толькі прылады і пры выкарыстанні сервера, а мал. 4 - параўнанне спажыванай энергіі пры розных сцэнарах [3].

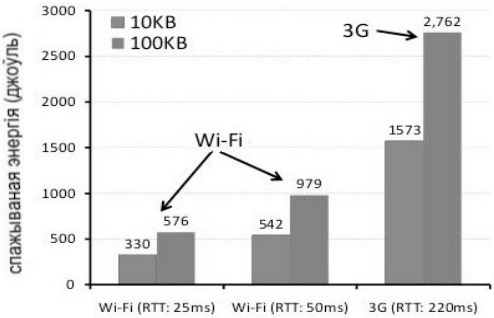


а) б)

Мал. 4. Час працы алгарытму (а) і спажыванне энергіі (б) на мабільнай прыладзе пры розных затрымках сеткі (а)

Галоўныя праблемы у дадзеным выпадку - методыка «падзелу» задачы на кавалкі іншым вузлам сеткі і эфектыўная перадача дадзеных па сетцы. Агульнае ва ўсіх прапанаваных рашэннях - запуск праграмы і на мабільнай прыладзе і на высокапрадукцыйным кампутары, з кіраваннем паслядоўнасцю апрацоўкі. На жаль, для кожнага тыпу задач трэба рэалізаваць асобную методыку. Для задач рэндэрынгу, напрыклад, у тэставых мэтах прымяняўся падыход, калі кожны кадр аблічвалі на маламагутнай прыладзе але з нізкай колькасцю дэталяў, а на высокапрадукцыйнай ў поўным дазволе і з усімі дэталямі. Па сетцы жа перадавалася розніца паміж кадрамі і выраблялася сумяшчэнне. Можна адзначыць і зваротны падыход, калі мабільная прылада аблічвае адзін кадр з поўнай якасцю, а кампутар у гэты час паспявае апрацаваць n кадраў.

Праца модуляў бесправадной сувязі гэтак жа звязана са значным выдаткам энергіі (мал. 5). Аднак з распаўсюджваннем высакахуткасных WiFi-сетак гэта перастае з'яўляцца нераадольнай праблемай.



Мал. 5. Спажыванне энергіі пры рознай якасці сеткі

Літаратура

1. G. Larkou et all. Managing Smartphone Testbeds with SmartLab // Proceedings of the 27th USENIX Large Installation System Administration Conference (LISA '13), Washington D.C., USA. Pp. 115-132. https://www.usenix.org/conference/lisa13/technical-sessions/presentation/larkou

2. C. Hu, I. Neamtiu. Automating GUI Testing for Android Applications / 3rd Int. conf. on Advanced Science and Technology, September 27--29, 2011, Seoul, Korea. - pp. 77--83. http://www.cs.ucr.edu/~neamtiu/pubs/ast11hu.pdf

3. E. Cuervo et all. MAUI: Making Smartphones Last Longer with Code Offload / / MobiSys'10, June 15–18, 2010, San Francisco, California, USA. pp. 49-62. http://research.microsoft.com/en-us/um/people/ranveer/docs/maui.pdf

4. E. Cuervo et all. Kahawai: High-Quality Mobile Gaming Using GPU Offload // MobiSys'15, May 18–22, 2015, Florence, Italy. pp. 121-135. http://homes.cs.washington.edu/~kklebeck/kahawai.pdf