

УДК 681.3.06

## МЕТОДИКА ОЦІНКИ ЗАВАНТАЖЕНОСТІ GRID-СЕРЕДОВИЩА НАН УКРАЇНИ

Д.М. КАРПЕНКО, С.Я. СВИСТУНОВ

Зроблено огляд найбільш поширених систем моніторингу Grid-середовища. Запропоновано методика його завантаженості в залежності від свідчень системи моніторингу. Ця методика дозволяє виразити завантаженість Grid у процентах, що показують відношення поточного завантаження до оптимального. Використання запропонованої методики дозволяє поліпшити представлення та аналіз інформації, яка збирається системою моніторингу.

### ВСТУП

Належне управління сучасною Grid-системою неможливе без ефективних систем моніторингу Grid-середовища. Важливою проблемою при побудові системи моніторингу є завдання інтерпретації і представлення отриманих даних, оскільки інформацією, зібраною монітором, користуються як програмні компоненти, що відіграють найважливішу роль в оптимальному розподілі завантаження в Grid для повноцінного використання всіх його ресурсів, так і користувачі Grid, яким інформація потрібна у доступному для аналізу та спостереження вигляді.

У даній роботі запропоновано методика, що дозволяє оцінити завантаженість Grid- системи та виразити її у термінах характеристик обчислювальних ресурсів і завдань, які виконуються у цій системі. Методика може бути використана планувальниками і оптимізаторами планування Grid та інтерфейсними програмами для надання користувачеві загальної інформації про стан завантаженості системи.

### 1. МОНІТОРИНГ GRID-СЕРЕДОВИЩА. ОСНОВНІ ПОЛОЖЕННЯ

Grid [1] [2] — це середовище, що складається з ресурсів, які знаходяться в різних місцях, телекомунікацій (мережеві ресурси), що сполучають їх, і взаємоузгодженого по всій інфраструктурі програмного забезпечення (ПЗ), який зв'язує ці ресурси — так званого програмного забезпечення проміжного рівня (ПЗПР — middleware), що підтримує виконання дистанційних операцій і реалізує функції контролю і керування операційним середовищем.

Успішність функціонування Grid-системи при виконанні задач користувачів та при плануванні завантаження обчислювальних ресурсів багато в чому залежить від вибору системи моніторингу Grid-середовища.

Моніторинг Grid-середовища, зазвичай, складається з чотирьох етапів: генерування подій (опитування сенсорами вузлів і представлення інформації у потрібному вигляді); обробки згенерованих подій; передачі зібраної інформації за вказаними адресами; презентації зібраної інформації. Слід зазна-

чити, що презентувати інформацію можна як користувачам, так і іншим програмним компонентам.

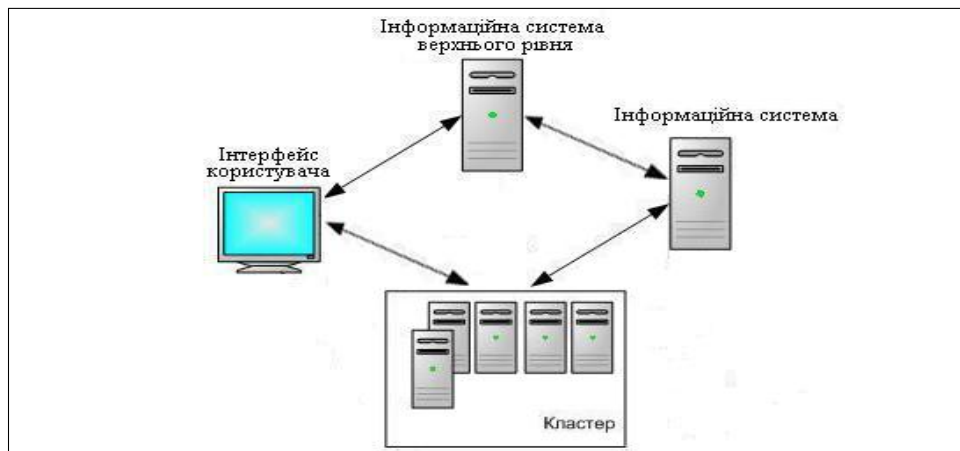


Рис. 1. Архітектура Grid-моніторингу

Архітектура Grid-моніторингу [3] складається з таких стандартних частин:

- постачальник. За його допомогою надається інформація про об'єкти моніторингу (обчислювальні вузли). Постачальником виступають спеціальні служби на кластері, які надають інформацію про об'єкти моніторингу інформаційної системи та користувачеві;
- замовник. Отримує інформацію про об'єкти як мінімум одного постачальника. Замовником виступає, по-перше, користувач, який за допомогою наданого йому інтерфейсу запитує інформацію про вузли конкретного кластеру; по-друге, — інформаційна система, що опитує всі кластери Grid-системи та збирає загальні дані про них; по-третє, — інформаційна система вищого рівня, що збирає дані про наявні в системі інформаційні системи нижчих рівнів;
- реєстр. Пошуковий сервіс, що дозволяє постачальникам публікувати дані, а замовникам отримувати необхідну інформацію. У реєстрі також зберігаються відомості про те, як зв'язатися з об'єктами (адреса, протокол, вимоги безпеки і т.д.). Реєстр може знаходитися в інформаційних системах як нижчого, так і вищого рівнів.

Обмін повідомленнями між архітектурними складовими відбувається за допомогою спеціального API (Application Programming Interface). Після знаходження потрібного постачальника за допомогою реєстру, замовник зв'язується з ним безпосередньо для отримання детальної інформації або для виконання необхідних дій (наприклад, запуску завдання).

Архітектура моніторингу також передбачає опціональні компоненти: перетворювач (republisher) — служба, що має інтерфейси як замовника, так і постачальника і виконує фільтрацію, накопичення, усереднювання та інші дії з отриманою інформацією; репозиторій схем, який зберігає інформацію щодо типів об'єктів. Перетворювачами виступають інформаційні системи обох рівнів.

## 2. СУЧАСНІ СИСТЕМИ МОНІТОРИНГУ GRID-СЕРЕДОВИЩА

### Зауваження щодо вибраних систем моніторингу

Існують та використовуються новітні, орієнтовані на користувача системи моніторингу, такі, як SAM [4] та побудовані на ній Grid-портالي GridView, GridICE, GridMaps [5] та ін. SAM (і, відповідно, побудовані на її основі системи), періодично виконуючи тестову програму, яка моделює роботу користувача у Grid-середовищі, надає інформацію щодо доступності та стабільного функціонування Grid-сервісів. При цьому дані щодо обчислювальних параметрів систем (частота процесору, об'єм пам'яті і т.п.) не надаються, хоча існує можливість отримати такі показники, як кількість процесорів та кількість задач, що виконуються. Зважаючи на це, використовуючи дані системи, неможливо вирахувати інтегральний показник завантаженості Grid-середовища за допомогою методики, описаної нижче. Тому вказані новітні системи моніторингу не розглядаються. Натомість приділено увагу тим системам, за допомогою яких можна реалізувати запропоновану методику [6].

### MonALISA

MonALISA [7–9] (Monitoring Agents using a Large Integrated Services Architecture) — монітор вузлів і мереж у масштабних розподілених системах.

Основою даної системи є багатопотокова служба паралельного та незалежного збору даних з багатьох джерел, якими виступають системи моніторингу кластерів типу Ganglia. Кожен модуль для збору даних конкретного типу або для роботи з конкретним постачальником виконується у своєму незалежному потоці. Система завжди створює керівний потік, який відповідає за видалення потоків, що збоїли, та, якщо вони не виконали свого завдання, їх повторно налагоджує. Потоки системи можуть працювати як за моделлю push, так і за моделлю pull.

Зібрані дані зберігаються локально, індексуються у вбудованій або в зовнішній базі даних і доставляються за запитом замовникові. Клієнт після знаходження відповідної служби може запитувати інформацію в реальному часі і з бази даних або підписатися на отримання інформації про певні параметри, а також використовувати сторонній фільтр для представлення інформації в потрібному вигляді. Для роботи з кожним клієнтом створюється окремий потік.

Для отримання інформації від служб клієнти обирають спеціальні проксі-сервіси, використання яких дозволяє отримувати інформацію від служб, що працюють на захищених firewall машинах, та здійснювати контроль над з'єднаннями служб. Ще однією функцією даних сервісів є мультиплексування інформації, в разі її одночасного запиту багатьма клієнтами (рис. 2).

За допомогою графічного інтерфейсу відбувається візуалізація зібраних даних, детальне відображення інформації про кожен вузол системи та систему в цілому. Графічний клієнт динамічно отримує інформацію про підключення нових служб та модулів і автоматично її відображає.

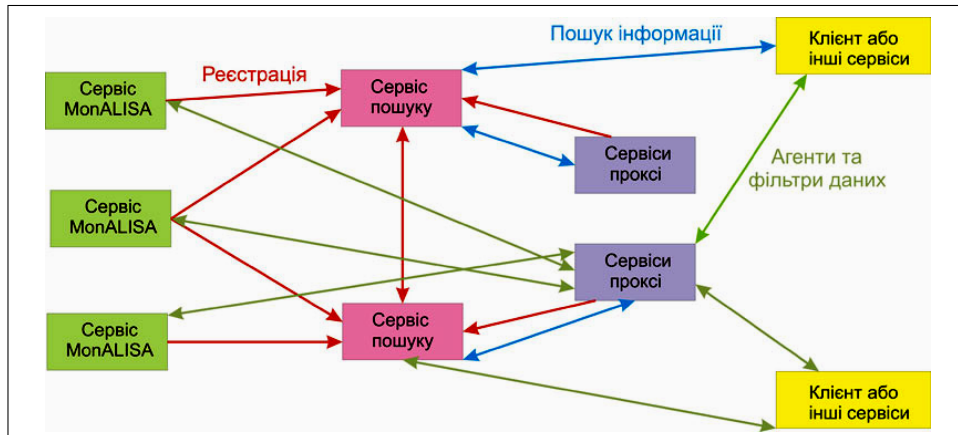


Рис. 2. Схема роботи системи MonALISA

Ця система є гнучким інструментом загального призначення і одним з найбільш універсальних засобів моніторингу Grid-середовища.

## RGMA

RGMA [10] — частина проекту DataGrid [11], заснована на реляційній моделі, об'єднує інформаційну і моніторингову системи. Даний монітор надає такий доступ до інформації, начебто вся вона зберігається в єдиній базі даних.

База даних розподілена на окремі таблиці. Внесення та запит даних здійснюється за допомогою звичайних SQL-запитів. Проте в системі не існує централізованого репозиторію, що зберігав би дані для таблиць. Віртуальна база даних фактично складається зі списків визначень таблиць (схеми), постачальників (реєстру) та набору правил, що визначають, які постачальники повинні надавати інформацію при надходженні конкретного SQL-запиту (рис. 3).

Користувачі та служби Grid-середовища запитують інформацію з віртуальної бази даних за допомогою абонентів. Потрібна інформація надходить до бази від постачальників. Абоненти звертаються до реєстру, щоб вибрати зі списку тих постачальників, які в змозі надати інформацію для відповіді на SQL-запит, що надійшов від клієнта, після чого звертаються до постачальників безпосередньо. Таким чином, RGMA є реляційною програмною реалізацією класичної архітектури Grid-моніторингу (див. розділ 2).

У RGMA використовується декілька видів постачальників. Постачальники баз даних призначені для отримання інформації, що була зібрана раніше та збережена у базі. Поточкові постачальники призначені для динамічного збереження в пам'яті даних, які щойно надійшли. Постачальники реєструються в системі, а в разі припинення роботи, система їх видаляє. Реєстр постачальників централізований, хоча існує потенційна можливість розподіленої реалізації.

RGMA може використовуватися разом з MDS 2 (див. нижче), але при цьому значно знижується швидкість. Ця система має великі потенційні можливості масштабування, хоча вони ще знаходяться на стадії розробки.

Очевидно, що об'єднання всієї інформації в одну логічну базу даних забезпечує виняткову зручність для користувача, однак потребує багато ресурсів.

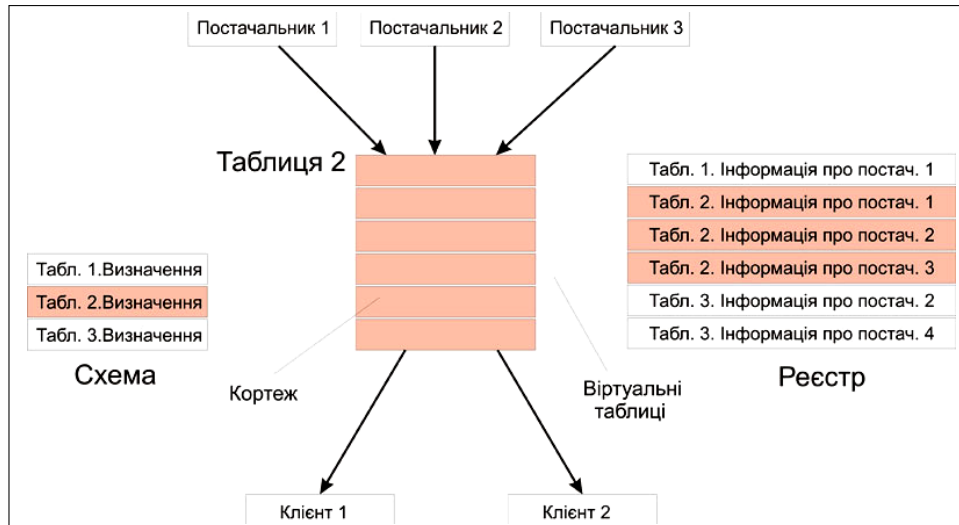


Рис. 3. Компоненти віртуальної бази даних

### Globus MDS 2/4

Globus MDS 2 [12] є складовою частиною програмного інструментарію Globus [13, 14].

MDS 2 заснований на двох ключових протоколах: Grid Information Protocol (GRIP) та Grid Registration Protocol (GRRP). Перший надає можливість запитів і відповідей на них та виконання операцій пошуку. Другий забезпечує реєстрацію компонентів монітору. Вся інформація об'єднується в ієрархічну структуру — дерево інформації каталогів.

Основу MDS складають служби інформації про ресурси Grid (GRIS) і сервіс індексування інформації Grid (GIIS). GRIS збирають та видають за допомогою GRIP інформацію про вузли системи і реєструються на одному або декількох GIIS за допомогою GRRP. GIIS об'єднуються в ієрархічну структуру, в якій кожен елемент накопичує інформацію з індексних сервісів нижчого рівня (GIIS першого рівня отримує інформацію від GRIS). На GIIS також покладаються завдання представлення інформації в потрібному вигляді. Замовники можуть звертатися як до GIIS, так і безпосередньо до GRIS або шукати необхідний GRIS, опитуючи GIIS. У будь-якому випадку використовується протокол GRIP (рис. 4).

Програмна реалізація Globus MDS 2 використовує протокол LDAP, який є базовим транспортним протоколом для GRIP та GRRP. GRIS та GIIS базуються на сервері з відкритим кодом OpenLDAP. Через це всі параметри представлені як об'єкти LDAP у вигляді пар параметр–значення, а сама система є зручною для користувачів, що мають досвід роботи з LDAP.

Оскільки в останніх версіях ПЗ Globus усі служби реалізовані як веб-сервіси (хоча залишена можливість їх запуску не у веб-режимі для збереження сумісності з минулими версіями), у системі моніторингу Globus

MDS 4 [15] GIIS є веб-сервісом. MDS 4 не має власних постачальників даних і в якості таких використовує системи моніторингу кластерів типу Ganglia.

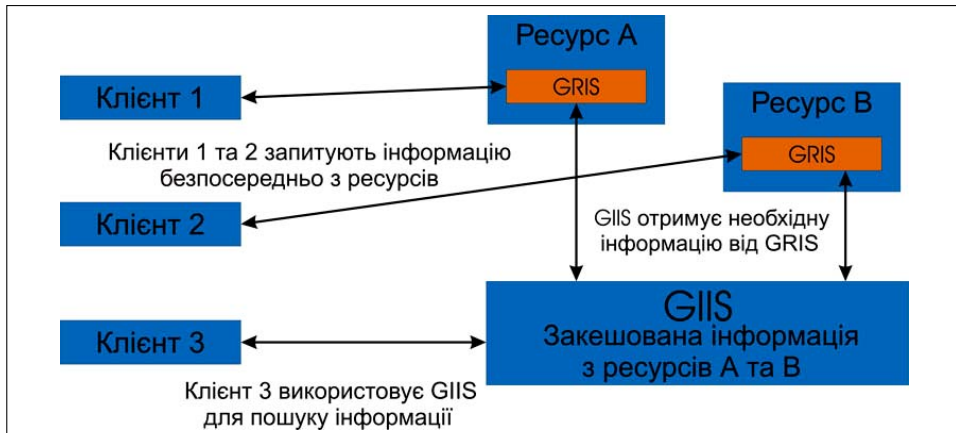


Рис. 4. Схема роботи системи моніторингу Globus MDS 2

### NorduGrid ARC Information System

Схема роботи системи моніторингу Globus MDS 2 NorduGrid ARC IS [16] є складовою частиною ПЗПР NorduGrid ARC та реалізує динамічну, розподілену інформаційну систему. Вона базується на системі моніторингу Globus MDS 2, тому ці системи мають практично ідентичну схему побудови й роботи, але великий обсяг зроблених модифікацій у ARC зумовлює їх несумісність.

Система складається з трьох основних компонентів: служби GRIS (в даній системі мають назву «локальне інформаційне дерево (ЛІД)»); служби GIIS; процеси реєстрації (працюють на локальних ресурсах і реєструють ЛІД в GIIS).

Інформацію на ресурсі збирають програми-постачальники. Вони отримують дані від системи пакетної обробки завдань, від локального Grid-рівня (Grid Manager або GridFTP сервер) або від локальної операційної системи (для Linux-систем це інформація з каталога/proc). Інформація кешується, оберігаючи вузли від надмірного навантаження через часте звернення до ЛІД.

### Об'єкти моніторингу розглянутих систем

Об'єктами моніторингу усіх зазначених вище систем є обчислювальні вузли та стан мережі, що з'єднує їх у кластерну систему. Система моніторингу NorduGrid ARC IS також може надавати інформацію про задачі, які виконуються або стоять у чергах в даному Grid-середовищі.

### 3.МЕТОДИКА ОЦІНКИ ЗАВАНТАЖЕНОСТІ GRID-СЕРЕДОВИЩА

Усі розглянуті системи моніторингу представляють зібрані дані у вигляді набору параметрів по кожному кластеру. За допомогою цих моніторів можливо дізнатися деталі стану окремих кластерів, але при цьому жодна з них не дає показника завантаженості системи в цілому.

### **Опис методики оцінки завантаженості Grid-середовища**

У даній роботі представлено розроблену авторами методику, яка дозволяє оцінити завантаженість Grid-середовища, беручи до уваги основні групи параметрів системи, а саме: обчислювальну потужність кластерів; характеристику каналів зв'язку між кластерами; параметри завдань, що виконуються.

Обчислювальна потужність кластеру визначається кількістю процесорів на кластері, тактовою частотою процесорів, об'ємом оперативної пам'яті, яка припадає на один процесор. Інші показники кластеру, порівняно з перерахованими, мало впливають на швидкість його роботи і, отже, на його потужність.

Єдиним параметром завдань, що має значення для даної методики, є час їх виконання на кластері. При цьому слід зауважити, що як «завдання» (або «задачу», англ. «job») у даному документі мають на увазі програму, яка надійшла на кластер і для свого виконання потребує лише тих ресурсів, які доступні локально. Сьогодні не є рідкістю комплексні задачі (workflow), що складаються з декількох частин, які можуть виконуватися паралельно на різних кластерах і навіть у різних Grid-середовищах. Але для реалізації такого паралелізму комплексна задача розбивається спеціальним планувальником на прості задачі (job), кожна з яких надсилається на свій кластер і використовує лише його ресурси. При цьому під час виконання комунікації між частинами комплексного завдання відсутні, обмін даними відбувається після їх виконання і забезпечується тим же самим планувальником. Таким чином, комплексні задачі не потребують особливого підходу в алгоритмі даної методики, оскільки частини комплексного завдання, що являють собою звичайні задачі, будуть нею враховані.

Ситуація у каналах зв'язку визначається пропускнуною спроможністю мережевого каналу на вході/виході кластеру та кількістю FTP-з'єднань на ньому. Мережеві показники беруться до розгляду через те, що мала пропускну спроможність каналу або велика кількість одночасних FTP-з'єднань, які ініціюються завданнями у системі, може спричинити простоювання вузлів кластеру під час приймання/відправки даних і, таким чином, — збільшення часу виконання задач.

На основі потужності кластеру та часу, потрібного для виконання усіх задач, можна зробити висновок про його завантаженість. Більшість з цих параметрів отримується за допомогою зазначених вище систем моніторингу.

Запропонована методика оцінки завантаженості Grid-системи базується на розрахунку параметрів, за наявності яких система була б оптимально завантажена, і на порівнянні даних параметрів з реальними. Співвідношення між ними, виражене у відсотках, дозволяє робити висновки про ступінь завантаженості.

На початку показники гетерогенного Grid-середовища усереднюються і таким чином розраховуються параметри «середнього» вузла (процесора та оперативної пам'яті, що йому належить) даного середовища. Розрахунок завантаженості базується на припущенні, що Grid-система завантажена оптимально, якщо на кожному вузлі системи виконується тільки одна задача,

яка на середньому вузлі виконалася б за  $T$  годин. Значення  $T$  обирається індивідуально для кожної Grid-системи в залежності від її призначення і типових задач, що виконуються в ній.

Параметри середнього вузла розраховуються за формулами:

$$\text{CPU}_{\text{av}} = \frac{\sum_{i=1}^n K_i * \text{CPU}_i}{\sum_{i=1}^n K_i}, \quad (1)$$

$$\text{RAM}_{\text{av}} = \frac{\sum_{i=1}^n K_i * \text{RAM}_i}{\sum_{i=1}^n K_i}, \quad (2)$$

де  $K_i$  — кількість комп'ютерів у  $i$ -му кластері;  $\text{CPU}_i$  — тактова частота процесора комп'ютера  $i$ -го кластеру;  $\text{RAM}_i$  — об'єм оперативної пам'яті комп'ютера  $i$ -го кластеру;  $n$  — кількість кластерів даної Grid-системи.

Після цього виконується коригування значення  $T$  для кожного кластеру. Оскільки воно є часом виконання задачі для вузла з показниками  $\text{CPU}_{\text{av}}$  та  $\text{RAM}_{\text{av}}$ , його потрібно перерахувати для кожного кластеру системи. Значення  $T$  коригується в залежності від того, наскільки відрізняються показники CPU і RAM даного кластера від середніх. Показники CPU та RAM з точки зору потужності комп'ютера рівноцінні, тому коригуючий коефіцієнт для параметра  $T$  можна розрахувати за формулою:

$$K = 0,5 * \frac{\text{CPU}_{\text{av}}}{\text{CPU}} + 0,5 * \frac{\text{RAM}_{\text{av}}}{\text{RAM}}. \quad (3)$$

Відповідно, відкориговане значення:

$$T_k = K * T. \quad (4)$$

Середній час виконання задач на кластері буде дорівнювати:

$$T_{\text{scl}} = \frac{\sum_{i=1}^n t_i}{N}, \quad (5)$$

де  $t_i$  — час виконання  $i$ -ї задачі;  $N$  — кількість процесорів на кластері.

$$t_i = t_{ds} + t_d,$$

де  $t_{ds}$  — час виконання задачі;  $t_d$  — час передачі даних.

Для розрахунку  $t_d$  визначається кількість FTP-з'єднань на вузлі, де виконується завдання, а також і на вузлі, що постачає або отримує дані. Для кожного з цих вузлів обчислюється смуга пропускання  $W$ , яку отримає завдання, що ініціювало чергове FTP-з'єднання, за формулою:

$$W_{\{1,2\}} = \frac{W_{\text{cl}}}{K_{\text{ftp}} + 1},$$



де  $W_{cl}$  — максимальна пропускна спроможність каналу і  $K_{fnp}$  — кількість FTP-з'єднань на вузлі.

З отриманих значень вибирається найменше:

$$W = \min\{W_1, W_2\}.$$

Далі розраховується власне час передачі даних:

$$t_d = \frac{V}{W},$$

де  $V$  — об'єм даних, які передаються між вузлами.

Відношення  $T_{scl}/T_k$  дозволяє отримати показник завантаженості кластеру:

$$L_i = T_{scl} / T_k. \quad (6)$$

Після розрахунку показника завантаженості кожного з кластерів можна визначити завантаження Grid-системи у цілому:

$$L = \sum_{i=1}^n K_i * L_i, \quad (7)$$

де  $K_i$  — відношення кількості процесорів у кластері до загальної кількості процесорів у системі;  $n$  — кількість кластерів у системі.

Отримане число  $L$  відображає відносне завантаження Grid-середовища у відсотках. 100% відповідає оптимальній завантаженості Grid-середовища.

### Алгоритм реалізації методики

Для практичної реалізації зазначеної методики були зроблені спрощення.

По-перше: якщо на кластері повністю завантажена черга завдань, то такий кластер не буде розглядатися планувальниками як можливий виконавець задач і, фактично, на деякий час просто «випадає» з Grid. Показник завантаженості такого кластеру не розраховується, а приймається рівним заздалегідь вибраній константі. Показник завантаженості кластеру із заповненою чергою вибирається емпірично, але очевидно, що він повинен бути значно більшим за 100%.

По-друге: у більшості випадків можна знехтувати часом на передачу даних, тому що він малий у порівнянні з часом, потрібним для виконання типової Grid-задачі. Отже, мережеву складову і час передачі даних слід розраховувати тільки тоді, коли має місце значна ймовірність простою ресурсів через невеликі пропускні можливості каналів зв'язку.

По-третє: за час виконання задачі приймається задане користувачем в її описі значення. Планувальник Grid намагається поставити задачу до черги саме на ті кластери, які найбільше відповідають вказаним у описі задачі вимогам до оперативної пам'яті та потужності процесора. Тому користувач, задаючи час, необхідний для виконання задачі, оцінює його, виходячи з потрібних параметрів потужності вузла. В більшості випадків задачу буде прийнято на відповідний цим вимогам кластер і, таким чином, час виконання приблизно дорівнюватиме вказаному в описі задачі параметру.

Враховуючи зазначені спрощення, алгоритм реалізації методики для розрахунку завантаження типової сучасної Grid-системи складається з таких кроків:

- розрахувати параметри середнього вузла Grid-середовища за формулами (1) та (2);
- розрахувати коригуючий коефіцієнт для кожного кластеру Grid-середовища за формулою (3);
- використовуючи параметр  $T$ , обчислити оптимальний час завантаження кластеру за формулою (4);
- розрахувати середній час виконання задач на кожному з кластерів за формулою (5), за  $t_i$  приймати час виконання задачі, вказаний в її описі;
- розрахувати завантаженість кожного кластеру за формулою (6);
- розрахувати показник завантаженості середовища в цілому за формулою (7).

Слід зазначити, що на даний момент вже зроблені кроки щодо розробки способу комплексної оцінки завантаженості Grid-середовища. У найновішій версії ПЗПР AliEn, що використовує як систему моніторингу MonALISA, показано оцінку відносної завантаженості Grid-середовища в залежності від кількості задач у ньому. Результат реалізації цього підходу доступний за адресою: <http://alimonitor.cern.ch:8889/show?page=index.html>. У такий спосіб лише приблизно оцінюється завантаженість середовища в цілому, оскільки задачі можуть мати різний час виконання, потребувати досить небагато ресурсів і тому можуть швидко виконуватись на потужних кластерах і т.п.

## ПРАКТИЧНА РЕАЛІЗАЦІЯ МЕТОДИКИ

Розроблена методика була практично реалізована та апробована у Grid-сегменті НАН України [17], загальна кількість вузлів у якому на момент написання статті становила 181. Тут використовувалося ПЗПР NorduGrid ARC, а як система моніторингу — NorduGrid ARC IS. Всі ресурси сегменту реєструються на одному GUIS-сервері першого рівня.

Основним клієнтом у системі моніторингу NorduGrid ARC IS є програма Grid Monitor [18], написана мовою PHP. Її завдання — візуалізація інформації, зібраної системою моніторингу. Програма починає роботу зі з'єднання з усіма доступними GUIS-серверами верхнього рівня і рекурсивно опитує усі служби індексації, поки не дізнається адреси ЛІД на всіх кластерах системі. Після отримання адрес програма з'єднується з кожним ЛІД та отримує потрібні дані.

Методика практично реалізована шляхом модифікації коду програми Grid Monitor для обрання одного з можливих шляхів її використання: представлення ступеня завантаженості Grid-середовища у зручному для користувача вигляді. Розрахунки проводяться по спрощеному алгоритму дій, наведеному в розділі 3.

Для обчислення середніх показників Grid-середовища функції програми, які опитують ЛІД, змінені таким чином, щоб з кожним запитом отримувати необхідні параметри кластерів, такі, як кількість вузлів, тактова частота процесорів і т.п. На основі отриманих даних за наведеними вище формулами розраховуються середні показники Grid-середовища.

Модифікації також внесені до функцій, які опитують черги задач на кластерах. Під час опитування черг збираються такі параметри: кількість задач з даної черги, що виконується (і локальних, і Grid-задач); кількість задач, які стоять у даній черзі на кластері (знову без розділення на локальні та Grid); максимальна кількість задач у даній черзі. Якщо на кластері повністю завантажена черга задач, то такий кластер вважається максимально завантаженим і його показник завантаженості приймається рівним 300%. Такий кластер більше не бере участі у подальших розрахунках. Оскільки на кластері може бути декілька черг, потрібно опитати кожну з них. Кластер буде вважатися максимально завантаженим тільки тоді, коли всі черги максимально заповнені.

На основі зібраних параметрів та еталонного значення  $T$ , що в сегменті НАН України дорівнює 60 хв., відбувається коригування показника  $T$  для кожного кластеру.

Для розрахунку часу виконання всіх задач на кластері підсумовується вказаний у описі час задач. За час виконання задачі приймається атрибут `reqwalltime`, що визначає час, протягом якого задача може займати процесор, незважаючи на те, чи завантажує вона його розрахунками чи ні. Якщо цей параметр відсутній, використовується атрибут `reqcputime`, що визначає, скільки часу задача може використовувати процесор для виконання розрахунків. Якщо користувач потребує більш, ніж одного процесору для своєї задачі, час виконання збільшується у відповідну кількість разів. Для отримання необхідних параметрів функції опитування задач на кластерах модифіковано.

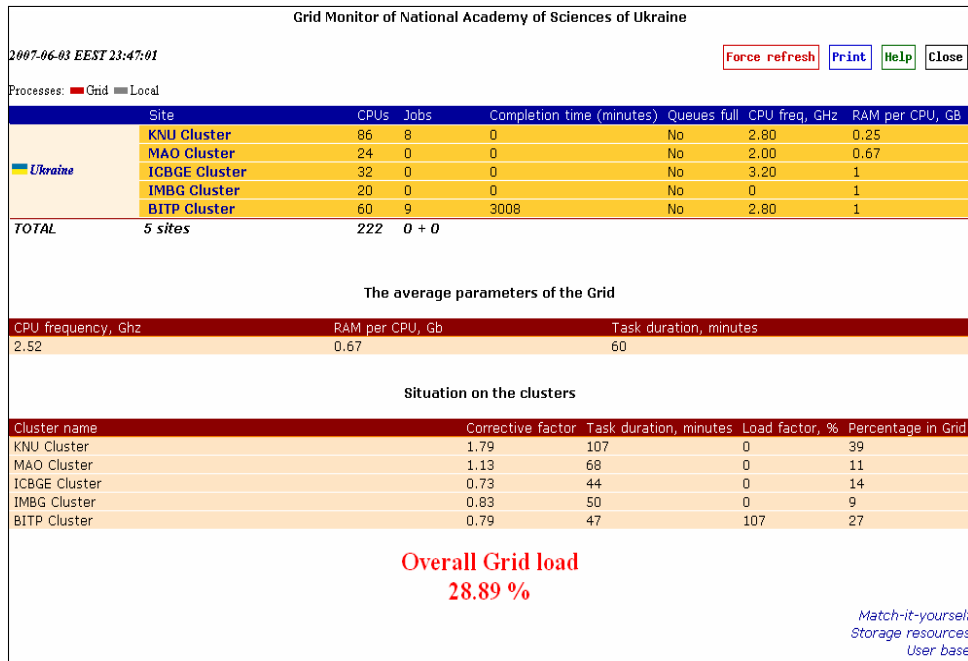


Рис. 5. Загальний вигляд головної Web-сторінки після модифікації коду

Після виконання всіх зазначених вище етапів розраховується показник завантаженості для кожного кластеру. На цій основі відбувається розрахунок показника завантаженості Grid-середовища в цілому. Розраховані показники виводяться на екран (рис. 5).

З роботою немодифікованої програми Grid Monitor можна ознайомитися за адресою: <http://lcg.bitp.kiev.ua>. Модифікований Grid Monitor працює у тестовому режимі (<http://lcg.bitp.kiev.ua/TESTMON/loadmon.php>).

## ВИСНОВКИ

Сучасні системи моніторингу можуть представити параметри стану кластерів, але при цьому не розраховують завантаженість Grid-середовища в цілому, хоча деякі кроки в цьому напрямку вже зроблені. Враховуючи, що єдиний інтегральний показник легше сприймається користувачем, в даній роботі запропонована методика оцінки завантаженості Grid-середовища, результатом якої є залежність поточної завантаженості системи від оптимальної.

Розрахований за допомогою зазначеної вище методики показник завантаженості Grid-середовища може бути застосований у таких прикладних задачах:

- представлення користувачеві Grid-системи, що надсилає до неї задачі, інформації про її завантаженість. Зараз користувач повинен сам робити висновки про загальну завантаженість Grid, базуючись на наборі, часто дуже великому, різних параметрів. Представлення завантаженості Grid-середовища в цілому забезпечить наочність для користувача і полегшить прийняття рішення про доцільність надсилання до Grid своїх задач;
- розробка нових алгоритмів роботи планувальника та оптимізатора завантаження, які базуються на використанні єдиного інтегрального параметра, а не їх набору;
- оцінка роботи Grid-середовища протягом тривалого часу і прийняття рішень про підключення/відключення додаткових ресурсів. Це завдання є актуальним, оскільки зараз здійснюється активна розробка так званих динамічних Grid-систем, що при зростанні навантаження здатні автоматично підключати кластери, що знаходяться у резерві. При спаді потоку задач, навпаки, деяку частину кластерів можна перевести до резерву і відключити від системи, після чого вони використовуватимуться для інших завдань і навіть входять до складу іншого динамічного Grid. Очевидно, що керівні програмні модулі таких систем повинні оцінювати завантаження не окремих кластерів, а всього середовища в цілому.

На даний момент методика реалізована в рамках системи моніторингу NorduGrid ARC IS. Логічним продовженням виглядає її реалізація в інших популярних сучасних системах моніторингу, зокрема, у розглянутих у даній статті.

Іншим напрямком дослідження ефективності практичного використання описаної методики є її інтеграція в існуюче програмне забезпечення для «динамічного» Grid, планувальників задач та оптимізаторів оцінки ефективності роботи модифікованих програм. Дана методика може бути також використана для розробки нових алгоритмів роботи планувальників задач та оптимізаторів Grid, що будуть брати єдиний показник, замість набору параметрів про стан окремих кластерів.

## ЛІТЕРАТУРА

1. Foster I., Kesselman C., Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations // International Journal of High Performance Computing Applications. — 2001. — **15**, № 3. — P. 200–222. — <http://www.globus.org/research/papers/anatomy.pdf>.
2. Foster I., Kesselman C., Nick J.M., Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. — <http://www.globus.org/research/papers/ogsa.pdf>.
3. A Grid Monitoring Architecture. Global Grid Forum Paper, 2002. — <http://www.didc.lbl.gov/GGF-PERF/GMA-WG/papers/GWD-GP-16-3.pdf>.
4. SAM documentation. — <http://sam-docs.web.cern.ch/sam-docs>.
5. Gridmaps homepage. — <http://gridmap.cern.ch/gm>.
6. Zankolas S., Sakellariou R. A taxonomy of Grid monitoring systems // Generation Computer Systems. — 2005. — **21**, № 1. — P. 163–188.
7. MonALISA documentation. — [http://monalisa.caltech.edu/monalisa\\_Documentation.htm](http://monalisa.caltech.edu/monalisa_Documentation.htm).
8. MonALISA: An agent based, dynamic service system to monitor, control and optimize Grid based applications. Computing in High Energy and Nuclear Physics, 2004. — [http://monalisa.cacr.caltech.edu/documentation/monalisa\\_chep04.pdf](http://monalisa.cacr.caltech.edu/documentation/monalisa_chep04.pdf).
9. MonALISA: A distributed monitoring service architecture. Computing in High Energy and Nuclear Physics, 2003. — <http://monalisa.cacr.caltech.edu/documentation/MOET001.pdf>.
10. RGMA architecture description. — <http://www.r-gma.org/arch-virtual.html>.
11. The DataGrid project. — <http://eu-datagrid.web.cern.ch/eu-datagrid/>.
12. Globus MDS documentation. — <http://www.globus.org/toolkit/mds/>.
13. Globus toolkit 2 documentation. — <http://www.globus.org/toolkit/docs/2.4/>.
14. Globus toolkit 3 documentation. — <http://www.globus.org/toolkit/docs/3.2/>.
15. Globus Toolkit 4.0 documentation. — <http://www.globus.org/toolkit/docs/4.0/>.
16. Konya B. The NorduGrid ARC Information System. Technical Description and Reference Manual. — 2006. — [http://www.nordugrid.org/documents/arc\\_infosys.pdf](http://www.nordugrid.org/documents/arc_infosys.pdf).
17. Ukrainian Grid Infrastructure: Practical Experience / M.Zynovyev, S.Svistunov, O.Sudakov, Y.Boyko // IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 6–8 September 2007. Dortmund, Germany. — 2007. — P. 165–169.
18. Smirnova O. The Grid Monitor.Usage manual. — <http://www.nordugrid.org/documents/monitor.pdf>.

Надійшла 28.12.2007