

УДК 004.45 : 681.324

Корочків О.В., Макаров Л.О., Фісун М.Т.

## Про створення GRID-системи у середовищі ALCHEMI

У статті розглянуті питання створення GRID-системи на технічній базі комп’ютерного класу вищого навчального закладу. Проаналізовані особливості, переваги та недоліки різних підходів до реалізації розподілених обчислень. Показано можливість створення GRID, яка може використовуватися як для розв’язання складних задач, що потребують значних обчислювальних ресурсів, так і для здійснення навчального процесу за цією тематикою.

The article considers questions of Grid-system creation based on the technical base of a higher educational establishment computer class. Features, advantages and lacks of different approaches to distributed calculations implementation are analyzed. Grid creation possibility is shown, which can be utilized to resolve intricate tasks requiring much computational resources, but also to realize an educational process based on the subject.

### Загальна постановка проблеми та її зв’язок з науково-практичними задачами.

Як відомо, обчислювальна техніка у своєму розвитку на шляху підвищення швидкодії ЕОМ наблизилася до фізичних меж. Час переключення електронних схем досяг часток наносекунди, а швидкість поширення сигналів у лініях, які зв’язують елементи й вузли машини, обмежена значенням 30 см/нс (швидкістю світла). Тому подальше зменшення часу переключення електронних схем не дозволить істотно підвищити продуктивність ЕОМ і тому подальше підвищення їх швидкодії може бути досягнути тільки шляхом поширення принципу паралелізму на самі пристрой обробки інформації та шляхом створення багатомашинних і багатопроцесорних обчислювальних систем [1]. Такі системи дозволяють робити розпаралелювання виконання програми або паралельне виконання декількох програм.

Вищі навчальні заклади, що готують фахівців за комп’ютерним напрямом, мають надавати відповідні навчальні послуги, щоб випускники їхніх ВНЗ отримали відповідні знання та здобули необхідні уміння щодо розподілених обчислень.

Багатопроцесорні системи, які раніше асоціювалися в основному із суперкомп’ютерами, у цей час міцно затвердилися у всьому діапазоні вироблюваних обчислювальних систем, починаючи від персональних комп’ютерів і закінчуючи суперкомп’ютерами на базі векторно-конвеєрних процесорів. Ця обставина, з одного боку, збільшує доступність суперкомп’ютерних технологій, а з іншого, підвищує актуальність їхнього освоєння, оскільки для всіх типів багатопроцесорних систем потрібне використання спеціальних технологій програмування для того, щоб програма могла повною мірою використовувати ресурси високопродуктивної обчислювальної системи. Тому створення програмно-технічного середовища для організації навчального процесу з дисциплін, де необхідно засвоювати або використовувати розподілені обчислення, є актуальним.

Метою даного дослідження є аналіз різних підходів щодо підвищення продуктивності обчислювальних систем та вибір одного із шляхів реалізації комп'ютерної системи розпаралелювання обчислень у вищих навчальних закладах, що не мають великих обчислювальних потужностей у вигляді суперкомп'ютерів або кластерних систем по кілька десятків комп'ютерів в одному кластері.

### Аналіз підходів підвищення продуктивності обчислювальних систем

**Суперкомп'ютери.** Відмінною рисою суперкомп'ютерів є те, що це, як правило, надзвичайно дорогі багатопроцесорні системи. Обчислювальна система називається багатопроцесорною, якщо вона містить кілька процесорів, що працюють із загальним полем оперативної пам'яті і керується однією загальною операційною системою [2].

**Кластерні системи.** В останні роки широко використовуються в усьому світі як дешева альтернатива суперкомп'ютерам. Кластер є багатомашинною обчислювальною системою – система, яка містить ЕОМ, кожна з яких має свою оперативну пам'ять і працює під управлінням своєї операційної системи, а також засобу обміну інформацією між машинами. Реалізація обміну інформацією відбувається, в кінцевому підсумку, шляхом взаємодії операційних систем машин між собою [2].

Ще одним напрямком розпаралелення обчислень є створення так званих **Grid-систем**, під якими розуміють "погоджене, відкрите й стандартизоване середовище, що забезпечує гнучкий, безпечний, скординований поділ ресурсів у рамках віртуальної організації" [3]. Такі системи не прийнято вважати кластерами, але їхні принципи в значній мірі подібні до кластерної технології. Головна відмінність – низька доступність кожного вузла, тобто неможливість гарантувати його роботу в заданий момент часу (вузли підключаються й відключаються в процесі роботи), тому завдання повинно бути розбито на ряд незалежних один від іншого процесів.

Grid-системи гармонійно доповнюють ряд обчислювальних архітектур, використовуваних сьогодні. Застосування технологій Grid дозволяє будувати систему управління розподіленими обчислювальними ресурсами. У такій ситуації користувачеві вже не важливо, на якому конкретному вузлі мережі виконується його завдання; він просто споживає певну кількість віртуальної процесорної потужності, наявної в мережі [4]. В Grid час взаємодії між вузлами вимірюється мілісекундами й секундами, тому такі системи не призначені для розв'язання паралельних задач, а націлені здебільшого на розв'язання пакетних завдань, коли кожне окреме завдання виконується на одному вузлі.

На сьогоднішній день для спрощення процесу організації й управління розподіленими обчисленнями створена велика кількість програмних комплексів, як комерційних, так і абсолютно безкоштовних.

При виборі платформи для побудови grid-системи в Миколаївському державному гуманітарному університеті (МДГУ) ім. Петра Могили порівнювалися платформи, які мають вільну ліцензію на розповсюдження та використання, а також дві комерційні платформи, які мають обмежену вільну ліцензію [6-13]. Як критерій до уваги бралися ознаки, наведені у табл. 1.

На основі виконаного аналізу перелічених програмних продуктів як програмну платформу grid-системи обрано середовище Alchemi. Вибір обґрунтovується:

- вільною ліцензією на розповсюдження та використання;
- відкритістю коду, що дозволяє модифікувати та вдосконалювати платформу власноруч;

- наявністю можливості поєднання окремих систем в рамках платформи у ієрархічну кластерну систему;
- платформа реалізації та використання – ОС MS Windows та мова C# на базі платформи .NET Framework, що значно спрощує розробку і впровадження grid-системи та майбутніх grid-додатків для найбільш розповсюдженій операційної системи у світі (адже ОС MS Windows встановлено на близько 90% користувальників комп’ютерів).

Таблиця 1

## Зведенна таблиця характеристик grid-платформ

<b>Платформа \ Властивість</b>	<b>BOINC</b>	<b>Condor</b>	<b>Globus</b>	<b>Alchemi</b>	<b>Didigipede</b>	<b>Sun Grid Engine</b>
Архітектура	Центра-лізованна	Ієрархіч-на	Центра-лізованна	Ієрархіч-на	Централізо-вана	Центра-лізованна
Технології реалізації	C++, win32	C	C and Java	C#, .NET	C#, .NET	Java
Операційна система	Клієнт – всі; сервер – всі	Unix	Клієнт – всі; сервер – Unix	Win32	Win32	Всі
Об’єднання кластерів	Hi	Так	Hi	Так	Hi	Hi
Вільна ліцензія	Так	Так	Так	Так	В академіч-них цілях	Так
Відкритий код	Так	Так	Так	Так	Hi	Hi

**Опис програмної платформи розроблюваної Grid-системи**

Як і інші grid-платформи, Alchemi віртуалізує ресурси багатьох обчислювальних вузлів, утворюючи великий мультипроцесорний віртуальний комп’ютер.

Аналогії між традиційною мультипроцесорною парадигмою програмування і парадигмою grid-програмування [5] наведено у табл. 2.

Таким чином, grid-система Alchemi може розглядатися як віртуальний комп’ютер з великою кількістю процесорів. Grid-додаток може скористатися цим і створювати незалежні блоки завдань, які будуть виконуватися паралельно на grid ( кожен блок буде виконуватися на певному виконуючому вузлі).

Підтримка виконання додатка одночасно на багатьох процесорах (нехай і віртуальних) потребує використання певних навичок паралельного програмування.

Платформа Alchemi підтримує дві моделі створення паралельних додатків:

- модель пакетних grid-задач;
- об’єктно-орієнтована модель grid-потоків.

**Модель пакетних grid-задач** є класичною моделлю для розподілених обчислень.

Це високорівнева абстракція віртуальної машини, в якій найменшим блоком паралельного виконання є процес. Специфікація задачі для виконання у grid-системі складається головним чином із входних файлів, вихідних файлів та додатка для виконання (процесу).

Таблиця 2  
Мультипроцесорна та grid-парадигма програмування

Мультипроцесорний комп'ютер	Grid
Операційна система	Платформа Alchemi
Процесор	Виконуючий вузол (Executer)
Сервіси ОС (високого рівня) – Системний програмний інтерфейс (System API)	Програмний інтерфейс (API) платформи Alchemi
Сервіси ОС (низького рівня)	Керуючий вузол (Manager)
Процес	Grid-додаток
Потік	Grid-потік

З іншого боку, об'єктно-орієнтована модель grid-потоків є основною програмною моделлю для платформи Alchemi. Вона пропонує більш низькорівневу (отже, більш потужну) абстракцію grid-системи, надаючи об'єктно-орієнтовану програмну модель, яка імітує страдиційне мульти-потокове програмування. Найменшим блоком паралельного виконання у даному випадку є grid-потік (об'єкт .NET), який є програмним аналогом звичайного потоку.

Але для розробника grid-додатків усі фікції створення та управління grid-потоків зібрано у дуже зручному програмному інтерфейсі платформи, що дозволяє зосередитись на розробці саме додатка, а не замислюватись над питаннями управління потоками. Більше того, для розробника немає різниці між локальним потоком та віддаленим grid-потоком, адже управління віддаленими потоками grid-платформа бере на себе.

Апаратний рівень розробленої системи схематично представлено на рис. 3.

### Типи вузлів платформи Alchemi

Платформа Alchemi складається з трьох основних розподілених компонентів (вузлів): керуючий вузол; обчислювальний вузол та користувач. Для створення робочої Grid-системи на базі платформи Alchemi необхідно встановити та налагодити всі три типи вузлів.

**Керуючий вузол** було встановлено як системну службу на сервер додатків на базі процесора AMD Athlon 64 3000+ (1.8ГГц) з обсягом оперативної пам'яті 1ГБ під управлінням ОС Windows Server 2003 R2 (32 біт). Він безпосередньо підключений до головного комутатора, що дозволить знизити можливі затримки при роботі grid-системи, адже керуючий вузол є центральним елементом у архітектурі grid-системи.

**Обчислювальні вузли** встановлювались у класі комп'ютерної графіки університету на персональні комп'ютери на базі процесора Intel Celeron 2,2ГГц з обсягом оперативної пам'яті 256МБ під управлінням ОС Windows XP SP2 (32 біт). Початкова кількість встановлених вузлів дорівнює п'яти. Обчислювальні вузли встановлювались як системна служба.

**Як користувальський вузол** було використано робочу станцію на базі процесора AMD Athlon XP 2000+ (1.3ГГц) з обсягом оперативної пам'яті 512МБ під управлінням ОС Windows XP SP2 (32 біт). На робочому вузлі немає необхідності встановлювати додаткове програмне забезпечення, адже на ньому лише виконується клієнтський grid-

додаток. Клієнтський додаток може бути запущено на будь-якому комп'ютері під управлінням ОС Windows локальної мережі.

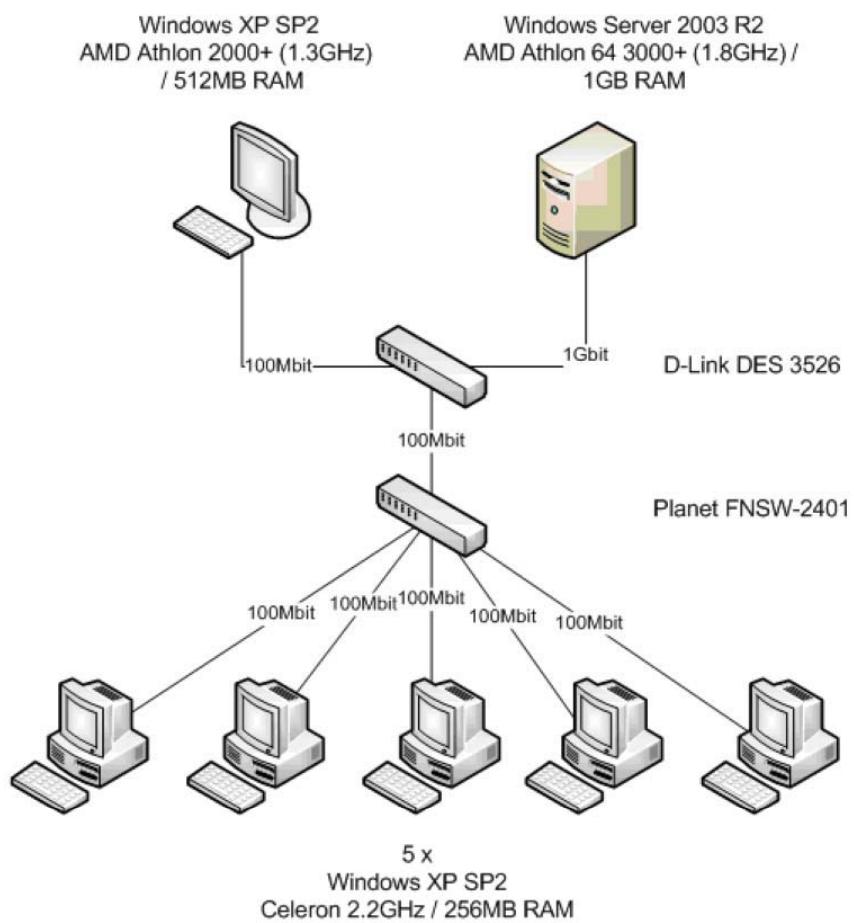


Рис. 3. Апаратні засоби розробленої grid-системи

Таким чином, було отримано функціонуючу grid-систему, яка поєднує ресурси п'яти персональних комп'ютерів комп'ютерного класу. Необхідно провести випробування системи щодо виконання у ній grid-додатків, а також провести порівняльні тести продуктивності обчислень з використанням розробленої grid-системи

#### Тестування обчислювальної продуктивності grid-системи

Для тестування було обрано 5 комп'ютерів з однаковою конфігурацією – Intel Celeron-2200MHz/256 МВ. Тестування здійснювалося під час проведення пари у комп'ютерному класі, отже, підключенні вузли використовувалися у звичайному офісному режимі із середнім навантаженням не більше 10%.

Для тестування обчислювальної продуктивності grid-системи будемо обчислювати число Рі з точністю до 1000 знаків. Ця задача є достатньо обчислювально складною і підходить для проведення тестів. Після запуску обчислення отримуємо таку картину на моніторі ресурсів grid-системи (рис. 4).

Усі 100 потоків були обраховані у Grid-системі та отримано результат. Час виконання обчислень становив 50 секунд.

Тестування було проведено для різної кількості підключених до grid-системи вузлів (1, 2, 3, 4, 5, 10). У табл. 5 наведено зведені результати щодо тестування

залежності продуктивності grid-системи від кількості підключених вузлів.

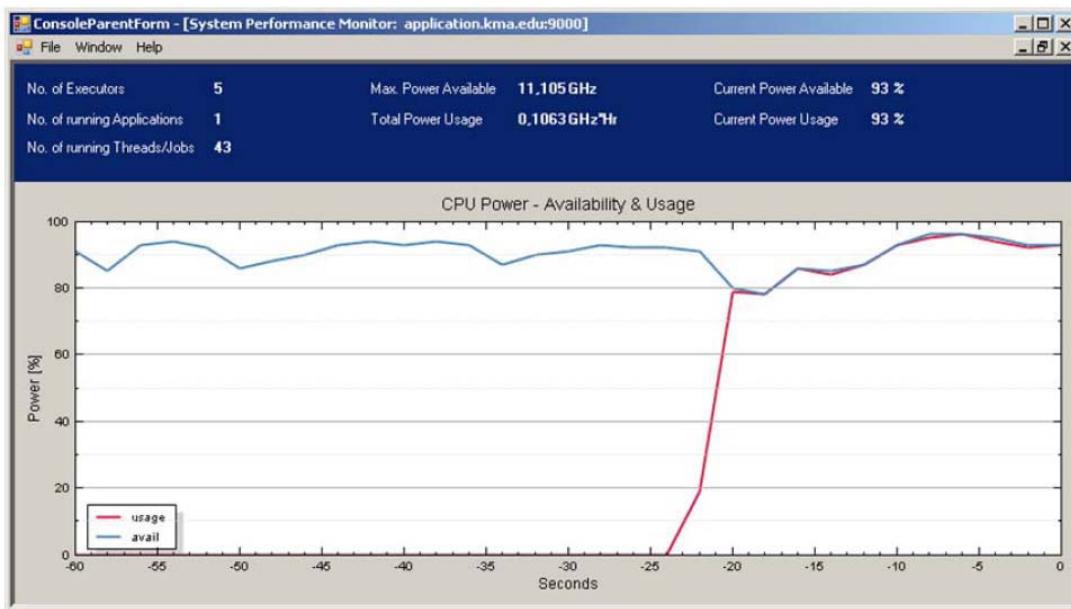


Рис. 4. Монітор ресурсів grid-системи. Доступні та використовувані ресурси Grid

З рис. 6. можна побачити, що залежність часу виконання обчислень від сумарної потужності вузлів grid-системи є ступеневою ( $y=427,1x^{-0,89}$ ) зі ступенем довіри до апроксимації  $R^2=0.993$

Таблиця. 5

Зведенна таблиця результатів тестування grid-системи

Показник \ № Експерименту	1	2	3	4	5	6
Кількість вузлів системи, шт	1	2	3	4	5	10
Сумарна можлива потужність, ГГц	2,221	4,442	6,663	8,884	11,105	24,6
Доступна потужність, %	100	100	99	100	90	96
Час виконання, сек	215	120	74	56	50	26

При тестуванні були виявлені деякі проблеми. Як правило, результат обчислень отримується після закінчення обчислень останнього grid-потоку. Це означає, що у випадку затримки якогось з потоків обчислень загальний результат також буде затримано. При проведенні експерименту з десятьма вузлами у grid-системі сталося так, що потік №9 затримав загальне завершення розрахунків до двох хвилин.

**Створення програмного додатка для перекодування цифрових аудіоматеріалів з використанням обчислювальних потужностей розробленої grid-системи.** В рамках практичного застосування розробленої grid-системи було розроблено також програмний додаток для перекодування цифрових аудіоматеріалів. Перекодування цифрової аудіоінформації – ресурсомісткий процес, який вимагає порівняно багато часу при залученні обчислювальних потужностей лише одного комп’ютера. Отже було вирішено спробувати задіяти потужності grid-системи для скорочення часу, необхідного для виконання задач такого класу.

Додаток дозволяє виконувати перекодування з дуже поширеного (та де-факто

стандартного сьогодні) формату mp3 до молодого альтернативного формату ogg vorbis, який має відкритий код. Обґрунтування доцільності перекодування пояснюється тим, що формат ogg vorbis дозволяє отримати таку ж якість звукового матеріалу, як і mp3, але при значно менших значеннях величини цифрового потоку, а отже, і меншого розміру вихідного файла завдяки використанню нових методів стискування та психоакустичних алгоритмів.

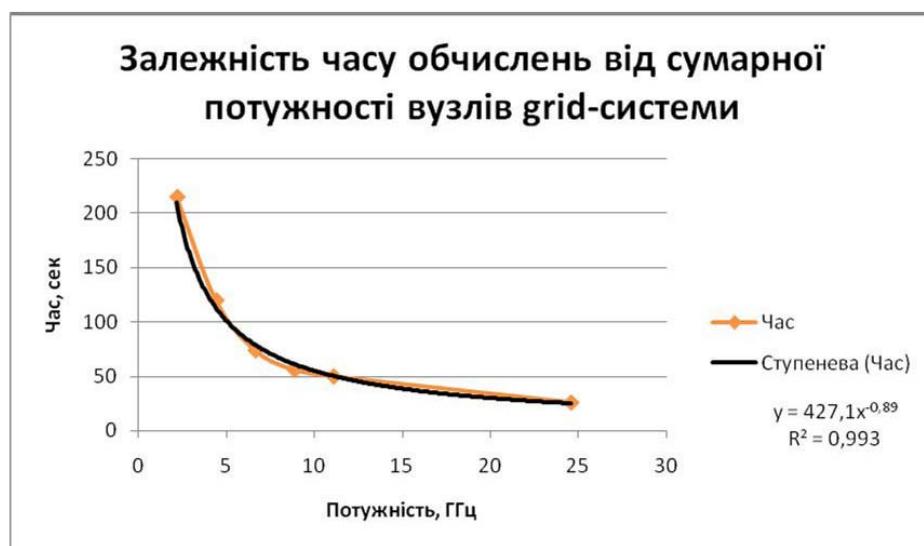


Рис. 6. Залежність часу обчислень від сумарної потужності вузлів grid-системи

Додаток використовує ресурси grid-системи у режимі пакетних задач. Для кожного файла утворюється власний grid-потік, який виконується на окремому grid-вузлі системи. Отже, додаток може максимально ефективно задіяти ресурси grid-системи у випадку, коли кількість grid-потоків (вхідних файлів) дорівнює кількості обчислювальних вузлів grid-системи або є більшою за неї.

#### Висновки щодо побудованої grid-системи та результатів тестування

Побудована grid-система на базі grid-платформи Alchemi дозволила об'єднати ресурси п'яти персональних комп'ютерів одного з комп'ютерних класів університету. Система дозволила віртуалізувати ресурси окремих вузлів та створити віртуальну мультипроцесорну обчислювальну систему, яку можна використовувати для виконання ресурсомістких обчислень. Як довели експериментальні спостереження, середнє завантаження персональних комп'ютерів під час звичайного офісного використання не перевищує 25%. Отже, залишається незадіянним близько 75% потужності вузлів. Ця потужність і використовується grid-системою для проведення обчислень.

Під час тестування була виявлена ступенева залежність часу виконання обчислень від сумарної теоретичної потужності вузлів grid-системи. Тобто додавання нових вузлів у систему не призводить до пропорційного підвищення потужності та скорочення часу обчислень, адже кожен новий вузол надає лише вільні ресурси, які складають 75%-100% від його теоретичної можливої потужності.

#### Література

1. Масич Г.Ф. "Многомашинные и многопроцессорные вычислительные системы".
2. Дацюк В.Н., Букатов А.А., Жегуло А.И. Методическое пособие по курсу "Многопроцессорные системы и параллельное программирование", Ростовский государственный университет.

3. Ian Foster, Carl Kesselman. The Grid. Blueprint for a new computing infrastructure. Morgan Kaufman, 1998.
4. Виктор Коваленко, Дмитрий Корягин. – “ Организация grid: есть ли альтернативы?” // Открытые системы. – № 12/2004.
5. Krishna Nadiminti, Akshay Luther, Rajkumar Buyya, Alchemi: A .NET-based Enterprise Grid System and Framework. User Guide for Alchemi 1.0, 2005.
6. Матеріали інтернет-ресурсу <http://wikipedia.org>.
7. Матеріали інтернет-ресурсу <http://www.boinc.ru/>.
8. Матеріали інтернет-ресурсу <http://www.cs.wisc.edu/condor/>.
9. Матеріали інтернет-ресурсу <http://gridclub.ru/>.
10. Матеріали інтернет-ресурсу <http://www.globus.org/>.
11. Матеріали інтернет-ресурсу <http://www.alchemi.net/>.
12. Матеріали інтернет-ресурсу <http://www.digipede.net/>.
13. Матеріали інтернет-ресурсу <http://sun.com/gridware/>.