

ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԶԳԱՅԻՆ  
ԱԿԱԴԵՄԻԱՅԻ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈՔԼԵՄՆԵՐԻ  
ԻՆՍՏԻՏՈՒՏ

Պետրոսյան Սերգեյ Հրանտի

ՀԵՐԹԻ ԴԻՆԱՄԻԿ ԿԱԶՄԱԿԵՐՊՈՒՄԸ ԳՐԻԴ ՄԻՋԱՎԱՅՐՈՒՄ

Ե.13.04 «Հաշվողական մեքենաների, համալիրների, համակարգերի և ցանցերի  
մաթեմատիկական և ծրագրային ապահովում» մասնագիտությամբ  
տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման  
ատենախոսության

ՍԵՂՄԱԳԻՐ

Երևան 2012

---

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ  
НАЦИОНАЛЬНОЙ АКАДЕМИИ НАУК РЕСПУБЛИКИ АРМЕНИЯ

Петросян Сергей Грантович

ДИНАМИЧЕСКОЕ УПРАВЛЕНИЕ ОЧЕРЕДЯМИ В СРЕДЕ ГРИД

АВТОРЕФЕРАТ

Диссертации на соискание ученой степени кандидата технических наук по  
специальности 05.13.04 – “ Математическое и программное обеспечение  
математических машин, комплексов, систем и сетей ”

Ереван 2012

Ատենախոսության թեման հաստատվել է Հայաստանի Հանրապետության  
Գիտությունների ազգային ակադեմիայի Ինֆորմատիկայի և ավտոմատացման  
պրոբլեմների ինստիտուտում

Գիտական ղեկավար՝	Ֆ. Գ. թ.	Վ. Գ. Սահակյան
Պաշտոնական ընդդիմախոսներ՝	տ. գ. ղ.	Հ. Հ. Հարությունյան
	տ. գ. թ.	Մ. Ղ. Գյուրջյան

Առաջատար կազմակերպություն՝ Երևանի մաթեմատիկական մեքենաների  
գիտահետազոտական ինստիտուտ

Ատենախոսության պաշտպանությունը կայանալու է 2012թ. Դեկտեմբերի 26-ին, ժամը 16<sup>00</sup> -ին, ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի 037 «Ինֆորմատիկա և հաշվողական համակարգեր» մասնագիտական խորհրդի նիստում, հետևյալ հասցեով 0014, Երևան, Պ. Սևակի փ. 1:

Ատենախոսությանը կարելի է ծանոթանալ ինստիտուտի գրադարանում:  
Սեղմագիրն առաքված է 2012թ. նոյեմբերի 26-ին:

Մասնագիտական խորհրդի գիտական  
Քարտուղար, ֆիզ. մաթ. գիտ. դոկտոր



Հ. Գ. Սարգսյան

---

Тема диссертации утверждена в институте проблем информатики и автоматизации  
НАН РА

Научный руководитель:

к.ф.м.н.

В. Г. Саакян

Официальные оппоненты:

д.т.н.

Г. А. Арутюнян

к.т.н.

М. К. Гюрджян

Ведущая организация:

Ереванский научно-исследовательский институт  
математических наук

Защита диссертации состоится 26 декабря 2012г. в 16<sup>00</sup> на заседании специализированного совета 037 "Информатика и вычислительные системы" Института проблем информатики и автоматизации НАН РА по адресу: 0014, Ереван, ул. П. Севака, 1.

С диссертацией можно ознакомиться в библиотеке ИПИА НАН РА.

Автореферат разослан 26 ноября 2012г.

Ученый секретарь специализированного совета,  
доктор физ. мат. наук



А. Г. Саруханян

## Աշխատանքի ընդհանուր բնութագիրը

**Թեմայի արդիականությունը:** Հաճախ գիտական և գիտատեխնիկական խնդիրներ լուծելիս պահանջվում են իրականացնել մեծաքանակ հաշվողական գործողություններ, որոնք էլ պահանջում են հզոր հաշվողական համակարգերի՝ գերկոմպյուտերների առկայությունը: Սակայն վերջիններս ունեն բավականին բարձր արժեք, որն էլ խոչընդոտում է նրանց առավել լայն տարածմանը: Վերջին տասնամյակներում ստեղծվել են առավել հասանելի հաշվողական միջոցներ՝ կլաստերային համակարգեր, որոնք ի տարբերություն գերկոմպյուտերների, ծրագրավորողից պահանջում են զուգահեռ ծրագրավորման մեթոդների և համապատասխան փաթեթների խորը իմացություն:

Հաշվողական կլաստերային համակարգերը տարբերվում են միմյանցից ունեցած հաշվողական ռեսուրսներով և կազմակերպման եղանակներով: Կախված հաշվողական կլաստերների կազմակերպման եղանակից և ծանրաբեռնվածությունից, տարբեր առաջադրանքներ նպատակահարմար է տեղադրել տարբեր համակարգերի հերթերում: Կլաստերների լայնորեն օգտագործումը խթանում է նրանցում հերթերի անհավասարաչափ ծանրաբեռնմանը: Այս և այլ պատճառները հանգեցնում են որոշ հաշվողական համակարգերի միավորմանը մեկ ընդհանուր ծառայության՝ հաշվողական գրիդ ծառայության մեջ: Հաշվողական գրիդ միջավայրի ռեսուրսների կառավարումը և առաջադրանքների սպասարկման պլանավորումը դժվարին խնդիրներից են, որոնց ոչ լիարժեք կազմակերպումը կարող է հանգեցնել ռեսուրսների ցածր ծանրաբեռնմանը և արդյունավետության նվազմանը:

Առաջադրանքների բնութագրիչներին լրացուցիչ սահմանափակումներ դնելով հնարավոր է կատարել առաջադրանքների դասակարգում: Այդպիսի սահմանափակումներից է սպասման ժամանակի սահմանափակումը: Սպասման ժամանակը դա այն ժամանակն է, որի ընթացքում առաջադրանքը կարող է սպասել հերթում մինչև կատարման անցնելը: Հաճախ առաջադրանքների արդյունքները ակտուալ են որոշակի ժամանակահատվածում, բայց հերթի ծանրաբեռնվածության պատճառով հնարավոր է առաջադրանքի սպասարկման ուշացում: Մտցնելով սպասման ժամանակի սահմանափակումը՝ հնարավոր է առաջադրանքը հերթում տեղադրելուց երաշխավորել հստակ ժամանակում սպասարկման անցնելը: Սպասման ժամանակի սահմանափակումը ստեղծում է նոր խնդիրների դաս, որոնցից են գրիդ միջավայրի ռեսուրսների ծանրաբեռնման խնդիրը, գրիդ միջավայրի կողմից սպասարկված առաջադրանքների ավելացման

խնդիրը կամ գրիդ միջավայրի ռեսուրսների հավասարաչափ ծանրաբեռնման խնդիրը:

***Ատենախառության նպատակներն են*** մշակել և իրականացնել սպասման ժամանակի սահմանափակմանը և հերթում ընդունած առաջադրանքի երաշխավորված սպասարկմանը բավարարող գրիդ միջավայրում հերթի ղեկավարման համակարգ՝ տարբեր վարքագծերով: Ստեղծել սիմուլացիոն (փորձարարական) գործիք գրիդ միջավայրի համար, որը ժամանակի դիսկրետ պահերին առաջարկում է հերթի ղեկավարման նախընտրելի վարքագիծ:

***Հետազոտման օբյեկտներն են*** հերթի կազմակերպման մոտեցումները և գրիդ միջավայրի առաջադրանքների պահանջած ռեսուրսների քանակի հավանականությունները:

***Գիտական նորությունը:***

Հետազոտվել և համեմատվել են բազմապրոցեստրային (բազմամեքենայական) հաշվողական համակարգերի հերթի ղեկավարման մեխանիզմները: Մշակվել է սպասման ժամանակի սահմանափակմամբ հերթի ղեկավարման համակարգ:

Առաջադրանքի երաշխավորված սպասարկման համար մշակվել են հերթի ղեկավարման նոր վարքագծեր: Մասնավորապես մշակվել են հերթի ղեկավարման առաջադրանքների հաջորդաբար մշակման, ռեսուրսների առավել ծանրաբեռնող, առաջադրանքների առավել սպասարկող և ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմները:

***Արդյունքների հիմնավորումը և ստույգությունը:*** Հիմնական արդյունքները ձևակերպված են պնդումների տեսքով և բերված են դրանց ապացույցները: Մշակված բոլոր ալգորիթմների համար ապացուցվել են դրանց կոռեկտությունները և տրվել բարդության գնահատականները:

***Գիրառական նշանակությունը:***

Իրականացված է հերթի ղեկավարման համակարգ, որը բավարարում է սպասման ժամանակի սահմանափակմանը և հերթի ընդունած առաջադրանքի երաշխավորված սպասարկմանը: Համակարգը իրականացված է ծրագրային փաթեթի տեսքով C++ և Java ծրագրավորման լեզուների միջոցով, Windows, և Linux Red Hat օպերացիոն համակարգերի համար:

Windows օպերացիոն համակարգի համար ստեղծված է սիմուլացիոն գործիք, որի օգնությամբ գրիդի ադմինիստրատորները կարող են ստուգել

առաջարկվող ալգորիթմները տրված գրիդի և առաջադրանքների հաջորդականության համար:

Linux Red Hat օպերացիոն համակարգերի համար ստեղծված է ծրագրային ապահովում, որը մուտքին ստանում է հերթի առաջադրանքների պահանջները և առաջարկում է հերթի կազմակերպման նախընտրելի վարքագիծը:

**Ներդրումներ:** Ատենախոսության շրջանակներում մշակված փորձարարական ծրագրային փաթեթը ներդրվել է ArmGrid համակարգում:

***Պաշտպանությանը ներկայացված են հետևյալ դրույթները.***

- տարբեր տիպի վարքագծերով հերթի դեկավարման համակարգ,
- ռեսուրսների ծանրաբեռնման խնդրի լուծումը մշակված վարքագծերի միջոցով,
- սպասարկված առաջադրանքների ավելացման խնդրի լուծումը մշակված վարքագծերի միջոցով,
- վարքագծերի գործակիցները որոշող հատկությամբ օժտված սիմուլյացիոն (փորձարարական) գործիքի նկարագիրը:

***Աշխատանքի ապրոքացիան***

Ատենախոսության հիմնական արդյունքները զեկուցվել են Կոմպյուտերագիտության և ինֆորմացիոն տեխնոլոգիաների միջազգային կոնֆերանսում (CSIT-2011, Երևան, Հայաստան) և ՀՀ ԳԱԱ ԻԱՊԻ մասնագիտացված և ընդհանուր սեմինարներում:

**Հրապարակումները:** Ատենախոսության թեմայով հրապարակվել է 3 գիտական աշխատություն, որոնց ցանկը բերված է սեղմագրի վերջում:

**Ատենախոսության կառուցվածքը և ծավալը:** Ատենախոսությունը բաղկացած է ներածությունից, երեք գլուխներից, ամփոփիչ եզրակացությունից և 87 անվանում ընդգրկող գրականության ցանկից: Աշխատանքի ընդհանուր ծավալը կազմում է 107 էջ:

# ԱՇԽԱՏԱՆՔԻ ՀԻՄՆԱԿԱՆ ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆԸ

Ատենախոսությունը կազմված է ներածությունից և երեք գլուխներից:

**Ներածության** մեջ հիմնավորված է ատենախոսության թեմայի արդիականությունը, ձևակերպված են ատենախոսության հիմնական նպատակները և ուսումնասիրության օբյեկտները:

**Առաջին գլխում** կատարված է ժամանակակից բազմապրոցետորային հաշվողական՝ կլաստերային, գրիդային և ամպային, համակարգերի դասակարգումը ըստ իրականացման եղանակների, տիպերի, բնութագրերի, ֆունկցիոնալ և տեխնոլոգիական հատկությունների: Կատարվել է գոյություն ունեցող հերթի ղեկավարման համակարգերի համեմատությունները:

**Երկրորդ գլխում** առաջարկվել է իրականացվել են սպասման ժամանակի սահմանափակմամբ ու հերթ ընդունած առաջադրանքի երաշխավորված սպասարկման հնարավորությամբ հերթի ղեկավարման վարքագծեր: Այդ սահմանափակումներով առաջարկվում է բարձրացնել հաշվողական ենթահամակարգերի ռեսուրսների ծանրաբեռումը, ավելացնել սպասարկվող առաջադրանքների քանակը, ինչպես նաև հավասարաչափ ծանրաբեռնել ենթահամակարգերը:

**§2.1-ում** կատարվում են հիմնական նշանակումները, տրվում են օգտագործվող սիմվոլների բացատրությունները և հաշվողական ռեսուրսների փնտրման և առաջադրանքների սպասարկման կարգը որոշող մոտեցումները:

**§2.2-ում** առաջարկվել է հերթի հաջորդաբար մշակման վարքագիծը: Նկարագրվել է գրիդ միջավայրի հերթի ղեկավարման համակարգի վարքը:  $j$  - րդ ( $i = 1, 2, \dots$ ) առաջադրանքի մուտքի պահը նշանակվում է  $t_i$  - ով, իսկ առաջադրանքը  $(\beta_i, \vartheta_i, \omega_i)$  - ով, որտեղ  $\beta_i$  - ն սպասարկման տևողությունն է,  $\vartheta_i$  - ն պահանջվող ռեսուրսների քանակը և  $\omega_i$  - ն առավելագույն սպասման ժամանակը:  $t_1$  պահին ընդունած  $(\beta_1, \vartheta_1, \omega_1)$  առաջադրանքի համար ընտրվում է նվազագույն ռեսուրսներ ունեցող ենթահամակարգը, որը կարող է սպասարկել առաջադրանքը:  $t_i$  պահին ընդունած  $(\beta_i, \vartheta_i, \omega_i)$  ( $i = 2, 3, \dots$ ) առաջադրանքի համար՝ նախ ընտրվում են այն հաշվողական ենթահամակարգերը, որոնք ունեն առնվազն  $\vartheta_i$  ռեսուրսներ, ապա կարգավորվում են ենթահամակարգերը ըստ

ունեցած ռեսուրսների քանակի՝ աճման կարգով: Ընտրված ենթահամակարգերի համար, սկսած նվազագույն ռեսուրսներ ունեցողից, հաշվարկում է առաջադրանքի սպասարկման անցնելու հարաբերական ժամանակը՝  $\tau_i$  մեծությունը: Գտնելով այնպիսի ենթահամակարգ, որի համար բավարարվում է  $\tau_i \leq \omega_i$  պայմանը, առաջադրանքը տեղադրում է ընտրված ենթահամակարգի սպասարկման հերթում՝  $t_i + \tau_i$  ժամանակին սպասարկվելու համար և անցնում հաջորդ առաջադրանքին: Չգտնելով համապատասխան ենթահամակարգ, որի համար բավարարվում է  $\tau_i \leq \omega_i$  պայմանը, առաջադրանքի սպասարկումը մերժվում է:

**§2.3-ում** առաջարկվել է ռեսուրսների առավել ծանրաբեռնող վարքագիծը: Ժամանակի  $t_i$  պահին ընդունած  $(\beta_i, \vartheta_i, \omega_i)$  առաջադրանքի համար ընտրվում է հաջորդաբար մշակման վարքագիծը:  $t_i$  պահին ընդունած  $(\beta_i, \vartheta_i, \omega_i)$  ( $i = 2, 3, \dots$ ) առաջադրանքի համար նախ ընտրվում են այն հաշվողական ենթահամակարգերը, որոնք ունեն առնվազն  $\vartheta_i$  ռեսուրսներ, ապա կարգավորվում են ենթահամակարգերը, ըստ ունեցած ռեսուրսների քանակի՝ աճման կարգով: Այնուհետև հերթի դեկավարման համակարգը, ընտրված ենթահամակարգերի համար, հերթականությամբ, սկսած նվազագույն ռեսուրսներ ունեցող ենթահամակարգից, կատարում է հետևյալ գործողությունները՝

1. որոշում է կայացնում ընդունած առաջադրանքի սպասարկելու մասին՝ հիմնվելով առաջադրանքի ռեսուրսների պահանջի, գրիդ միջավայրի ծանրաբեռնվածության և նախապես սահմանված գործակիցների հիման վրա:
2. հաշվարկում է առաջադրանքի սպասարկման անցնելու հարաբերական ժամանակը՝  $\tau_i$  մեծությունը:

Հերթի դեկավարման համակարգը՝

- գնելով այնպիսի ենթահամակարգ, որի համար դրական որոշում է կայացրել առաջին գործողությունը և բավարարվում է  $\tau_i \leq \omega_i$  պայմանը, առաջադրանքը տեղադրում է ընտրված ենթահամակարգի սպասարկման հերթում՝  $t_i + \tau_i$  ժամանակին սպասարկվելու համար և անցնում հաջորդ առաջադրանքին,

- չգտնելով համապատասխան ենթահամակարգ, որի համար բավարարվում է  $\tau_i \leq \omega_i$  պայմանը, առաջադրանքի սպասարկումը մերժում է:

**Տ2.4-ում** առաջարկվել է առաջադրանքների առավել սպասարկող վարքագիծը: Ժամանակի  $t_1$  պահին ընդունած  $(\beta_1, \vartheta_1, \omega_1)$  առաջադրանքի համար ընտրվում է հաջորդաբար մշակման վարքագիծը:  $t_i$  պահին ընդունած  $(\beta_i, \vartheta_i, \omega_i)$  ( $i = 2, 3, \dots$ ) առաջադրանքի համար նախ ընտրվում են այն ենթահամակարգերը, որոնք ունեն առնվազն  $\vartheta_i$  ռեսուրսներ, ապա կարգավորվում են ենթահամակարգերը, ըստ  $[t_i, t_i + \beta_i + \omega_i]$  միջակայքում հերթում գտնվող առաջադրանքների խտության գործակիցների՝ նվազման կարգով:  $M_j$  ենթահամակարգի հերթի խտությունը  $[t', t'']$  միջակայքում, հաշվարկվում է հետևյալ բանաձևով՝

$$\tilde{D}(t', t'') = \frac{\sum_{\tau_{l_k} \in [t', t'']} \vartheta_{l_k} \cdot (\min(t'', \tau_{l_k} + \beta_{l_k}) - t')}{R_j \cdot (t'' - t')}$$

որտեղ  $R_j$  – ն  $M_j$  ենթահամակարգի ունեցած ռեսուրսների քանակն է:

Այնուհետև հերթի ղեկավարման համակարգը, ընտրված ենթահամակարգերի համար, հերթականությամբ, սկսած առավելագույն խտության գործակից ունեցող ենթահամակարգից, կատարում է հետևյալ գործողությունները՝

1. որոշում է կայացնում ընդունած առաջադրանքի սպասարկելու մասին՝ հիմնվելով առաջադրանքի ռեսուրսների պահանջի, գրիդ միջավայրի ծանրաբեռնվածության և նախապես սահմանված գործակիցների հիման վրա:
2. հաշվարկում է առաջադրանքի սպասարկման անցնելու հարաբերական ժամանակը՝  $\tau_i$  մեծությունը:

Հերթի ղեկավարման համակարգը՝

- գնելով այնպիսի ենթահամակարգ, որի համար դրական որոշում է կայացրել առաջին գործողությունը և բավարարվում է  $\tau_i \leq \omega_i$  պայմանը, առաջադրանքը տեղադրում է ընտրված ենթահամակարգի սպասարկման



հերթում՝  $t_i + \tau_i$  ժամանակին սպասարկվելու համար և անցնում հաջորդ առաջադրանքին,

- չգտնելով համապատասխան ենթահամակարգ, որի համար բավարարվում է  $\tau_i \leq \omega_i$  պայմանը, առաջադրանքի սպասարկումը մերժում է:

Հարկ է նշել, որ առաջադրանքների առավել սպասարկող ալգորիթմում առաջադրանքի սպասարկման անցնելու որոշման կայացման, և ռեսուրսների առավել ծանրաբեռնող ալգորիթմում առաջադրանքի սպասարկման անցնելու որոշման կայացման ալգորիթմները տարբեր են:

**§2.5-ում** առաջարկվել է ռեսուրսների հավասարաչափ ծանրաբեռնող եղանակը: Ժամանակի  $t_1$  պահին ընդունած  $(\beta_1, \vartheta_1, \omega_1)$  առաջադրանքի համար ընտրվում է հաջորդաբար մշակման վարքագիծը:  $t_i$  պահին ընդունած  $(\beta_j, \vartheta_j, \omega_j)$  ( $i = 2, 3, \dots$ ) առաջադրանքի համար նախ ընտրվում են այն ենթահամակարգերը, որոնք ունեն առնվազն  $\vartheta_j$  ռեսուրսներ, ապա կարգավորվում են ենթահամակարգերը, ըստ *Rait* – ի արժեքի՝ աճման կարգով:

$$Rait = R_j - \vartheta_j - \left[ \frac{R_j - \vartheta_j}{Average} \right]$$

որտեղ *Average* – ը ընդունած առաջադրանքների պահանջած ռեսուրսների թվաբանական միջինն է: Ընտրված ենթահամակարգերի համար, սկսած նվազագույն *Rait* ունեցողից, հաշվարկում է առաջադրանքի սպասարկման անցնելու հարաբերական ժամանակը՝  $\tau_i$  մեծությունը: Գտնելով այնպիսի ենթահամակարգ, որի համար բավարարվում է  $\tau_i \leq \omega_i$  պայմանը, առաջադրանքը տեղադրում է ընտրված ենթահամակարգի սպասարկման հերթում՝  $t_i + \tau_i$  ժամանակին սպասարկվելու համար և անցնում հաջորդ առաջադրանքին: Չգտնելով համապատասխան ենթահամակարգ, որի համար բավարարվում է  $\tau_i \leq \omega_i$  պայմանը, առաջադրանքի սպասարկումը մերժվում է:

Վերը նշված վարքագծերի համար բերված են բոլոր պրոցեդուրաների և ենթապրոցեդուրաների նկարագրությունները բանաձևերով և պսևոկոդով, ոչ տրիվյալ պրոցեդուրաների համար ձևակերպված և ապացուցված են կոռեկտության պնդումները, ինչպես նաև տրված են ալգորիթմների բարդության գնահատականները: Իրականացված են բոլոր վարքագծերի ծրագրային ապահովումները:

**Երրորդ գլխում** ձևակերպվել են ռեսուրսների ծանրաբեռնման և սպասարկվող առաջադրանքների ավելացման խնդիրները: Առաջարկվել և իրականացվել են լուծումներ՝ հիմնված երկրորդ գլխում նկարագրված ալգորիթմների վրա: Կատարվել է ստացված լուծումների համեմատություն: Իրականացվել է նախընտրելի ալգորիթմ որոշող ծրագրային փաթեթ՝ համակարգի վիճակի և առաջադրանքների հոսքի վերլուծության հիման վրա:

**§3.1-ում** տրվել են գրիդ միջավայրի ռեսուրսների ծանրաբեռնման ու գրիդի կողմից սպասարկված առաջադրանքների ավելացման խնդիրների ձևակերպումները: Հիմնավորվել են ձևակերպված խնդիրների արդիականությունը: Կատարվել են պարագրաֆում օգտագործված սիմվոլների նշանակումները:

**§3.2-ում** դիտարկվել է գրիդի ռեսուրսների ծանրաբեռնման խնդիրը: Այդ խնդրի լուծման համար առաջարկվել են նախորդ գլխում նկարագրված առաջադրանքների հաջորդաբար մշակման, ռեսուրսների առավել ծանրաբեռնող և ռեսուրսների հավասարաչափ ծանրաբեռնող վարքագծերը: Համեմատվել են նախ առաջադրանքների հաջորդաբար մշակման և ռեսուրսների առավել ծանրաբեռնող, ապա առաջադրանքների հաջորդաբար մշակման և ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմների վարքագծերը՝ կախված առաջադրանքների հոսքից:

Առաջադրանքների հաջորդաբար մշակման և ռեսուրսների առավել ծանրաբեռնող ալգորիթմների համեմատման արդյունքում ձևակերպվել և ապացուցվել է պնդում, որում նշված են այն նախապայմանները, որոնց բավարարման դեպքում նպատակահարմար է հերթը սպասարկել ռեսուրսների առավել ծանրաբեռնող վարքագծով: Նախապայմանները հետևյալն են՝

- $0 \leq K_1 < K_2 < 1$
- $1 - K_1 \leq K_1 + K_2 < 1$
- $\sum_{i=m}^{m(1-K_2)} p(i) > \frac{1}{2}$

Որտեղ  $K_1 - p$  և  $K_2 - p$  ռեսուրսների առավել ծանրաբեռնող վարքագծի ղեկավարման գործակիցներն են,  $m - p$  առավելագույն ռեսուրսներ ունեցող համակարգի ռեսուրսների քանակն է, իսկ  $p(i) -$  ն  $i$  միավոր հաշվողական ռեսուրս պահանջող առաջադրանքի հավանականությունն է:

Առաջադրանքների հաջորդաբար մշակման և ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմների համեմատման արդյունքում

ձևակերպվել և ապացուցվել է թեորեմ, որում բերվում են այն նախապայմանները, որոնց բավարարման դեպքում նպատակահարմար է հերթը սպասարկել ռեսուրսների հավասարաչափ ծանրաբեռնող վարքագծով:

Պարագրաֆում տրվել է ընդհանրական եզրակացություն ալգորիթմների կիրառման ոլորտների մասին:

- ռեսուրսների առավել ծանրաբեռնող ալգորիթմը հարմար է կիրառել այն համակարգերում, որտեղ նախապես հայտնի են առաջադրանքների սպասարկման համար պահանջվող ռեսուրսների պատահման հավանականությունները: Անհրաժեշտ է կատարել հաշվարկներ ալգորիթմի ղեկավարման գործակիցները ստանալու համար:
- ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմը հարմար է օգտագործել այն համակարգերում, որտեղ հայտնի են առաջադրանքների սպասարկման համար պահանջվող ռեսուրսների պատահման հավանականությունը, սակայն այս ալգորիթմը ինքնաղեկավարվող է, և նախապես գործակիցներ որոշելու կարիք չկա:
- հաջորդաբար մշակման ալգորիթմը հարմար է օգտագործել այն համակարգերում, որտեղ առաջադրանքների սպասարկման համար պահանջվող ռեսուրսների պատահման հավանականությունը կամ հայտնի չէ, կամ չի բավարարում ռեսուրսների առավել ծանրաբեռնող և ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմների նախապայմաններին:

**§3.3-ում** դիտարկվել է գրիդ միջավայրի կողմից սպասարկված առաջադրանքների ավելացման խնդիրը: Այդ խնդրի լուծման համար առաջարկվել են նախորդ գլխում նկարագրված՝ առաջադրանքների հաջորդաբար մշակման, առաջադրանքների առավել սպասարկող և ռեսուրսների հավասարաչափ ծանրաբեռնող վարքագծերը: Համեմատվել են նախ առաջադրանքների հաջորդաբար մշակման և առաջադրանքների առավել սպասարկող, սպա առաջադրանքների հաջորդաբար մշակման և ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմների վարքագծերը կախված առաջադրանքների հոսքից:

Առաջադրանքների հաջորդաբար մշակման և առաջադրանքների առավել սպասարկող ալգորիթմների համեմատման արդյունքում ձևակերպվել և ապացուցվել է թեորեմ, որում բերվում են այն նախապայմանները, որոնց բավարարման դեպքում նպատակահարմար է հերթը սպասարկել առաջադրանքների առավել սպասարկող վարքագծով:

Առաջադրանքների հաջողաբար մշակման և ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմների համեմատման արդյունքում ձևակերպվել և ապացուցվել է թեորեմ, որում բերվում են այն նախապայմանները, որոնց բավարարման դեպքում նպատակահարմար է հերթը սպասարկել ռեսուրսների հավասարաչափ ծանրաբեռնող վարքագծով:

Պարագրաֆում տրվել է ընդհանրական եզրակացություն ալգորիթմների կիրառման ոլորտների մասին:

- առաջադրանքների առավել սպասարկող ալգորիթմը հարմար է կիրառել այն համակարգերում, որտեղ նախապես հայտնի են առաջադրանքների սպասարկման համար պահանջվող ռեսուրսների պատահման հավանականությունները: Անհրաժեշտ է կատարել հաշվարկներ ալգորիթմի դեկավարման գործակիցները ստանալու համար:
- ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմը հարմար է օգտագործել այն համակարգերում, որտեղ հայտնի են առաջադրանքների սպասարկման համար պահանջվող ռեսուրսների պատահման հավանականությունը, սակայն այս ալգորիթմը ինքնադեկավարվող է և նախապես գործակիցներ որոշելու կարիք չկա:
- հաջողաբար մշակող ալգորիթմը հարմար է օգտագործել այն համակարգերում, որտեղ առաջադրանքների սպասարկման համար պահանջվող ռեսուրսների պատահման հավանականությունը կամ հայտնի չէ, կամ չի բավարարում առաջադրանքների առավել սպասարկող և ռեսուրսների հավասարաչափ ծանրաբեռնող ալգորիթմների նախապայմաններին:

**§3.4-ում** մշակվել և իրականացվել է սիմուլյացիոն (փորձարարական) ծրագրային փաթեթ, որը ընդգրկվելով գրիդ միջավայրին, ստանում է ընդունած առաջադրանքների հոսքը և հաշվարկում նրանց պատահման հավանականությունները: Հաշվարկվել են նաև այն պայմանները, որոնց բավարարելու դեպքում կատարվում են վարքագծերի ընտրություն:

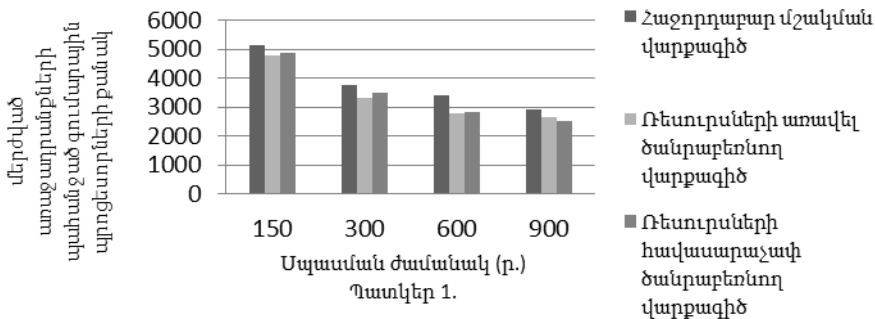
***Միմուլյացիոն (փորձարարական) գործիքի ստացած արդյունքները:***

Ստորև պատկերված են գրաֆիկներ, որոնք ցույց են տալիս իրականացված փորձարարական գործիքից ստացած արդյունքները: Որպես գրիդ միջավայր ընտրվում է 6 կլաստերներից կազմված միջավայր, որոնցից չորսը ունեն 48 պրոցեսոր, իսկ երկուսը 128 պրոցեսոր հաշվողական ռեսուրս:

Պատկեր 1-ում ներկայացված է ըստ վարքագծերի մերժված առաջադրանքների պահանջած գումարային պրոցեստրների քանակի կախումը առաջադրանքների համար սահմանված առավելագույն սպասման ժամանակի պատահման տիրույթից:

Համակարգ ընդունած առաջադրանքների բնութագրերն են

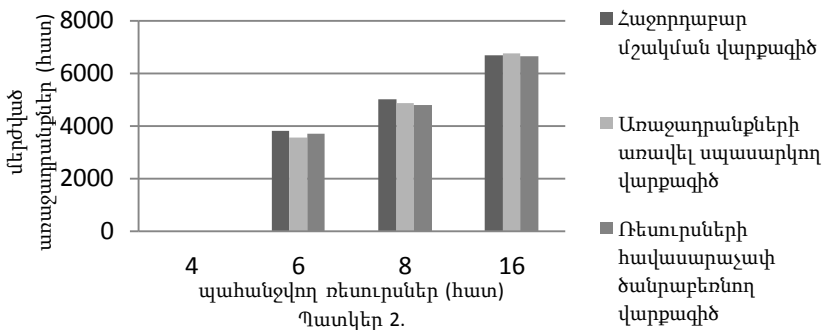
$$p(k) = \begin{cases} \frac{9}{160}, k = 1, 2, \dots, 16 \\ \frac{1}{1120}, k = 17, 18, \dots, 128 \end{cases}, t_i \in [0, 45], \beta_i \in [1, 900], K_1 = \frac{1}{10}, K_2 = \frac{17}{20}$$



Պատկեր 2-ում ներկայացված է ըստ վարքագծերի մերժված առաջադրանքների քանակի կախումը առաջադրանքների սպասարկման համար պահանջվող ռեսուրսների պատահման տիրույթից:

Համակարգ ընդունած առաջադրանքների բնութագրերն են:

$$\omega_i \in [0, 150], t_i \in [0, 150], \beta_i \in [1, 900], K_1 = \frac{3}{50}, K_2 = \frac{87}{100}$$



### ***Այսպիսով աշխատանքի հիմնական արդյունքները հետևյալն են՝***

1. Մշակվել և իրականացվել են գրիդ միջավայրի հերթի ղեկավարման համակարգ, որը բավարարում է սպասման ժամանակի սահմանափակմանը և երաշխավորում ընդունած առաջադրանքի սպասարկումը:
2. Առաջարկվել և իրականացվել է հերթի ղեկավարման համակարգի պարամետրիզացված վարքագիծ՝ կախված ռեսուրսների ծանրաբեռնման աստիճանի և սպասարկված առաջադրանքների քանակի տարբեր պահանջներից:
3. Առաջարկվել և իրականացվել են հերթի ղեկավարման համակարգի դինամիկ վարքագծեր՝ սպասարկված առաջադրանքների քանակի ավելացման համար, հաշվի առնելով հերթ ընդունած առաջադրանքների հավանականային բնութագրերը:
4. Կատարվել են մշակված ալգորիթմների ծրագրային փաթեթի իրականացում, կարգավորում և ներդրում գրիդ միջավայրում:

### **Ատենախոսության թեմատիկայով հրապարակված աշխատությունների ցուցակ**

1. S. Petrosyan, “Grid static queue dividing into high performance clusters queue,” In Proc. of the 8 th International Conference on Computer Science and Information Technologies, *Yerevan, Armenia, September 26 - 30, 2011, pp.263-265.*
2. V. Sahakyan, S. Petrosyan, “Simulation of the queue with the restriction on the waiting time for multiprocessor systems,” In Proc. of the 8 th International Conference on Computer Science and Information Technologies, *Yerevan, Armenia, September 26 - 30, 2011, pp.272-273.*
3. V. Sahakyan, S. Petrosyan, “Workload management for Grid environment with the restriction on the waiting time” *Mathematical Problems of computer science XXXVI, Yerevan, Armenia, 2012, pp.121-127*

## ДИНАМИЧЕСКОЕ УПРАВЛЕНИЕ ОЧЕРЕДЯМИ В СРЕДЕ ГРИД

### *РЕЗЮМЕ*

В различных областях науки и техники часто возникают задачи, для решения которых требуется выполнение многочисленных вычислительных операций и мощные вычислительные системы, суперкомпьютеры. С конца 20-ого века развиваются многопроцессорные системы кластерного типа, которые предоставляют из себя объединение большого количество компьютеров. Вычислительные кластерные системы, или просто кластеры, можно сконфигурировать для конкретных типов вычислений, что приводит к повышению производительности этих типов вычислений. В основах кластеров лежит эффективная и согласованная организация параллельных процессов.

Поскольку, кластеры конфигурируются по разному и отличаются по специфике решаемых задач, пользователь должен иметь возможность выполнять свои задачи на различных кластерах. Поэтому, кластеры объединяются в единую грид систему, и пользователи могут выполнять свои задачи в единой инфраструктуре.

Эффективное управление ресурсами и планирование выполнения задач в кластерных, грид и облачных средах являются труднорешаемыми задачами. Несмотря на значительные результаты, полученные ранее, остаются проблемы оптимального управления вычислительными ресурсами. Такой проблемой является своевременное выполнение задачи, поступающей в грид систему. Своевременное выполнение задачи является актуальной задачей для многих приложений в экономике и системах управления. Наличие ограничения на время ожидания приводит к необходимости гарантирования выполнения задачи в момент ее поступления в систему или отказ от обслуживания.

*Основная цель исследования* состоит в исследовании, разработке и внедрении неприоритетной системы организации очереди для вычислительных грид инфраструктур, которая удовлетворяет ограничениям времени ожидания и гарантирует выделение соответствующих ресурсов для выполнения задач, а также в оценке эффективности, предложенных для реализации системы алгоритмов. Система должна также иметь возможность дальнейшего расширения.

### *Научная новизна работы*

Проведено исследование и сравнение систем организации очередей для многомашинных вычислительных комплексов. Предложено использование системы управления очередями с учетом ограничений на время ожидания.

Предложены новые подходы для организации очереди с гарантированным обслуживанием задач. Для таких очередей разработаны алгоритмы последовательного выполнения задач, наибольшей загрузки ресурсов, равномерной загрузки ресурсов, выполнения наибольшего количества заданий.

***Основные результаты работы следующие:***

1. Разработана и внедрена система управления очередью для вычислительной грид инфраструктуры, удовлетворяющая ограничениям времени ожидания и гарантированного выполнения задач.
2. Предложены и реализованы параметрические сценарии организации очереди с различными требованиями на загрузку ресурсов и количества выполненных задач.
3. Предложены и реализованы динамические сценарии организации очередей для увеличения количества обслуживаний с учетом вероятностных характеристик поступающего потока задач. Разработан алгоритм динамического вычисления управляющих параметров.
4. Осуществлена программная реализация и внедрение разработанных алгоритмов в грид среде.



WORKFLOW MANAGEMENT IN COMPUTING GRID ENVIRONMENTS

**ABSTRACT**

In various scientific areas there are many tasks that require a large number of computational operations and high power computing systems like supercomputers. Since the end of 20-rd century till now there were developed cluster type of multiprocessor systems. The cluster system or just clusters may be configured for custom computing types, this will increase the productivity for that type of computing. The core component of clusters is coordinated parallel programming.

As clusters differ in their settings and designed for a specific class of tasks, users should have the opportunity to work on different clusters, to be able to run their tasks in different clusters. This is a reason of collecting all clusters into one computational grid infrastructure, and permit users to run their task in that computational grid infrastructure.

Efficiently resource management and job scheduling in cluster, grid and cloud environments are challenging problems. Although significant results were achieved in the past, there are some problems that still exist. Such problem is the job execution on schedule. Job execution on schedule is urgent for many commercial systems. The existence of waiting time restriction either results to guaranty the job execution or to decline it on incoming time.

***The main aims of the research are*** to research, develop and implement a workflow management system for computational grid environment that meet waiting time restriction and guaranty availability of appropriated resource for task execution. This systems must have availability to extend. The aim is also to get an evaluation for developed systems.

***Scientific novelty of the work are:***

Investigated and compared the workflow management systems for multiprocessor computational complexes. Suggested the usage of workflow management system with waiting time restriction.

Suggested new ways for workflow management with guaranteed job execution. Developed tasks sequentially execution, resource utilization, more tasks execution and equal resource utilization algorithms.

***The main results of the works are following:***

1. Developed and implemented four workflow management systems for computational grid environment that meet of waiting time restriction and warranty availability of appropriated resource for task execution.
2. Suggested and implemented parameterized scenarios of workflow management with varies demands on resource load and executed job quantity.
3. Suggested and implemented dynamic scenarios of workflow management system to increase the quantity of executed jobs considering probabilistic characteristics of job flow. Developed algorithm for dynamic calculating of management parameters.
4. Developed a solution for grid environment more task execution problem, depending on incoming tasks frequency and required resource occurring probability.
5. Developed programing realization and implementation of algorithms for grid environment.

A handwritten signature in black ink, appearing to be 'A. J. ...', written in a cursive style.