

В.В. ХРАМОВ, О.О. СУДАКОВ, М.В. КОНОНОВ

ПЕРШИЙ КРИМСЬКИЙ ЕКСПЕРИМЕНТАЛЬНИЙ ОБЧИСЛЮВАЛЬНИЙ GRID-РЕСУРС: СТВОРЕННЯ І ПЕРШІ РЕЗУЛЬТАТИ ВИПРОБУВАНЬ І ЕКСПЛУАТАЦІЇ

***Анотація.** У статті представлено специфіку основних проблем застосування комп'ютерного моделювання в гуманітарних і екологічних дослідженнях. Описані переваги використання Grid-кластера для паралельної обробки інформації у вищевказаних галузях. Обґрунтовується вибір стратегії для створення експериментального обчислювального Grid-кластера. Експериментально доводиться, що вибраний підхід з застосуванням кластерних технологій та методів паралельної обробки інформації є ефективним для цих задач. Наведено особливості побудови і перші результати використання обчислювального Grid-кластера на основі запропонованої стратегії.*

***Ключові слова:** Grid, кластер, проміжне програмне забезпечення.*

***Аннотация.** В статье представлена специфика основных проблем применения компьютерного моделирования в гуманитарных и экологических исследованиях. Описаны преимущества использования Grid-кластера для параллельной обработки информации в вышеуказанных отраслях. Обосновывается выбор стратегии для создания экспериментального вычислительного Grid-кластера. Экспериментально доказывается, что предложенный подход с применением кластерных технологий и методов параллельной обработки информации является эффективным для задач данного класса. Приведены особенности построения и первые результаты использования вычислительного Grid-кластера на основе предложенной стратегии.*

***Ключевые слова:** Grid, кластер, промежуточное программное обеспечение.*

***Abstract.** Specifics of main problems of computer modeling in humanitarian and environmental studies is presented. Advantages of using Grid-cluster for parallel information processing in these areas are described. The choice of a strategy for the creation of a pilot computing Grid-cluster is substantiated. The proposed approach using cluster technologies and parallel information processing methods is experimentally proven to be effective for the given class of problems. Peculiarities of construction and first usage results of the computational Grid-cluster based on the proposed strategy are revealed.*

***Key words:** Grid, cluster, middleware.*

1. Вступ

Метою проекту побудови першого Кримського експериментального Grid-кластера є необхідність вирішення актуальних завдань у галузі екології, соціології, економіки. До таких задач належить, зокрема, прогнозування лісових пожеж і їх запобігання. Особливо актуальною є проблема оцінки інтенсивності пожеж і зв'язку частоти їх появи з певними причинами. Важливою задачею є оцінка інтенсивності забруднень води на виході з очисних споруд у зв'язку з об'ємом води, що очищається, і пропускною спроможністю очисних споруд. Потребує вирішення проблема аналізу погодних умов з прогнозуванням таких небезпечних явищ, як зливові дощі, атмосферні вихори, смерчі, шторми та ін. З цими задачами пов'язана комп'ютерна картографія: розробка і впровадження оперативних способів відображення екологічної інформації на картах. До цієї галузі належать задачі економічного прогнозування, моделювання ринку, аналізу соціологічних досліджень та ін.

У Таврійському гуманітарно-екологічному інституті у місті Сімферополі створено матеріальну базу і у них є досвід експериментальних та теоретичних досліджень у галузі гуманітарних наук, проводяться наукові конференції і симпозіуми, функціонує комп'ютерний центр з можливістю виходу в Інтернет, розвинена інфраструктура локальної мережі з великою кількістю ресурсів: інформаційні портали й бібліотеки, інтелектуальні бази даних і форуми, поштова система.

Наприкінці 2008 року в ТГЕІ був запущений перший Кримський експериментальний Grid-ресурс [1] (<http://cluster.crimea.ua>), підключений до Українського академічного сегмента Grid (<http://uag.bitp.kiev.ua>). В Українському академічному сегменті центром сертифікації Grid (<https://www.uagrid.org>) для кластера були отримані цифрові сертифікати безпеки X.509 [2] і здійснено його підключення до Українського академічного сегмента Grid [3].

З того часу почалося його активне налагодження і тестування. Предметом статті є опис особливостей архітектури даного ресурсу і результати його тестування.

2. Приклади гуманітарно-екологічного моделювання

Прикладом досліджень у галузі екології може служити Європейський центр передбачення погоди (ECMWF) [4] у Сполученому Королівстві, який зараз обслуговує 560 активних споживачів і опрацьовує 40 000 пошукових запитів щодня, залучаючи дані з-понад 2 000 000 метеорологічних постачальників. Кожного дня додається інформація з близько 4 000 000 інших джерел, що становить приблизно 0,5 терабайт нових даних. Сховище таких даних зараз містить близько 330 терабайт. Зростання в об'ємі метеорологічних даних становить близько 80% за рік. На даний момент створено багато національних послуг щодо передбачення погоди, таких як Німецький Синоптичний Центр DWD (Deutscher Wetterdienst), які прогнозують погоду на ділянках з розміром менше 10 км, використовуючи моделі NWP, до чотирьох разів день, щоб забезпечити необхідні дані для громадськості. Виконання моделей NWP є складною обчислювальною задачею: 48-годинне передбачення утворює близько 20 Гбайт даних. Крім того, короткий прогноз погоди – це критичне завдання за часом, яке потрібно завершити менш, ніж за дві години. Тому тільки обчислювальні центри високої продуктивності (HPC) здатні виконувати такі прогнози.

Актуальним для Криму є завдання моделювання лісових пожеж. У найзагальнішому вигляді моделювання зводиться до оцінки імовірнісного зв'язку між частотою пожеж і їх причинами, а також перевірки інших статистичних гіпотез, таких як оцінка статистичних розподілів інтенсивностей пожеж у різних місцях. Такі завдання зводяться до системи диференціальних рівнянь відносно функції розподілу, математичного очікування, дисперсії і т.д. Процес моделювання включає такі етапи: 1) введення даних, 2) групування і ранжирування даних, 3) обробка даних емпіричною функцією розподілу, інтерполяція, створення лінії тренда, 4) порівняння експериментальних даних з теоретичними, 5) перевірка гіпотези щодо X^2 – розподілу, 6) завантаження даних в екологічний Grid або збереження в розподіленій базі даних, 7) уточнення параметрів емпіричного розподілу ітераційними методами. Типовий час моделювання при використанні інтерактивного програмного забезпечення Mathcad [5], Statistika [6], Exel [7] на одному робочому місці оператора сягає не менше, ніж 40 хвилин.

Використання паралельної обробки (N процесорів) дає можливість істотно знизити час обрахунку. Враховуючи досвід моделювання з використанням інтерактивних систем, кластер з 15 вузлів дасть можливість обробляти дані в об'ємі, рівному кількості вузлів, помноженому на час моделювання одного завдання, і працювати з різними тестовими наборами параметрів, що відповідає реальним практичним вимогам імітаційного моделювання. Як показали експерименти, час моделювання поширення пожеж за допомогою пакета FDS на робочій станції з піковою продуктивністю 7,2 Gflops (виміряна $\sim 3,2$ Gflops) склала 20 годин, що не прийнятно для практичного використання, оскільки обчислення мають бути закінчені за час порядку 12 годин, щоб отримати результати наступного дня, тобто продуктивність має бути близько 14 Gflops.

Багато завдань моделювання, у тому числі в галузі гуманітарно-екологічних досліджень, зводяться до вирішення комплексу диференціальних рівнянь такими методами, як Монте-Карло, Гауса, Лапласа, Рунге-Кутта [8]. При обчисленнях виконується обробка ве-

ликих масивів даних, яка вимагає значних ресурсів і неможлива без використання сучасних багатопроцесорних систем. Конкретний вигляд рівнянь залежить від моделі досліджуваних процесів. Використання Grid-технологій дозволяє, маючи незначні ресурси кластера, який використовується для введення і зберігання даних, отримувати високу продуктивність за рахунок використання інших ресурсів Grid-системи. Дана робота присвячена задачі створення і організації роботи обчислювального Grid-ресурсу. Конкретні моделі будуть представлені в подальших публікаціях.

3. Архітектура

Grid-ресурс складається з багатопроцесорного кластера і спеціалізованого Grid -шлюза, що знаходиться на головному вузлі і здійснює зв'язок з Grid-інфраструктурою для завантаження завдань і обміну інформацією.

Структура і характеристики кластера: вузлів-8,12 ядер-4 x 2 CPU AMD 64bit 1000-4000 GHz + 4 x 1 CPU I686 PIII-700 MHz, RAM6Gb, HDD -8-80Gb, мережа GigabitEthernet 1000Mbit +(2Mbit WAN), NAS 1,5 TB. Програмне забезпечення: Linux Fedora Core10 PBS torque [9] MPI середовище-open MPI [10], системи моніторинга: ganglia, lm_sensors, MRTG+RRD. Доступ до вузлів здійснюється за протоколом ssh і rsh.

Для кластера вибрано програмне забезпечення Grid проміжного рівня (middleware) ARC (Advanced Resource Connector), що також відоме під назвою проекту NorduGrid [11, 14, 16].

Мультикомп'ютерна архітектура Беовульф [13], що застосовувалася для паралельних обчислень. Це система, що звичайно складається з одного серверного вузла і декількох клієнтських вузлів, сполучених за допомогою мережевої топології Ethernet. Система побудована з готових промислових компонентів, стандартних адаптерів Ethernet і комутаторів, не містить специфічних апаратних компонентів і легко відтворюється. Беовульф також використовує програмні продукти, такі як ОС Linux, середовище програмування Parallel Virtual Machine (PVM) і Message Passing Interface (MPI). Серверний вузол керує всім кластером і є файловим сервером для клієнтських вузлів. Він також є вузлом доступу до кластера і шлюзом у зовнішню мережу.

Головною сполучною ланкою кластера Беовульф із зовнішнім світом, а також з інтерфейсом для завантаження завдань на кластер і здійснення моніторингу, стала Grid-система.

Для Grid-кластера було вибрано програмне забезпечення проміжного рівня (middleware) ARC (Advanced Resource Connector), що також відоме під назвою проекту NorduGrid [16]. Вибір ARC базується на таких методологічних положеннях:

- 1) ARC – це вільно поширювана програмна система з відкритим кодом, яка має вичерпну документацію щодо установки і підтримки;
- 2) ARC не орієнтована на певну область прикладних обчислень (фізика, хімія), а є загальносистемним програмним забезпеченням для побудови Grid-систем;
- 3) ARC не вимагає спеціальних умов до установки (операційна система, система управління локальними завданнями і т.д.), що дозволяє встановити необхідні Grid-сервіси на вже працюючих обчислювальних ресурсах;
- 4) ARC дозволяє підключати нові обчислювальні ресурси без змін і порушень у роботі існуючого сегмента.

4. Інфраструктура ARC (Advanced Resource Connector) [11]

Основні архітектурні рішення ARC дотримуються загальноприйнятих підходів до побудови Grid. При ній використовується організація ресурсів, аналогічна тій, яка застосовується у проекті EU DataGrid [15].

ARC забезпечує такі функції:

- 1) інформаційні (збір і надання інформації про ресурси Grid -системи);
- 2) динамічне включення ресурсів до Grid -системи і їх моніторинг;
- 3) відправлення завдань і виконання їх у Grid-системі, а так само подальше управління їх завданнями;
- 4) розподіл завдань за ресурсами;
- 5) управління даними і ресурсами.

На рис. 1 представлена структурна схема з'єднання Grid-ресурсу з інфраструктурою [11]. Нижче представлений короткий опис основних компонентів.

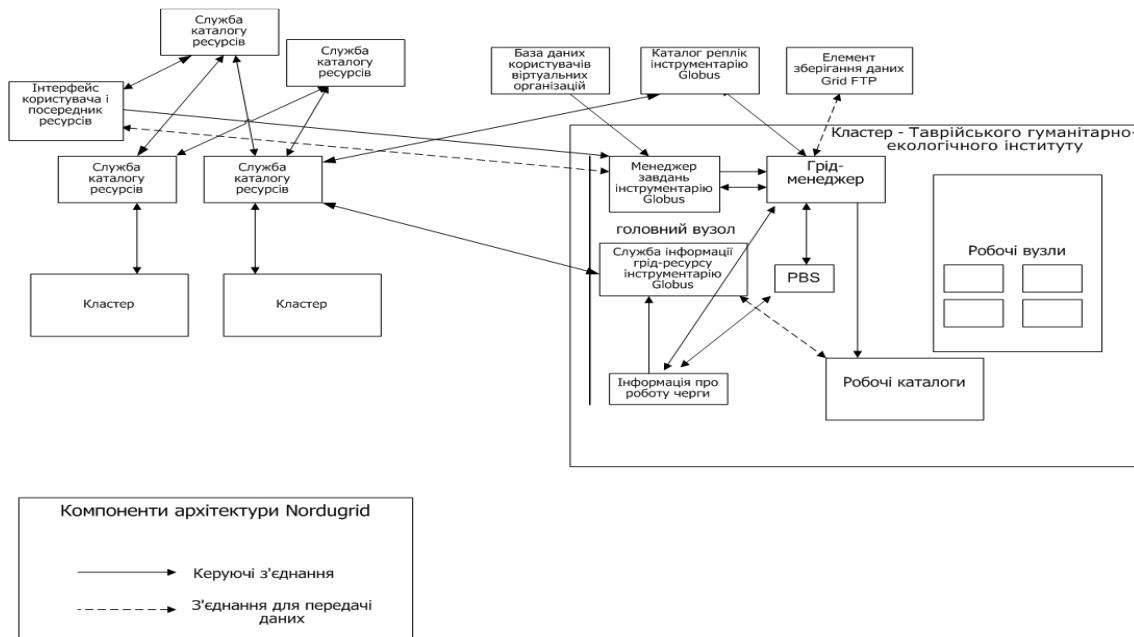


Рис. 1. Структурна схема з'єднання Grid-ресурсу з інфраструктурою

Програмне забезпечення проміжного рівня ARC [11] являє собою надбудову над інструментами Globus Toolkit 2 (GT2). Відмінною рисою NorduGrid ARC є те, що, хоча ця платформа і спирається на протоколи GT2 і реалізована за допомогою API GT2, в ній запропонований власний набір служб, який замінює служби GT2. ARC не використовує інструменти із складу Globus: GRAM-компоненти (Globus Resource Allocation Manager), відповідальні за створення і видалення процесів; утиліти управління завданнями; Gatekeeper і скрипти Job-manager, сервер Wuftp, схеми і інформаційні постачальники MDS (Monitoring and Discovery Service). Зате ARC пропонує власні засоби: Grid – Manager (сервіс-управління завданнями користувачів); gridftp (сервіс – передачі файлів – ARC/NorduGrid GRIDFTP server); User Interface (інтерфейс користувача); Broker (планувальник завдань); систему моніторингу. Окрім цього, запроваджена нова інформаційна схема, для якої розроблені постачальники даних і розширена мова опису ресурсів (xRSL). Інтерфейс користувача (UI) є ключовим сервісом, який був розроблений в рамках проекту NorduGrid [16]. Він призначений для забезпечення запуску завдань користувачів Grid-системи і управління виконувальними завданнями.

Інтерфейс користувача являє собою набір команд для запуску, моніторингу і управління завданнями, а також переміщення файлів і здобуття інформації про достаток ресурсів.

Усі функції реалізовані у вигляді служб, які спираються на відомі програмні засоби з відкритим кодом: OPENLDAP (Lightweight Directory Access Protocol), OPENSSL (Open Secure Socket Layer) і SASL (Simple Authentication Security Layer). Реалізація здійснена на

основі бібліотек Globus Toolkit 2 (GT2), безпека досягається шляхом використання протоколів та інфраструктурних вирішень GSI (Grid Security Infrastructure), заснованих на надійній і поширеній інфраструктурі криптографії з відкритим ключем (Public Key Infrastructure – PKI).

При розробці ARC формувалася мета створення програмного забезпечення високої якості, специфікою досягнення якої є принцип максимальної і повної децентралізації. Тому на кожному робочому місці користувача Grid-мережі встановлюється персональний брокер, функція якого – вибір якнайкращого ресурсу для виконання завдання користувача, яке необхідно реалізувати в Grid-мережі. Цей підхід відрізняє ARC від централізованої схеми EU DataGrid [15] з єдиним брокером на всіх робочих місцях.

Інформаційна система ARC є розподіленою динамічною системою, що надає інформацію про розміщення ресурсів в Grid-середовищі і використання для роботи MDS (Monitoring and Discovery Service) – інформаційної системи Globus [17]. Стандартний пакет MDS є відкритим інструментарієм для створення інформаційної системи для Grid і побудований на базі програмного забезпечення OPENLDAP. Дані, що збираються інформаційною системою, можуть бути найрізноманітнішими і містити, наприклад, дані про конфігурацію або достаток як всієї системи, так і окремих її ресурсів (тип ресурсу, доступний дисковий простір, кількість процесорів, об'єм пам'яті, продуктивність та ін.). Вся інформація логічно організована у вигляді дерева, і доступ до неї здійснюється за стандартним протоколом LDAP – псевдорозподіленої бази даних.

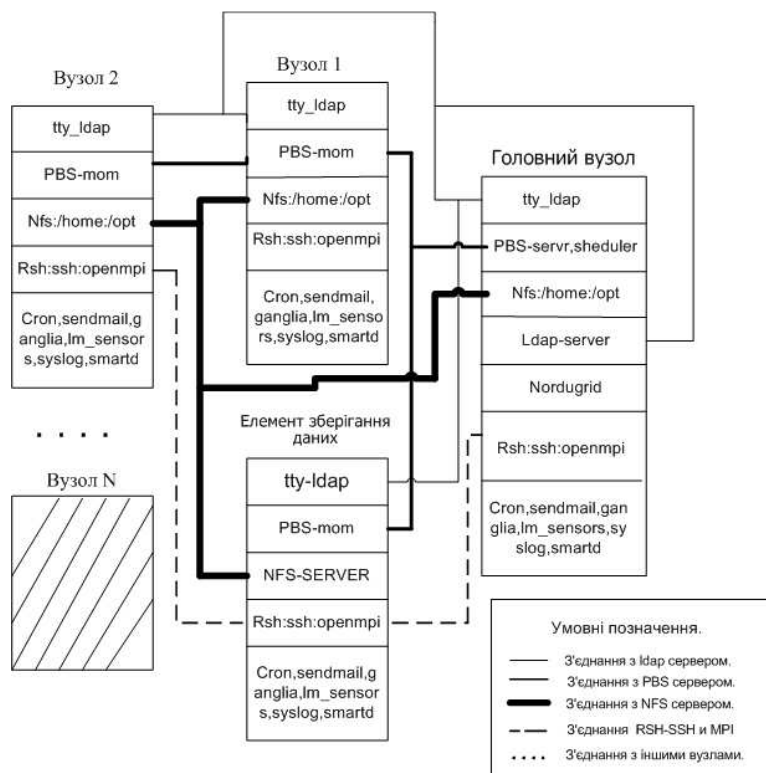


Рис. 2. Структурна схема програмного забезпечення кластера

члені мережеві каталоги. Rsh-ssh- безпарольна авторизація, що забезпечує вхід до вузлів і копіювання інформації в середовищі - MPI. Базові служби (cron, sendmail, ganglia, lm_sensors, syslog, smartd) [19] виконують основну роботу linux-систем на робочих станціях і на головній станції – frond-end node і проводять моніторинг та діагностику.

5. Схема програмного забезпечення кластера

На рис. 2 наведена схема програмного забезпечення кластера і вказаний взаємозв'язок службових компонентів: tty_ldap – механізм авторизації користувачів через сервер ldap, PBS server – Portable Batch System – система управління завданнями для високопродуктивних обчислювальних кластерів спільно з scheduler – планувальником і mom – клієнтом, встановленим на всіх вузлах кластера. Вони розподіляють, планують і встановлюють по черзі завдання для виконання на кластері. NFS-server [18] використовується як сховище інформації, за допомогою якого на всіх вузлах підклю-

6. Тестування Grid-ресурсу

Тестування продуктивності обчислювального кластера проводилося у декілька етапів: тестування і діагностика апаратного забезпечення, установка і оптимізація операційної системи Linux-Fedora core 10, налагодження сервісів (sshd, rshd, NFS, ganglia+RRDtool, lm_sensors, apache+php+mysql, openldap, менеджера ресурсів і системи черг PBS TORQUE-2.3.6 [9]), установка програмного забезпечення, що підтримує технологію паралельного програмування MPI, OPENMPI, налагодження на front-end станції програмного забезпечення Grid-(middleware) Arc – nordugrid [16].

З метою оцінки процесорної продуктивності кластера було використано набір широкоскопних тестів High Performance Linpack Benchmark. Тест полягає у вирішенні системи лінійних алгебраїчних рівнянь вигляду $Ax = f$ методом LU-факторизації з вибором провідного елемента стовпця, де A – щільно заповнена матриця заданої розмірності (в даних тестах 5000×5000). Продуктивність у тесті Linpack вимірюється в кількості виконуваних операцій з плаваючою комою в секунду (флопс). Оскільки кількість операцій, необхідна для вирішення цієї задачі, відома і залежить від її розмірності N , то вимірювана характеристика продуктивності обраховується як частка від ділення числа операцій на час, витрачений на вирішення задачі. Матриці (і деякі інші параметри) не є фіксованими, а задаються користувачем тесту. Вимірювана продуктивність кластера склала $\sim 9,1$ Gflops при піковій $\sim 16,4$ Gflops, що відповідає ефективності використання процесорної потужності при паралельній обробці $\sim 56\%$. Таке значення ефективності пояснюється сильним впливом пропускної спроможності мережі передачі і гетерогенністю кластера. Слід також зауважити, що в тесті Linpack кількість процесорних операцій є $O(N^3)$, а кількість операцій передачі даних – $O(N^2)$. В задачах моделювання, для яких будується кластер, у більшості випадків передача даних між процесорами буде відсутня (статистичне моделювання) або матриця системи буде розріджена (розв'язання рівнянь переносу), і кількість операцій передачі даних не буде залежати від N при заданій кількості процесорів. Отже, для таких задач ефективність системи буде значно вищою.

З метою вимірювання пропускної спроможності комунікаційної підсистеми кластера для програм, побудованих з використанням технології MPI, застосовувався пакет mpi-bench-suite-1.1 (parallel.ru) [15]. Цей тест вимірює час обміну повідомленнями між процесорами паралельної програми в різних топологіях. У даному випадку розглядаються три логічні топології.

Повний граф (Chaos) – взаємодія кожного процесу з кожним. Ця топологія вживається в задачах з нерегулярними зв'язками, таких, як нейронні мережі.

Зірка (Star) – взаємодія основного процесу з підлеглими. Дана топологія використовується в централізованих схемах планування, зокрема, при статистичному моделюванні.

Кільце (Ring) – взаємодія кожного процесу з наступним. Ця топологія відповідає конвеєрній обробці.

Під логічним каналом мається на увазі неурегульована пара вузлів (A,b), які в даній топології можуть обмінюватись повідомленнями. Логічна топологія повністю визначається всією сукупністю задіяних логічних каналів. Для кожної топології розглядаються однонаправлені і двонаправлені обміни. По кожному логічному каналу між вузлами A і B інформація в ході тесту передається в обидві сторони: від вузла A до вузла B і назад передається по L байт інформації. Проте в разі однонаправлених обмінів один з вузлів, наприклад, B, чекає надходження повідомлення від A і лише тоді може передавати A своє повідомлення. У другому ж випадку інформація може передаватися в обидві сторони одночасно. Конкретні способи організації пересилок засобами MPI (блокуючі, неблокуючі пересилки) тут не регламентуються; передбачається, що зі всіх відповідних по семантиці даної топології і способу обмінів буде вибраний варіант з найменшими накладними витратами.

Таблиця 1. Пропускна спроможність для випадку декількох вузлів

	Star	Chaos	Ring
N(P)	$P-1$	$P(P-1)$	P
R(local)	$\frac{2L(P-1)}{T}$	$\frac{2L(P-1)}{T}$	$\frac{4L}{T}$
R(total)	$\frac{2L(P-1)}{T}$	$\frac{LP(P-1)}{T}$	$\frac{2LP}{T}$
R(avg)	$\frac{2L}{T}$		

Для кожної топології задіяні $N(P)$ логічних каналів, де P – число вузлів. Сумарний об'єм інформації, що передається по мережі, є $I = 2LN$. Вузол з номером i має $N_i(P)$ логічних зв'язків з іншими вузлами. Цей вузол передає і приймає всього $2LN_i(P)$ байт інформації. Під локальною пропускною спроможністю R_{local} в кожному тесті розуміється відношення сумарної довжини посланих і прийнятих на даному вузлі повідомлень до витраченого часу. Під сумарною пропускною спроможністю R_{total} мережі розуміється

Таблиця 2. Спільні параметри копіювання даних

Передача даних MB/sec						
Size, K	Star	Star2	Chaos	Chaos2	Ring	Ring2
1	8,79	12,79	10,88	14,61	12,00	15,28
2	12,68	20,82	15,83	18,88	15,50	19,02
4	24,61	33,86	25,57	29,50	25,63	29,24
8	33,54	48,97	34,90	38,04	36,07	38,95
16	46,40	59,47	42,51	46,95	42,55	46,97
1	9,49	12,74	11,53	14,92	11,95	15,27
2	12,63	20,34	15,93	18,99	15,59	19,00
4	24,31	33,47	25,20	29,06	25,47	29,00
8	34,51	49,47	35,46	38,61	35,81	39,17
16	46,23	60,92	42,66	45,89	42,44	46,99

відношення кількості всієї переданої по мережі інформації до витраченого часу. Під середньою пропускною спроможністю R_{avg} одного логічного каналу розуміється відношення сумарної пропускної спроможності до кількості задіяних каналів. Для трьох вказаних вище логічних відомі такі вирази для величин R_{local} , R_{total} , R_{avg} через число вузлів P , довжину повідомлення L і час T , витрачений на всі обміни. При цьому в разі топології Star

Таблиця 3. Максимальна швидкість копіювання даних

Максимальна швидкість копіювання даних MB/sec					
Star	Star2	Chaos	Chaos 2	Ring	Ring 2
46,40	60,92	42,66	46,95	42,55	46,99

розглядається лише основний вузол. У табл. 1 наведена залежність пропускної спроможності від кількості процесорів, звідки видно, що мінімальний час повинен бути для зіркоподібної топології. Тест було запущено на 3 процесорах.

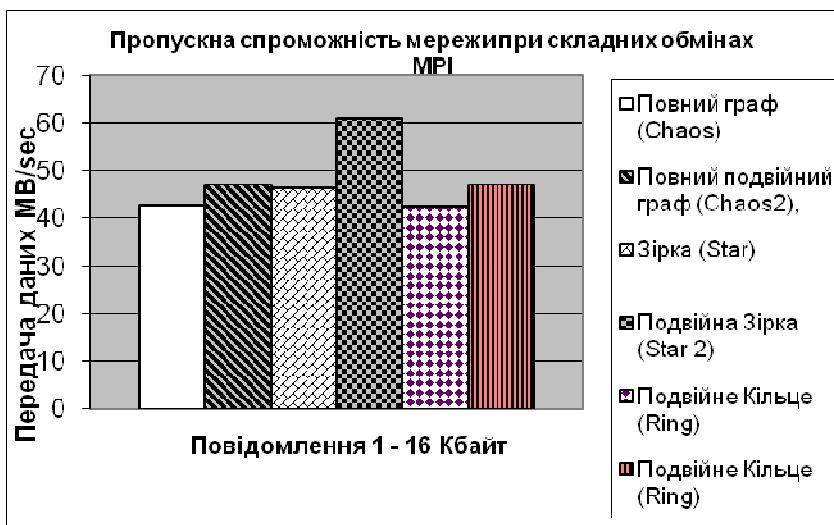


Рис. 3. Тестування мережі MPI

Як одиниця довжини повідомлення було вибрано 1 Кбайт, довжина повідомлення змінювалась в геометричній прогресії від 1 до 16 Кбайт. Кожна тестова процедура складалася з 1000 ітерацій, а весь тест був повторений двічі. Спільні параметри копіювання даних представлені в табл.2. Максимальна швидкість копіювання даних представлена в табл.3. Тест виконався за 56,51 с.

Таким чином, з графіка на рис. 3 бачимо, що найбільша швидкість копіювання даних виконується при структурі “Подвійна зірка” і сягає ~61 Мбайт/с (~488 Мбіт/с). Зірко-подібна топологія, як уже було зазначено, найактуальніша для статистичного моделювання, а отже створений ресурс дасть можливість забезпечити оптимальну швидкість передачі інформації для вказаних задач. Значення одержаної швидкості передачі відповідає приблизно половині пропускної спроможності каналів і є достатнім для складного обміну з використанням MPI.

Тестування проміжного програмного забезпечення Grid було виконано з метою перевірки правильності роботи інформаційної системи, системи передачі файлів та менеджера Grid, а також системи безпеки GSI. Оскільки програмне забезпечення ARC не має власних стандартних тестів, то критерієм коректності роботи інформаційної системи було обрано правильність відображення всіх ресурсів у моніторі (рис. 4), критерієм коректної роботи системи передачі файлів, менеджера Grid та системи GSI – коректність запуску задач і одержання результатів. Всі вказані тести були успішно пройдені.

Для Grid-ресурсу був створений портал за адресою: <http://cluster.crimea.ua>, де ті, що бажають, можуть зареєструватися і дістати доступ до обчислювального кластера, проглянути статистику через мережеві монітори. Надалі за допомогою системи тургоху [17] планується створити авторизацію Grid-сертифікатів. На рис. 4 показаний інформаційний портал Grid-ресурсу.

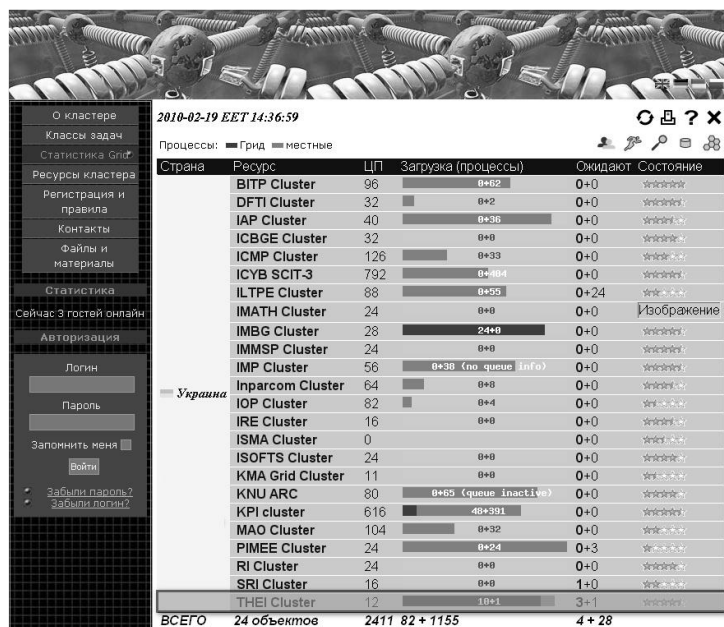


Рис. 4. Інформаційний портал Grid-ресурсу

нути статистику через мережеві монітори. Надалі за допомогою системи тургоху [17] планується створити авторизацію Grid-сертифікатів. На рис. 4 показаний інформаційний портал Grid-ресурсу.

7. Висновки

В результаті даної роботи був запущений перший Кримський експериментальний Grid-ресурс – <http://cluster.crimea.ua>, підключений до Українського академічного сегмента Grid – <http://uag.bitp.kiev.ua>. Результати тестування підтвердили стабільність його роботи і відповідність параметрів вимогам задач моделювання.

Запуск симулятора пожеж FDS з використанням обчислювального кластера і MPI дозволив скоротити час виконання моделювання з 20 годин до 9, а так само, використовуючи масив накопичення даних кластера, збільшилася швидкість звертання до даних при візуалізації. Використання Grid-системи дозволить і надалі скоротити час обробки і створити зручну інфраструктуру для нових програм. Ресурс буде застосований і в галузі гуманітарно-екологічних досліджень. Заплановано і буде проведено модернізацію мережевого апаратного забезпечення, оскільки це впливає на збільшення швидкості мережевого обміну.

СПИСОК ЛІТЕРАТУРИ

1. Khramov V.V. The first Crimean experimental computing grid-resource for research and simulation in the humanities / V.V. Khramov, O.O. Sudakov // Ninth international young scientists conference on applied physics, (Kyiv, June 17–20 2009). – Kyiv, 2009. – P. 105.
2. Стандарт X.509. – Режим доступу: <http://www.itu.int/rec/T-REC-X.509-200508-I>.

3. Український академічний Grid: досвід створення і перші результати експлуатації / Ю.В. Бойко, М.Г. Зінов'єв, М.Я. Свістунів [та ін.] // Математичні машини і системи. – 2008. – № 1. – С. 67 – 84.
4. Європейський Центр Передбачення Погоди (ECMWF). – Режим доступу: <http://www.ecmwf.int>.
5. Каганов В.І. Комп'ютерне обчислення в середовищі Excel і Mathcad / Каганов В.І. – М.: Гаряча лінія – Телеком, 2003. – 328 с.
6. Халафян А.А. STATISTICA 6. Статистичний аналіз даних / Халафян А.А. – [3-е изд.]. – М.: 000«Біном-Прес», 2007. – 512 с.
7. Орвис В. Excel для учених, інженерів і студентів / Орвис В. – Київ: Юніор, 1999. – 528 с.
8. Воєводін В. Паралельне обчислення / Воєводін В. – Спб.: БХВ-Петербург, 2002. – 609 с.
9. Portable Batch System (PBS). – Режим доступу: <http://www.openpbs.org>.
10. Сайт <http://www.mcs.anl.gov/mpi> (Argonne National Laboratory, США).
11. The NorduGrid project: using Globus toolkit for building GRID infrastructure / M. Ellert, A. Konstantinov [et al.] // Nuclear Instruments and Methods in Physics Research. – 2003. – Vol. 502. – P. 24 – 28.
12. Foster I. The Anatomy of the Grid / I. Foster, C. Kesselmann, S. Tuecke. – Blueprint for a New Computing Infrastructure, Morgan Kaufmann, USA, 2000. – 250 p.
13. Sterling T. BEOWULF: A Parallel Workstation for Scientific Computation / T. Sterling [et al.] // Proc. of the International Conference on Parallel Processing, (Urbana- Champaign, Illinois, USA, August 14–18, 1995). – Urbana-Champaign, Illinois, USA: CRC Press, 1995. – Vol. I: Architecture. – P. 11 – 14.
14. Architecture Proposal / M. Ellert, A. Konstantinov, B. Konya [et al.]. – Copenhagen: NORDUGRID-TECH-1, 2002. – Vol. – P. 1 – 5.
15. Сайт <http://www.parallel.ru> (НІВЦ МГУ).
16. The NorduGrid project. – Режим доступу: <http://www.nordugrid.org>.
17. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration [Електронний ресурс] / I. Foster, C. Kesselman, J. Nick [et al.]. – Режим доступу: <http://www.globus.org/research/papers/ogsa.pdf>.
18. The NFS Distributed File Service, NFS White Paper, Sun Microsystems. – 1995. – March. – Режим доступу: <http://www.sun.com/software/white-papers/wp-nfs>.
19. Колесніченко Д.Н. Аллен Linux повне керівництво / Д.Н. Колесніченко, В. Пітер; під ред. М.В. Фінкова. – Спб.: Наука и техника, 2006. – 784 с.

Стаття надійшла до редакції 23.04.2010