

УДК 621

А.І. Петренко, д-р техн. наук

Комп'ютерне моделювання грид-систем

Рассмотрен и проанализирован программный инструмент моделирования GridSim, предназначенный для исследования механизмов планирования и использования ресурсов в грид-системах с целью оценки пригодности их использования для моделирования поведения разных возможных вариантов национальной грид-инфраструктуры.

The program design tool GridSim is considered and analysed, which is intended for research of planning and resources use mechanisms in the Grid-systems for the purpose of estimation of their capability to behavioral simulation of the national Grid-infrastructure different possible variants.

Вступ

Останнім часом за кордоном і в Україні активно ведуться дослідження з грид-технологій, які забезпечують доступ користувачів до географічно розподілених комп'ютерних ресурсів. Об'єднуючи їх в єдину розподілену систему обчислень, зберігання і передачі інформації [1,2].

На сьогодні вже створена технічна база (телекомунікаційні мережі великої пропускної спроможності, високопродуктивні кластери) і зроблені кроки по створенню проміжного програмного забезпечення для роботи в грид-середовищі. В Україні розповсюджено проміжне ПЗ типу **NorduGrid** [3] і **gLite** [4] з відкритим початковим кодом, які застосовуються для створення національної грид-інфраструктури і забезпечують засоби безпеки в розподіленому середовищі, засоби надійної передачі великих об'ємів даних, засоби запуску і отримання результатів виконання завдань на віддалених обчислювальних ресурсах.

Не дивлячись на вже існуючі засоби створення грид-інфраструктур, існує ряд важливих наукових завдань, без вирішення яких неможливе повномасштабне використання можливостей грид-технологій в науці, освіті і промисловості. Одним з таких актуальних завдань є ефективне управління ресурсами в розподіленому середовищі [5]. Відсутність ефективного планувальника (брокера) в NorduGrid, що забезпечує управління потоком завдань, не тільки значно знижує ефективність використання всієї грид-інфраструктури, але може зробити необґрунтованим її створення. Вибір (можливо, розроблення) такого планувальника для національної грид-інфраструктури повинні спи-

ратися не тільки на наявність брокера в gLite, але і на вивчення тенденцій їх розвитку на основі проведення експериментів на базі існуючих грид-сегментів (академічного і освітянського) з застосуванням високорівневих моделей, що адекватно відображають властивості таких складних систем.

Інструментарій моделювання

У зв'язку з цим одним з актуальних завдань є вибір чи створення системи моделювання розподіленої обчислювальної грид-інфраструктури, яка дозволить адекватно оцінювати її поведінку за умов, коли змінюються і оптимізується стратегія управління потоками завдань. В даний час існує декілька проектів з розробки систем моделювання розподілених грид-інфраструктур. Серед них найбільш відомими є Bricks [6], MicroGrid [7], OptorSim [8], SimGrid [9], GridSim [10]. Кожна з них має свої переваги і недоліки. Серед недоліків можна відзначити вузьку спеціалізацію деяких систем (OtorSim), відсутність публічно доступних версій (Bricks), а також обмеженість модельованої архітектури грид-систем. Особливості реалізацій деяких з них накладають обмеження на кількість одночасно існуючих елементів в грид-системі і вимагають від користувача знання спеціальних мов програмування, що істотно знижує ефективність їх використання (табл.1).

Планувальники у розподіленому обчислювальному середовищі, що називають брокерами ресурсів, виконують пошук ресурсів, а також вибір і агрегацію різних наборів розподілених ресурсів для індивідуального користувача. Всі користувачі повинні надавати свої задачі в центральний планувальник, що може бути орієнтований на здійснення глобальної оптимізації, наприклад, більш високий ступінь використання системних ресурсів і підвищення задоволеності користувачів у залежності від політики розподілу ресурсів, або на оптимізацію роботи для високо пріоритетних користувачів.

З даних табл.1, видно, що SimGrid і GridSim здебільшого є універсальними програмами моделювання грид-систем обчислювального типу (Computing Grid). Отже, вони в них відсутня функція як копіювання даних і запит на місце розташування копії, яка є базовою для грид-систем даних (Data Grid). В свою чергу, MicroGrid з програмою моделювання мережі MaSSF є зручним для оцінювання динамічної поведінки проміжного ПЗ, прикладних програм і мережевих сервісів.

Таблиця 1. Порівняльні характеристики засобів моделювання ґрід-систем

	MicroGrid	OptorSim	SimGrid	GridSim
Мова	C++	Java	C	Java
Схема моделювання	конфігураційні файли (xml, plain text). Топологія мережі з описом на описом на SSFNET DML	конфігураційні файли (xml, plain text).	C, конфігураційні файли (xml)	Java конфігураційні файли у будь-якій комбінації
Вхідні коди	так	так	так	так
Докумен-тація	так	так	так	повна документація класів з використанням javadoc
Приклади	так	так	так	так
Платформа			SURF	SimJava
Моделювання	емуляція з використанням реальних застосунків (підтримка лише тільки Globus <=2.4.2)	реплікацію даних	центральний планувальник	CE, SE, мережі, множені планувальників
Моделювання мережі	статичний/ динамічний роутинг, транспортні протоколи TCP/UDP, пакетування даних, runtime статус, фоновий трафік з заданим розподіленням	рудиментарна підтримка (статичний роутинг)	статичний роутинг, транспортний протокол TCP, runtime статус, фоновий трафік згідно Network Weather Service (NWS)	статичний/динамічний роутинг, транспортний протокол UDP, пакетування даних, runtime статус, QoS, фоновий трафік з заданим розподіленням
Призначення	повна емуляція ґрід-мережі	моделювання алгоритмів доступу до даних і стратегій реплікації	достатньо повне моделювання, в том числі і алгоритмів планувальників.	достатньо повне моделювання, в тому числі дослідження ефективності брокерів ресурсів і алгоритмів планувальників.

В ІПСА в рамках робіт з паралельних обчисленнях і ґрід-технологій проведено порівняльне дослідження існуючих засобів моделювання ґрід-систем і вибрано інструментарій GridSim для проведення більш детальних експериментів (рис.1).

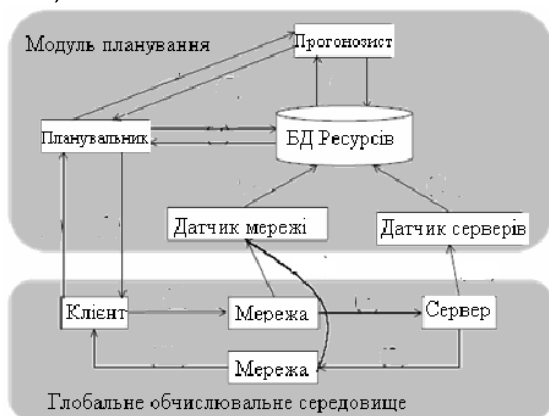


Рис. 1. Схема потоків даних в GridSim

Архітектура будь-якого середовища моделювання ґрід-систем складається з наступних

основних компонентів: графічний інтерфейс користувача, модель ґрід-системи, транслятор, виконувана модель, монітор виконання, модуль статистики. Основний варіант використання такої моделі можна описати таким чином:

- Користувач за допомогою графічного інтерфейсу задає характеристики модельованої ґрід-системи і визначає параметри моделювання.

- Транслятор перетворює цей опис ґрід-системи в код програми моделювання. Отримання виконуваної моделі відбувається автоматично, від користувача не вимагається ніяких додаткових дій. Якщо опис містить помилки, транслятор повідомляє про це користувача.

- Користувач запускає модель. Під час роботи програми монітор здійснює збір інформації про події, що відбуваються (рис.1). Дана інформація зберігається в файлі виконання, аналізується модулем статистики і виводиться користувачеві. Після завершення виконання цей файл може аналізуватися окремо.

Опис модельованого середовища в різних програмах моделювання проводиться за допомогою спеціальної мови, що має як графічну, так і текстову нотацію. У призначеному для користувача інтерфейсі реалізована можливість перегляду і редагування різних аспектів моделі: списку кластерів, потоків завдань, характеристик окремого завдання, брокерів, мережових з'єднань і так далі. Потік завдань в моделі визначається наступними параметрами: властивостями завдання; періодом часу до породження чергового завдання; кількістю породжуваних завдань; затримкою перед початком породження першого завдання. Користувач має можливість задати визначення свого власного потоку завдань.

Середовище моделювання грід-систем дозволяє описати алгоритми розподілу потоку завдань між ресурсними центрами розподіленого середовища брокером, а також алгоритми розподілу завдань локальними планувальниками на вузлах кожного кластера. Алгоритм може бути описаний користувачем за допомогою вбудованої мови, або підключений у вигляді Java-класа, що реалізовує один з відомих планувальників: **Nimrod-G** [11], **GraDS** [12], **Condor-G** [13].

Основні задачі, розв'язувані системою моделювання грід-систем:

- Виявлення "слабких місць" системи ще на етапі проектування.

- Можливість побудови різних варіантів мережі, з метою пошуку найбільш прийняттого.
- Пошук найкращої конфігурації для оптимального розподілу ресурсів

Моделювання української грід-інфраструктури

Проведене моделювання грід-інфраструктури України з централізованим брокером (у випадку ППЗ gLite) і локальними брокерами в окремих вузлах (у випадку ППЗ NorduGrid) визначає критичне значення одночасно виконуваних завдань в мережі, яке обумовлює доцільність переходу до використання ППЗ gLite.

Для моделювання грід-інфраструктури використовуються наступні вхідні дані:

- Загальна кількість задач, що потрібно обробити (шт.).
- Довжина задачі (MIPS).
- Здатність ресурсів (MIPS).
- Пропускна здатність кожного ресурсу (Мбіт/с)
- Час очікування (WaitTime) – час очікування завдань в черзі (с).
- Час виконання (RunTime) – час виконання завдань (с).

Всього було проведено шість експериментів, перший з яких присвячено дослідженню залежності часу моделювання від довжини задач.

Таблиця 2. Приблизні характеристики кластерів української грід-інфраструктури

№	Назва	Місце знаходження	Пропускна здатність ресурсу	ОС	Тип процесору	Кількість процесорів	Здатність ресурсу (MIPS)
1	BITP_Cluster	Київ	100 Мбіт/с	Scientific Linux 4	Intel(R) Xeon(TM) CPU 2.80GHz	96	5600
2	THEI_Cluster	Київ	10 Мбіт/с	Fedora Core 9	3x2AMD 64bit, 6x6 I686 P III	9	2000
3	SRI_Cluster	Київ	10 Мбіт/с	CentOS 4	2xAMD(R) Opteron(TM) CPU 2.80GHz	16	5660
4	PIMEE_Cluster	Київ	10 Мбіт/с	Linux	-	24	3992
5	MAO_Cluster	Київ	10 Мбіт/с	Debian GNU/Linux 4.0	2xIntel(R) Xeon(R) CPU 5130 2.00GHz	104	4000 5000
6	KPI_Cluster	Київ	100 Мбіт/с	CentOS-5.2	4xXeon 64bit, 8xXeon 64bit	624	6000 5600
7	KNU ARC_Cluster	Київ	10 Мбіт/с	Linux-2.6.32, glibc-2.10.1, Fedora-11	Intel(R) Xeon(R) X5355@ 2.66 GHz	72	5320

Таблиця 2. Приблизні характеристики кластерів української гід-інфраструктури (продовження)

8	KMA GRID_Cluster	Київ	10 Мбіт/с	SunOS 5.11 snv_111b	2xAMD(R) Opteron(TM) CPU 2.80GHz	13	5600
9	ISOFTS_Cluster	Київ	10 Мбіт/с	Scientific Linux 4	8xIntel(R) Xeon(TM) C P U 2.80GHz	40	5600
10	IOP_Cluster	Київ	10 Мбіт/с	Linux- 2.6.18	2xIntel(R) Xeon(R) C P U 3050 @ 2.13GHz	80	4260
11	Inpar- com_Cluster	Київ	10 Мбіт/с	CentOS 5.1	2xIntel(R) Xeon(TM) C P U 2.33GHz	-	4660
12	IMP_Cluster	Київ	10 Мбіт/с	Scientific Linux 5.0	Intel(R) Xeon(R) C P U E 5320 @ 1.86GHz @ 1869.000 MHz	56	3738
13	IMMSP_Cluster	Київ	10 Мбіт/с	glibc-2.7-2, Fedora Core 8	Intel(R) Xeon(R) CPU E 5405 @ 2.00GHz	24	4000
14	IMBG_Cluster	Київ	10 Мбіт/с	Linux- 2.6.26.7A , glibc-2.8.8, Fedora-9	Dual Core AMD Opteron (tm) Proces- sor 270 @ 1000.000 MHz	28	2000
15	ICYB SCIT- 3_Cluster	Київ	100 Мбіт/с	CentOS-5	4xXeon 64bit, 8x Xeon 64bit	792	6000 4666
16	DFTI_Cluster	Донецьк	10 Мбіт/с	2.6.18- 128.1.1.el5	Intel(R) Xeon(TM) CPU 2.33GHz	32	4655
17	ICMP_Cluster	Львів	10 Мбіт/с	CentOS 4.4	2xIntel(R) Xeon(TM) 5130 CPU 2.0GHz	120	5660
18	IAP_Cluster	Суми	10 Мбіт/с	Schientific Linux 5	Intel(R) Xeon(TM) CPU 3.20GHz @ 3192.250 MHz	40	4000
19	RI_Cluster	Харків	10 Мбіт/с	Scientific Linux 5.2	Intel(R) Xeon(R) CPU X5355 @ 2.66GHz	24	6384
20	ISMA_Cluster	Харків	10 Мбіт/с	CentOS 5.1	Intel(R) Xeon(R) CPU 2.33GHz	88	5320
21	IRE_Cluster	Харків	10 Мбіт/с	Linux-2.6.27. 7-smp, glibc- 2.16.6 Slack- ware-12.2.0	Intel(R) Xeon(R) CPU E 5420 @ 2.50GHz	16	4660
22	ILTPE_Cluster	Харків	10 Мбіт/с	CentOS-5.2	Intel(R) Xeon(TM) CPU 2.83GHz	88	5000

В першому експерименті для тестування використовується 100 задач, що мають змінну довжину середнього значення MI. Задачі оброблялися на наступних ресурсах (табл.2): KPI Cluster, IMBG Cluster і групі з 3-х кластерів (BITP Cluster, DFTI Cluster, IAP Cluster). Спочатку завдання поступали на кластери безпосередньо в оригінальній послідовності, вказаній у файлі завантаження, тобто для їх розподілу не використовувався брокер. При цьому на кожному кластері використовувалася евристика "перший

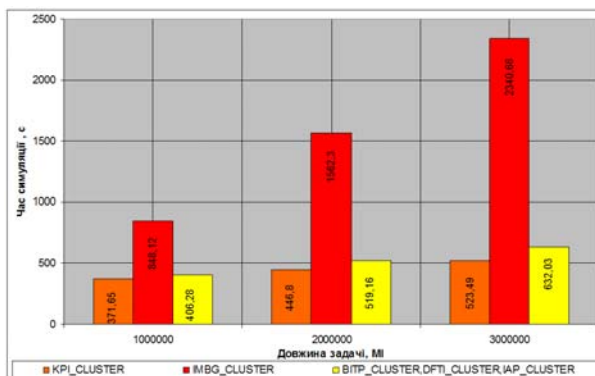
відповідно". Завдання, що знаходяться в черзі на кластері, перебираються в тому порядку, в якому вони поступили. Якщо завдання можливо в даний момент розмістити на кластері, то вона розміщувалася. Якщо процесори зайняті, то завдання поділяє процесорний час з задачею, якій потрібно саме більше часу на виконання. В табл.3 і 4 приведені результати моделювання, а на рис.2 показані графіки залежності часу моделювання від довжини задач.

Таблиця 3. Залежність часу моделювання від довжини задач при 100 задачах без використання брокера

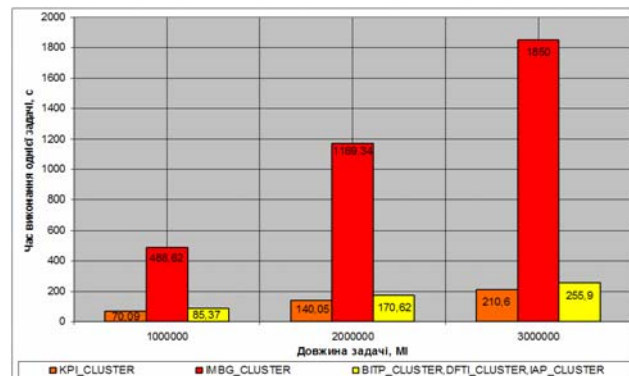
Розмір задачі, MI	KPI_CLUSTER		IMBG_CLUSTER		BITP_CLUSTER DFTI_CLUSTER IAP_CLUSTER	
	Час виконання,с	Час виконання однієї задачі, с	Час виконання,с	Час виконання однієї задачі, с	Час виконання,с	Час виконання однієї задачі, с
1000000	891,96 ≈ 14,86 хв.	168,22 ≈ 2,8 хв.	2035,51 ≈ 33,92 хв.	1172,69 ≈ 19,54 хв.	975,09 ≈ 16,25 хв.	204,9 ≈ 3,41 хв.
2000000	1072,33 ≈ 17,87 хв.	336,12 ≈ 5,6 хв.	3797,53 ≈ 63,29 хв.	2806,43 ≈ 46,77 хв.	1245,99 ≈ 20,76 хв.	409,51 ≈ 6,82 хв.
3000000	1256,39 ≈ 20,93 хв.	504 ≈ 8,4 хв.	5617,59 ≈ 93,62 хв.	4440,02 ≈ 74 хв.	1516,89 ≈ 25,28 хв.	614,17 ≈ 10,23 хв.

Таблиця 4. Залежність часу моделювання від довжини задач при 100 задачах з використанням брокера

Розмір задачі, MI	KPI_CLUSTER		IMBG_CLUSTER		BITP_CLUSTER DFTI_CLUSTER IAP_CLUSTER	
	Час виконання,с	Час виконання однієї задачі, с	Час виконання,с	Час виконання однієї задачі, с	Час виконання,с	Час виконання однієї задачі, с
1000000	371,65 ≈ 6,19 хв.	70,09 ≈ 1,16 хв.	848,12 ≈ 14,13 хв.	488,62 ≈ 8,14 хв.	406,28 ≈ 6,77 хв.	85,37 ≈ 1,42 хв.
2000000	446,8 ≈ 7,44 хв.	140,05 ≈ 2,33 хв.	1562,3 ≈ 26,37 хв.	1169,34 ≈ 19,48 хв.	519,16 ≈ 8,65 хв.	170,62 ≈ 2,84 хв.
3000000	523,49 ≈ 8,72 хв.	210,6 ≈ 3,5 хв.	2340,66 ≈ 39,01 хв.	1850 ≈ 30,83 хв.	632,03 ≈ 10,53 хв.	255,9 ≈ 4,26 хв.



а



б

Рис. 2. Залежність часу моделювання від довжини задачі при 100 задачах без використання брокера (а) і з використанням брокера (б)

При порівнянні табл.3 і 4 видно, що використання брокера зменшує загальний час виконання 100 задач і час виконання однієї задачі в 2.4 рази незалежно від розміру задачі.

Під час **другого** експерименту досліджувалася залежність кількості задач від часу їх моделювання. В цьому пункті для тестування використовується змінна кількість задач від 50 до 150 є середньою довжиною задачі, що рівна 10000 MI. В табл.5 і 6 приведені результати моделювання, а на рис.3 показані графіки залежності часу моделювання від довжини задач.

Під час **третього** експерименту досліджувалася залежність середньої кількості завдань в черзі в залежності від їх загальної кількості при середньому розмірі задачі 1000000 MI для випадків використання брокера і без нього. Результати відображені в табл.7 і на рис.4.

Під час **четвертого** експерименту моделювався загальний середній час очікування завдань в черзі. Для тестування використовується кількість завдань, рівна 100 штук, розміром вхідного файлу 500 Кбайт, 1000000 MI кожна. На рис.5 показано графік залежності часу

очікування в черзі від кількості завдань. В табл.8 приведені дані, отримані при моделюванні на всіх ресурсах при різній кількості завдань.

Мета **п'ятого** експерименту полягала в оцінюванні загальної пікової кількості завдань в черзі. При цьому для тестування використовується різна кількість завдань, від 50 до 250 розміром 1000 000 MI кожна. Відповідні результати відображені в табл.9 і на рис.6.

В **шостому** експерименті досліджувалася залежність часу моделювання від пропускної здатності кожного ресурсу української грид-інфраструктури. Для тестування використовується 100 задач розміром 1000 000 MI кожна. Моделювання проходить на кожному ресурсі української грид-інфраструктури в залежності від їх пропускної здатності для випадків відсутності брокера і при його використанні. Отриманні результати наведені в табл.10 і на рис.7.

Для кластерів ICUB_SCYT-3_CLUSTER, KPI_CLUSTER та BITP_CLUSTER було взято пропускну здатність каналу рівну 100 Мбіт/с, для всіх інших було прийнято, що пропускну здатність каналу рівна 10 Мбіт/с.

Таблиця 5. Залежність часу моделювання від кількості задач без використання брокера

Розмір задачі, MI	KPI_CLUSTER		IMBG_CLUSTER		BITP_CLUSTER DFTI_CLUSTER IAP_CLUSTER	
	Час виконання, с	Час виконання однієї задачі, с	Час виконання, с	Час виконання однієї задачі, с	Час виконання, с	Час виконання однієї задачі, с
50	536,06 с ≈ 8,93 хв.	167,04 с ≈ 2,78 хв.	1112,87 с ≈ 18,54 хв.	656,64с ≈ 10,9 хв.	606,37 с ≈ 10,1 хв.	203,86с ≈ 3,39 хв.
100	891,96 с ≈ 14,8 хв.	168,22 с ≈ 2,80 хв.	2035,51 с ≈ 33,9 хв.	1172,69с ≈ 19,54 хв.	975,09 с ≈ 16,25 хв.	204,9 с ≈ 3,41 хв.
150	1249,74 с ≈ 20,8 хв.	168,38 с ≈ 2,806хв.	3035,98 с ≈ 50,59 хв.	1691,53с ≈ 28,19 хв.	1334,1 с ≈ 22,23 хв.	205,5 с ≈ 3,42 хв.

Таблиця 6. Залежність часу моделювання від від кількості задач з використанням брокера

Розмір задачі, MI	KPI_CLUSTER		IMBG_CLUSTER		BITP_CLUSTER DFTI_CLUSTER IAP_CLUSTER	
	Час виконання, с	Час виконання однієї задачі, с	Час виконання, с		Час виконання, с	Час виконання однієї задачі, с
50	223,35с ≈ 3,72 хв.	69,6 с ≈ 1,16 хв.	463,69 с ≈ 7,72 хв.	273,6 с ≈ 4,56 хв.	252,65 с ≈ 4,21 хв.	84,94 с ≈ 1,41 хв.
100	371,65 с ≈ 6,19 хв.	70,09 с ≈ 1,168 хв.	848,12 с ≈ 14,13 хв.	488,62 с ≈ 8,14 хв.	406,28 с ≈ 6,77 хв.	85,37 с ≈ 1,42 хв.
150	520,72 с ≈ 8,67 хв.	70,15 с ≈ 1,169 хв.	1264,99 с ≈ 21,08 хв.	704,8 с ≈ 11,74 хв.	555,87 с ≈ 9,26 хв.	85,62 с ≈ 1,42 хв.

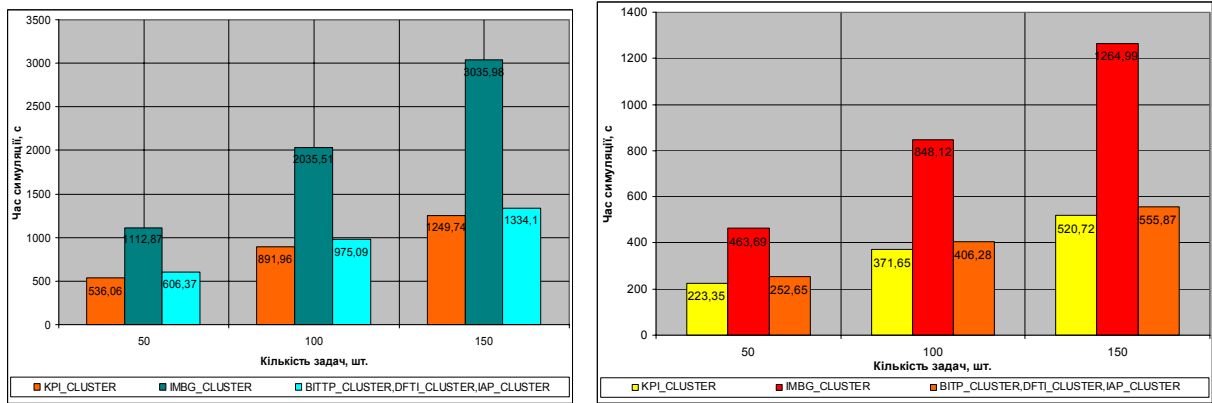


Рис. 3. Залежність часу моделювання від кількості задач без використання брокеру (а) і з використанням брокеру (б)

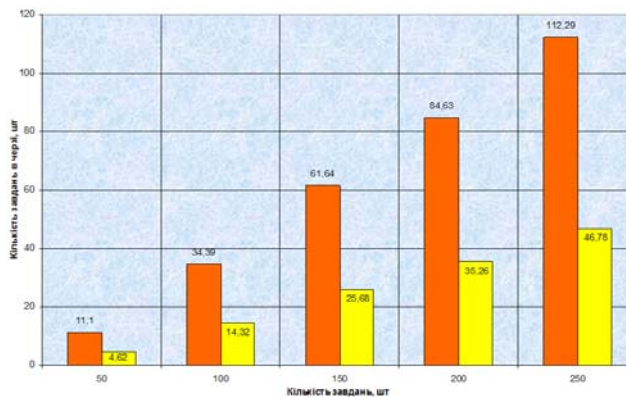


Рис. 4. Графік залежності середньої кількості завдань в черзі без використання брокеру і з брокером при різній кількості завдань

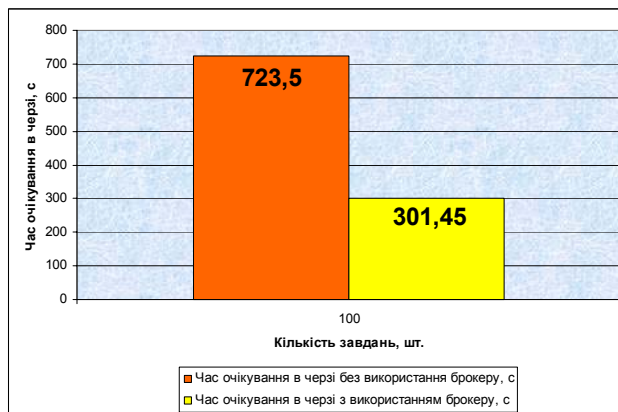


Рис. 5. Графік залежності часу очікування завдань в черзі від кількості завдань

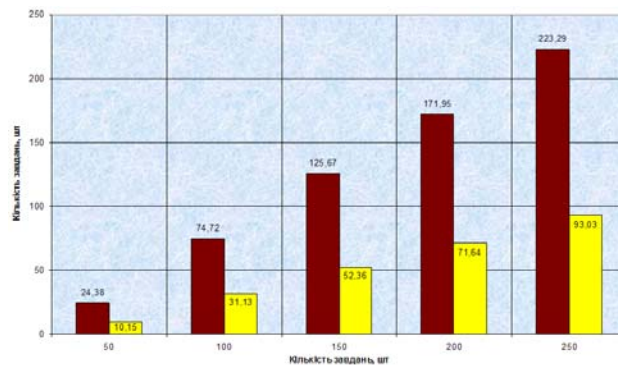


Рис. 6. Графік залежності пікової кількості завдань в черзі без використання брокеру і з брокером при різній кількості завдань

Таблиця 7. Залежності середньої кількості завдань в черзі при різній кількості завдань

Кількість завдань, шт.	Середня кількість завдань в черзі без використання брокера, шт.	Середня кількість завдань в черзі з використанням брокера, шт.
50	11,1	4,62
100	34,39	14,32
150	61,64	25,68
200	84,63	35,26
250	112,29	46,78

Таблиця 8. Залежність часу очікування завдань в черзі від кількості завдань

Кількість завдань, шт.	Час очікування в черзі без використання брокера, с		Час очікування в черзі з використанням брокера, с	
	с	хв.	с	хв.
100	723,5	12,05	301,45	5,02

Таблиця 9. Залежності пікової кількості завдань в черзі при різній кількості завдань

Кількість завдань	Пікова кількість завдань в черзі без використання брокера, шт.	Пікова кількість завдань в черзі з використанням брокера, шт.
50	24,38	10,15
100	74,72	31,13
150	125,67	52,36
200	171,95	71,64
250	223,29	93,03

Таблиця 10. Залежність часу моделювання від пропускної здатності каналу кожного ресурсу

Назва ресурсу	Реальна пропускна здатність каналу, Мбіт/с	Час моделювання без використання брокера, с		Час моделювання з використанням брокера, с	
ICYB_SCYT-3_CLUSTER	100 Мбіт/с	1626,13	27,10	669,19	11,15
KPI_CLUSTER	100 Мбіт/с	1607,19	26,78	661,39	11,02
BITP_CLUSTER	100 Мбіт/с	1693,02	28,21	696,71	11,61
DFTI_CLUSTER	10 Мбіт/с	1654,75	27,57	680,96	11,34
IAP_CLUSTER	10 Мбіт/с	1665,9	27,76	685,55	11,42
ILTPE_CLUSTER	10 Мбіт/с	1615,50	26,92	664,81	11,08
ILTPE_GRID_CLUSTER	10 Мбіт/с	4793,85	79,89	1972,77	32,87
IMMSP_CLUSTER	10 Мбіт/с	1827,47	30,45	752,04	12,53
IMP_CLUSTER	10 Мбіт/с	1710,92	28,51	704,08	11,73
Inparcom_CLUSTER	10 Мбіт/с	1655,52	27,59	681,28	11,35
IOP_CLUSTER	10 Мбіт/с	1676,57	27,94	689,94	11,49
IRE_CLUSTER	10 Мбіт/с	1640,51	27,34	675,11	11,25
ISMA_CLUSTER	10 Мбіт/с	1655,52	27,59	681,28	11,35
ISOFTS_CLUSTER	10 Мбіт/с	1616,54	26,94	665,24	11,08
KMA_GRID_CLUSTER	10 Мбіт/с	1616,54	26,94	665,24	11,08
KNU_ARC_CLUSTER	10 Мбіт/с	1626,62	27,11	669,39	11,15
MAO_CLUSTER	10 Мбіт/с	1645,74	27,42	677,25	11,28
PIMEE_CLUSTER	10 Мбіт/с	1692,57	28,20	696,53	11,60
SRI_CLUSTER	10 Мбіт/с	1660,15	27,66	683,19	11,38
THEI_CLUSTER	10 Мбіт/с	5905,61	98,42	2430,29	40,50
IMBG_CLUSTER	10 Мбіт/с	2244,93	37,41	923,83	15,39
RI_CLUSTER	10 Мбіт/с	1626,63	27,11	669,39	11,15
ICMP_CLUSTER	10 Мбіт/с	1692,036	28,20	696,31	11,60

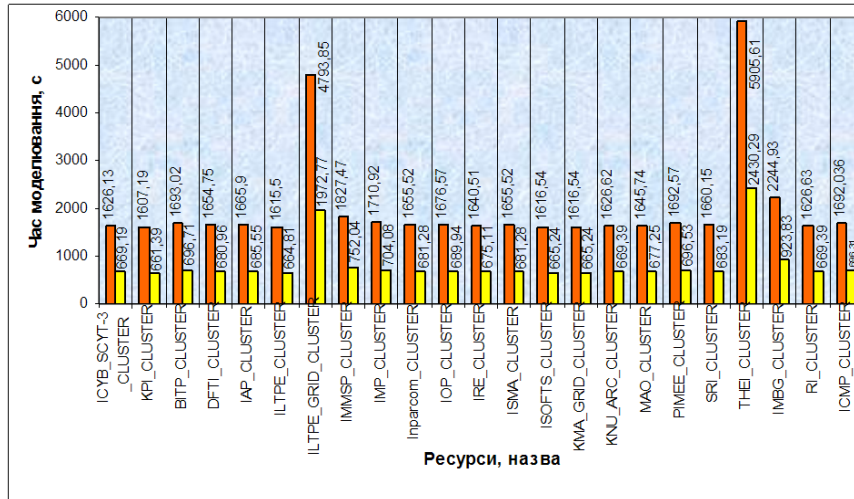


Рис.7. Залежність часу моделювання від пропускну здатності каналу кожного ресурсу

Висновки

В роботі проілюстровані можливості програмного інструменту моделювання GridSim, призначеного для моделювання поведінки різних типів гід-систем в умовах зростаючого потоку завдань, які одночасно вирішуються, і, зокрема, для дослідження алгоритмів планування і використання ресурсів в гід-системах. Проведені комп'ютерні експерименти з моделювання параметрів національної української гід-інфраструктури в умовах відсутності центрального планувальника (брокера), як то притаманне використанню проміжного шару NorduGrid, та при наявності механізму розподілу завдань серед вільних на певний час обчислювальних ресурсів, як то характерне при використанні проміжного шару gLite.

Обчислювальні експерименти на моделі гід-інфраструктури показують, що при наявності брокера уповільнюються в 2.4 рази залежності часу моделювання і часу виконання задач від довжини задачі і кількості задач, що вирішуються одночасно, зменшується середній час затримки задач в черзі, а також середня кількість і пікова кількість завдань в черзі. Додатковим фактором, що суттєво впливає на показники оброблення завдань в гід-інфраструктурі, як показали експерименти, є швидкість каналу передачі (пропускну здатність каналу) кожного з її обчислювальних вузлів. При цьому більш швидкий канал забезпечує більшу працездатність гід-інфраструктури.

Створений стенд моделювання гід-систем дозволить у найближчому майбутньому провести порівняльне дослідження різних евристик розподілу завдань брокером і обґрунтувати вибір найкращою з них для впровадження в українській гід-інфраструктурі.

Автор дякує студентам Єлусеву Щ. і Куцу В. за допомогу в проведенні обчислювальних експериментів.

Література

1. Згуровський М.З., Петренко А.І. Grid-технології для науки і освіти. // Наукові вісті НТУУ «КПІ». — №2. — 2009. — с.10-17.
2. Zagrodny A., Zgurovsky M., Zinovjev G., Martynov E., Petrenko A. Integrating Ukraine into European Grid Infrastructure. // Системні дослідження і інформаційні технології. — №2. — 2009. — с.35-49.
3. <http://www.nordugrid.org/middleware/>
4. <http://glite.web.cern.ch/glite/>
5. Торопков В.В. Модели распределенных вычислений. — М.: ФИЗМАТЛИТ, 2004. — 320 с.
6. <http://www.is.ocha.ac.jp/~takefusa/bricks>
7. <http://www.wcsag.ucsd.edu/projects/grid/microgrid.html>
8. <http://edg-wp2.web.cern.ch/edg-wp2/optimization/optorsim.html>
9. <http://gcl.ucsd.edu/simgrid>
10. <http://www.gridbus.org/gridsim/>
11. <http://www.gridbus.org/>
12. <http://hipersoft.cs.rice.edu/>
13. <http://www.cs.wisc.edu/condor/>