



Електротехнички факултет, Универзитет у Београду

Катедра за рачунарску технику и информатику

Пројекат из предмета

Перформансе рачунарских система

професор: Јелица Протић
асистент: Немања Којић

студент: Милан Бранковић
119/07

јул 2011.

Садржај

Опис проблема	3
Аналитичко решење проблема	5
Објашњење симулације	7
Одступање параметара система	8

Опис проблема

Мултипрограмски рачунар има два процесора, два системска диска и K корисничких дискова. Процесорска обрада траје у просеку $S_p = 3\text{ms}$, просечно време опслуживања за системске дискове износи по $S_{d1} = 10\text{ms}$, $S_{d2} = 12\text{ms}$, а за корисничке дискове по $S_{dk} = 20\text{ms}$. Сва времена имају експоненцијалну расподелу. После процесорске обраде у 10% случајева захтева се приступ првом системском диску, у 15% случајева другом, а у 75% случајева приступа се неком од K корисничких дискова (са једнаком вероватноћом за сваки диск). После приступа неком системском диску у 20% случајева не врши се никаква обрада већ се процеси поново враћају у процесорски ред. У преосталих 80% случајева приступа се неком од K корисничких дискова (са једнаком вероватноћом за сваки диск). После приступа корисничком диску у 95% случајева процес се враћа у процесорски ред, а у преосталих 5% случајева процес иде на први системски диск. Процесори имају јединствен ред за чекање и подједнако су оптерећени.

За затворену мрежу којом се моделира овај рачунарски систем, потребно је одредити искоришћења ресурса, протоке кроз ресурсе, просечан број послова у сваком од ресурса овог система и време одзива система (просечно трајање једног циклуса, од када процес дође у процесорски ред, док не дође у тај ред следећи пут) за K од 2 до 6. Ови параметри се одређују за степен мултипрограмирања од 10, 15 и 25. Одредити критични ресурс у систему.

Задаци

- 1) Потребно је написати програм који ће за улазни параметар програма n (степен мултипрограмирања):
 - а) Симулирати дати систем и на основу резултата симулације одредити и приказати (уписати у први излазни фајл) све тражене параметре система за свако K (број корисничких дискова). Потребно је обезбедити могућност подешавања симулираног времена рада система (у минутима), а подразумевано време симулације је 12h (не реалног времена рада програма, већ симулираног времена рада система!)
 - б) Бјузеновом методом аналитички одредити све тражене параметре и уписати их у други излазни фајл. Анализу спровести за свако K (број корисничких дискова)
 - с) Одредити релативно одступање (у процентима) резултата симулационе методе од аналитичке. Ова одступања је потребно израчунати за сваки број корисничких дискова K и уписати их у засебан (трећи) излазни фајл
- 2) Потребно је написати документацију која детаљно објашњава метод симулације и аналитичко решавање проблема. У документацији приложити извештаје

релативног одступања тражених параметара система, као и вредности датих параметара у виду хистограма (хистограм по параметру) који приказују вредности ових параметара у зависности од броја корисничких дискова за следеће вредности параметра **n**: 10, 15, 20 послова

Аналитичко решење проблема

Gordon-Newell – ове једначине за дати систем:

$$-(1-p_{11})\mu_1x_1 + p_{21}\mu_2x_2 + \dots + p_{k1}\mu_kx_k = 0$$

$$p_{12}\mu_1x_1 - (1-p_{22})\mu_2x_2 + \dots + p_{k2}\mu_kx_k = 0$$

⋮

⋮

$$p_{1k}\mu_1x_1 + p_{2k}\mu_2x_2 + \dots - (1-p_{kk})\mu_kx_k = 0$$

где су

- k - број сервера
- $\mu_1, \mu_2, \dots, \mu_k$ - брзине појединачних сервера
- $p_{11}, p_{22}, \dots, p_{kk}$ - вероватноћа да се после посете првом опслужном центру врати поново на њега
- p_{ij} - вероватноћа да се после посете i -том опслужном центру посао упути j -том опслужном центру ($i = 1, 2, \dots, k; j = 1, 2, \dots, k$)
- x_i - нормализована потражња за i -тим сервером

За следеће вредности

$$\mu_1 = \mu_2 = \frac{1}{s_p} = \frac{1}{3} \quad \mu_3 = \frac{1}{s_{d1}} = \frac{1}{10}$$

$$\mu_4 = \frac{1}{s_{d2}} = \frac{1}{12} \quad \mu_{k+3} = \frac{1}{s_{dk}} = \frac{1}{20}$$

и вероватноће

$$p_{13} = p_{23} = 0,1 \quad p_{14} = p_{24} = 0,15 \quad p_{k3} = 0,05$$

$$p_{31} = p_{32} = p_{41} = p_{42} = \frac{0,2}{2} \quad p_{3k} = p_{4k} = \frac{0,8}{k}$$

$$p_{1k} = p_{2k} = \frac{0,75}{k} \quad p_{k1} = p_{k2} = \frac{0,95}{2}$$

остале вероватноће су 0, и за $K=2..6$ добијамо следећа решења

$$x_1 = x_2 = x_p = 1$$

$$x_3 = x_{s1} = 0.9965278$$

$$x_4 = x_{s2} = 1.2$$

$$x_k = x_{ku} = 13.1944444/K$$

Одређивање $G(n)$ Вузен-овом методом

$i \backslash j$	x_1	...	x_{j-1}	x_j	...	x_k
0	1		1	1		1
1	x_1		$x_1 + \dots + x_{j-1}$	$x_1 + \dots + x_j$		$g(1,k) = G(1)$
...						
$i-1$	x_1^{i-1}		$g(i-1, j-1)$	$g(i-1, j)$		$g(i-1,k) = G(i-1)$
i	x_1^i		$g(i, j-1)$	$g(i, j)$		$g(i,k) = G(i)$
...						
n	x_1^n		$g(n, j-1)$	$g(n, j)$		$g(n,k) = G(n)$

где је $g(i, j) = g(i, j-1) + x_j g(i-1, j)$

За ове добијене вредности, даља анализа се извршава на следећи начин:

- Формира се матрица која у својој последњој колони даје вредност $G(n)$
- Израчунају се остали параметри који се траже у задатку
- Добијене вредности се упишу у излазни фајл

Објашњење симулације

Сваки ресурс представљен је помоћу класе `Resource` која представља појединачни ресурс у систему. Сваком ресурсу придружен је један ред за чекање. Ови елементи су логички повезани тако да моделирају задати систем.

Позивом методе *simulate* симулација се извршава на следећи начин:

- Најпре се иницијализује систем тако да се сви процеси расподеле у процесорске редове за чекање
- Формира се ред заузећа ресурса који је сортиран по вредностима времена у којем ресурс завршава своју обраду
- Узима се ресурс који је на почетку реда заузетих ресурса и системско време се постави на време тог ресурса, његов посао се прослеђује следећем ресурсу, узима се следећи процес из реда за чекање (уколико такав постоји)
- Све ово се ради док не истекне време симулације
- Када време симулације истекне, израчунају се тражени параметри и упишу се у излазни фајл

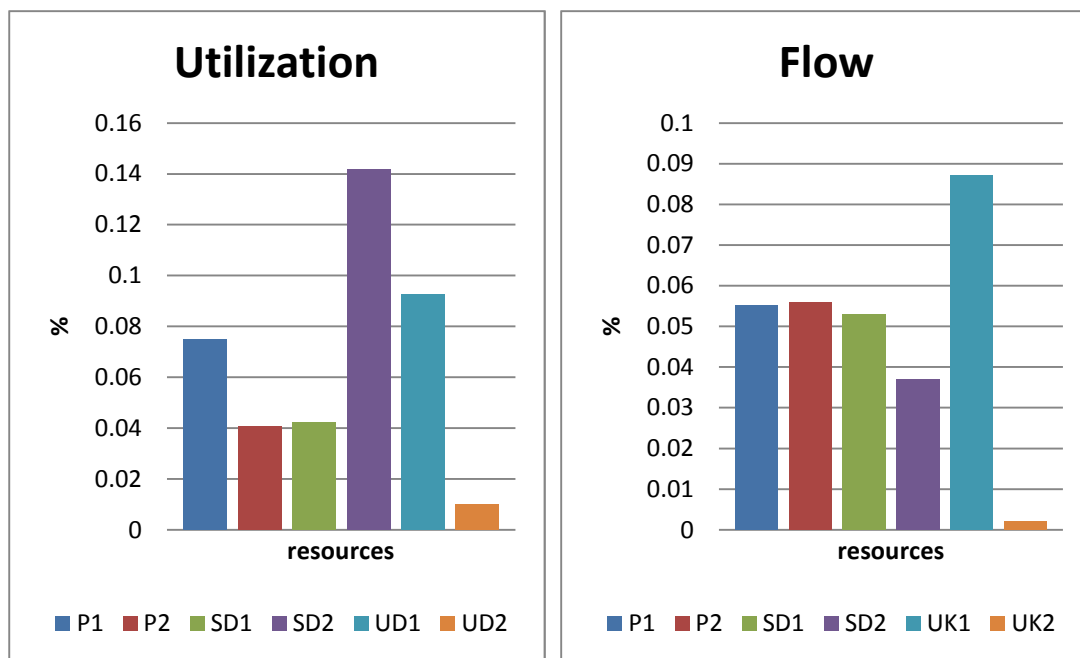
Одступање параметара система

Након извршене анализе и симулације датог система, позивом методе *deviation* у трећи излазни фајл ће бити уписане вредности релативног одступања аналитичке и симулационе методе за тражене параметре.

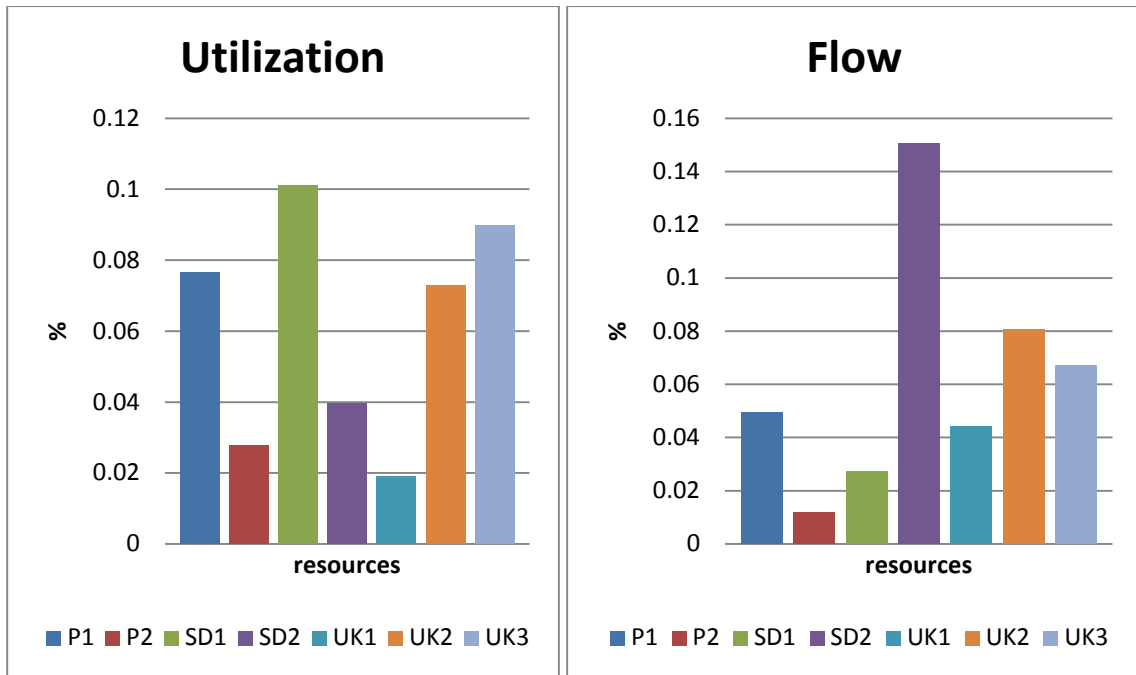
На основу добијених резултата, можемо утврдити да симулација само потврђује оно што смо аналитичком путем добили. Посматрајући резултате можемо утврдити да је релативно одступање мање од 1%.

Као пример одступања наводи се графичка представа за степен мултипрограмирања $N = 10$.

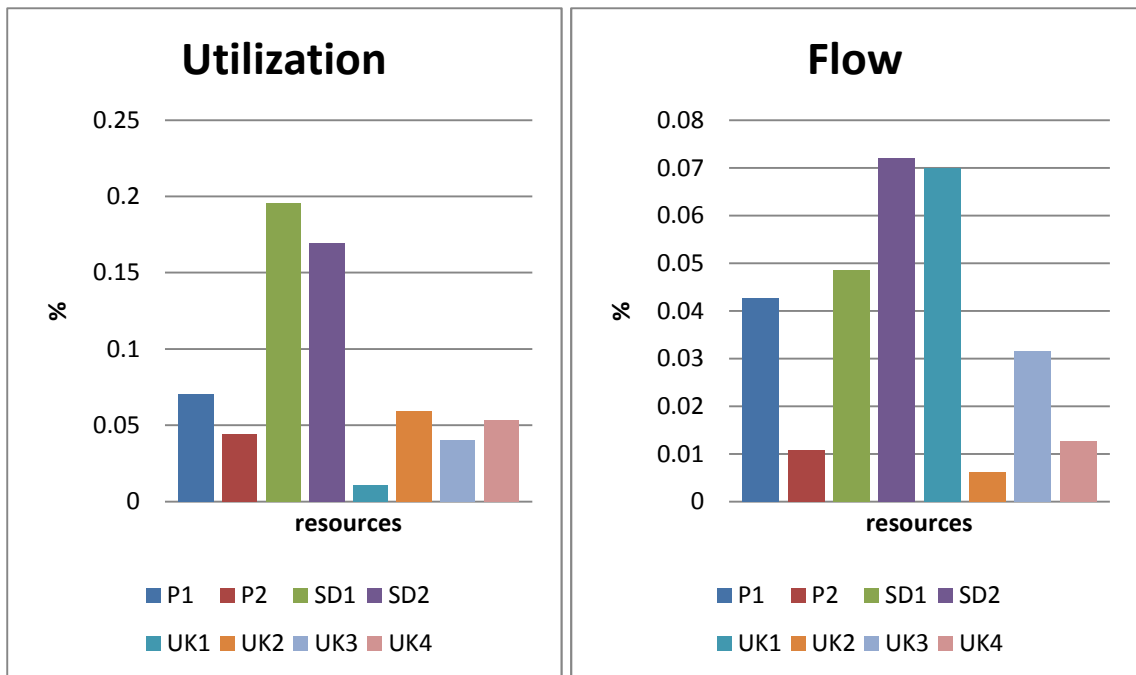
$$K = 2$$



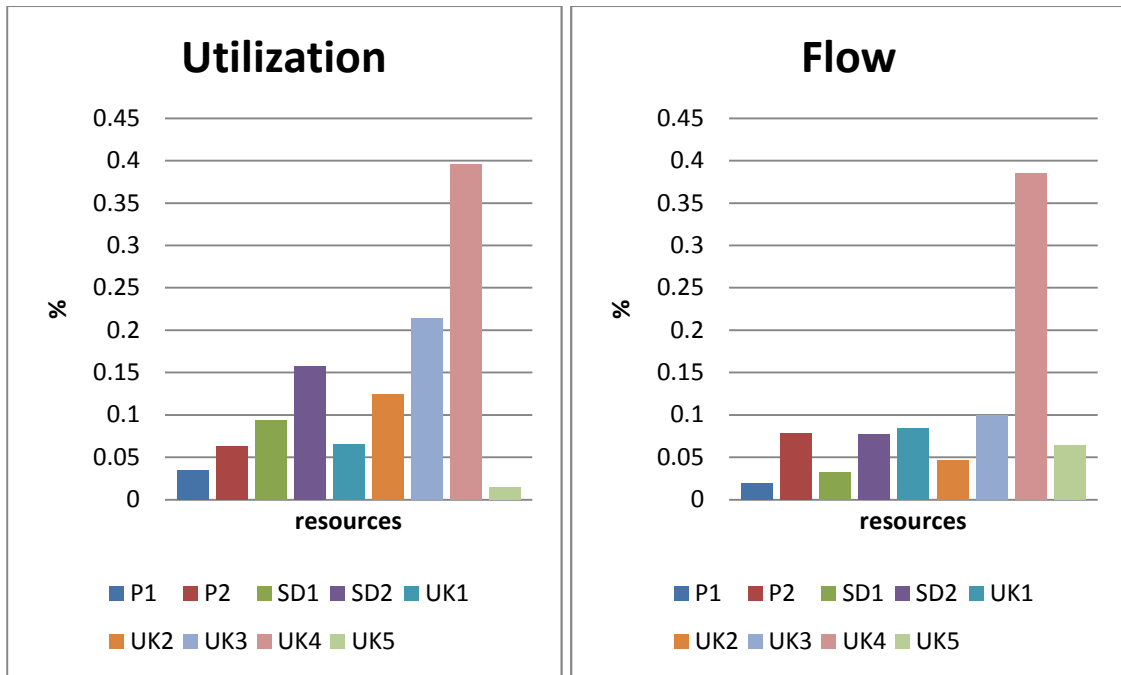
K = 3



K = 4



K = 5



K = 6

