

## ОРГАНІЗАЦІЯ РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ ЗАСОБАМИ ММС SAGE

*В статті розглянуто засоби мобільного математичного середовища Sage для організації розподілених обчислень.*

*В статье рассмотрены средства мобильной математической среды Sage для организации распределенных вычислений.*

*The article presents an abilities of mobile mathematical environment Sage for Distributed Computing.*

**Постановка проблеми.** Концепція мережецентричних обчислень, запропонована засновником фірми Sun Скоттом Мак-Нілі під гаслом «Комп'ютер – це мережа», в останнє десятиріччя призвела до появи нового класу програм, зорієнтованих виключно на роботу в середовищі Web-браузера. Такий спосіб роботи суттєво знижує вимоги до апаратури мережного терміналу та мінімізує вартість його прикладного програмного забезпечення, що є важливим фактором в умовах світової фінансової кризи. Тому перспективним є застосування прикладного програмного забезпечення спеціального призначення, незалежного від використовуваної операційної системи та апаратної платформи, що працює у мережному середовищі [1, с.259–262]. Особливо значний ефект від застосування концепції мережецентричних обчислень може бути досягнутий в математичних програмних засобах, традиційно вимогливих до швидкодії комп'ютерної системи та обсягу використовуваної пам'яті.

**Аналіз актуальних досліджень.** Огляд існуючих розподілених систем показує, що, за рідкісними винятками, вони є вузькоспеціалізованими, призначеними для рішення одного завдання чи класу задач. Найкраще на метакомп'ютерах розв'язуються завдання пошукового й переборного характеру. При цьому обчислювальні вузли практично не взаємодіють під час розрахунку й основну частину

роботи виконують незалежно один від одного [2; 3].

Проте головною проблемою для дослідника є не стільки вибір засобу мережного зв'язку, скільки ефективна реалізація обчислювальних алгоритмів. Одним з ефективних засобів для ефективної організації обчислень є системи комп'ютерної математики, проте їх пряме застосування у грід-комп'ютингу донедавна було досить обмежене.

Усунення цього протиріччя шляхом об'єднання можливостей систем комп'ютерної математики та розподілених обчислень в єдиному динамічному мережному середовищі і визначає актуальність нашого дослідження.

Головною перевагою застосування систем комп'ютерної математики є можливість проведення експериментальної роботи математиками без залучення програмістів. Виходячи з цього, постає актуальна проблема – *надати засоби розподілених обчислень користувачам систем комп'ютерної математики*. Однією з найбільш вдалих спроб зробити це є модуль Distributed Sage – частина вільно поширюваного MMC Sage [4], широко відомої закордоном, але практично невідомої в Україні. Тому в процесі розв'язання поставленої проблеми ми виходили з того, що застосування MMC Sage для реалізації розподілених обчислень дозволить спростити метакомп'ютерні дослідження для непрограмуючих математиків за рахунок автоматизації процесу розрахунку, розподілу завдань та збирання статистики.

**Метою статті** є розгляд засобів MMC Sage для організації розподілених обчислень.

**Виклад основного матеріалу.** Мережні СКМ забезпечують проведення інтерактивних обчислень у середовищі Web-браузера, дозволяють підготувати високоякісні навчальні ресурси з математичних дисциплін, за умов мобільного доступу до обчислювальних програм та даних. Більш того, вони не вимагають установки обчислювального ядра СКМ на клієнтській машині, тим самим вирішуючи проблему інсталяційної бази та ліцензування програмного забезпечення.

Представниками класу мережних систем комп'ютерної математики на сьогодні є Matlab Web Server, webMathematica, wxMaxima та Sage. В якості інструме-

нтального засобу для організації розподілених обчислень була обрана нова відкрита вільно поширювана система комп'ютерної математики, що водночас є і мобільним математичним середовищем – Sage.

Організація паралельних обчислень у Sage можлива на трьох рівнях:

- 1) рівень комунікаційної бібліотеки (як правило, це MPI);
- 2) рівень бібліотек Sage (pynex, python)
- 3) рівень користувача (DSage, lpython)

Найбільш придатним для розв'язання задач дослідження виявився рівень користувача. Дві бібліотеки, що працюють на цьому рівні, мають схожі можливості, проте сфери їх застосування можна розділити достатньо чітко: lpython найбільше придатна для великих гетерогенних середовищ (систем зі спільною пам'яттю та кластерів), в той час як DSage – для розподілених систем з ненадійним мережним зв'язком.

В якості прикладу паралельних обчислень розглянемо чисельне розв'язання рівняння Лапласа на квадратній сітці:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0.$$

Після дискретизації ми паралелізуємо його поділом всієї сітки на дві частини, що взаємодіють на границі:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2} = 0,$$

звідки знаходимо вираз для  $u_{i,j}$ :

$$u_{i,j} = \frac{1}{2((\Delta x)^2 + (\Delta y)^2)} \left( (\Delta y)^2 (u_{i+1,j} + u_{i-1,j}) + (\Delta x)^2 (u_{i,j+1} + u_{i,j-1}) \right).$$

В тексті програми, крім lpython, застосовуються також бібліотеки MPI та weave. Результати роботи програми подані у таблиці 1. Застосовувався метод Якобі, реалізований у бібліотеці weave, кількість ітерацій дорівнювала 1000. З таблиці видно, що на малих масивах швидше виконуються послідовні обчислення, в той час як на великих – паралельні. Крім того, зростання часу при викорис-

танні паралельних бібліотек є лінійно залежним від розміру матриці, а при послідовних – квадратично.

Таблиця 1

### Порівняння послідовної та паралельної реалізації методу Якобі

<i>Розмірність матриці</i>	<i>Послідовні обчислення, с</i>	<i>Паралельні обчислення засобами MPI, с</i>
50x50	0,25	0,61
100x100	0,27	0,71
150x150	0,58	0,89
200x200	1,03	1,13
250x250	1,61	1,44
500x500	6,52	4,31
1000x1000	31,05	17,48
2000x2000	162,94	84,07

Традиційно розподілені обчислення реалізуються за клієнт-серверною технологією. В пакеті DSage (Distributed Sage) клієнтським комп'ютером виступає сервер Sage, що отримує запит від Web-браузера за протоколом HTTP. Локальними серверами, що виконують за запитами клієнта окремі частини завдання, виступають інші системи, на яких має працювати ядро Sage (рис. 1). Подібна організація використовується в MPI, проте тут користувачеві не потрібно явно вказувати сервери; крім того, дана реалізація завдяки застосуванню інтерпретованої мови Python є незалежною від ОС та апаратної платформи (компіляція засобами Cython виконується на локальних серверах після отримання завдання).

Distributed Sage дозволяє організувати розподілені обчислення безпосередньо в середовищі Sage. DSage включає сервер, клієнти та виконавці, представлені у вигляді набору класів (рис. 2):

1. Сервер відповідає за розподіл робіт, отримання та збір результатів. Він також містить Web-інтерфейс, за допомогою якого можна відслідковувати виконання завдань та виконувати інші адміністративні дії.
2. Клієнт відповідає за подання нової задачі на сервер та збір результатів.
3. Виконавці виконують безпосередні обчислення.

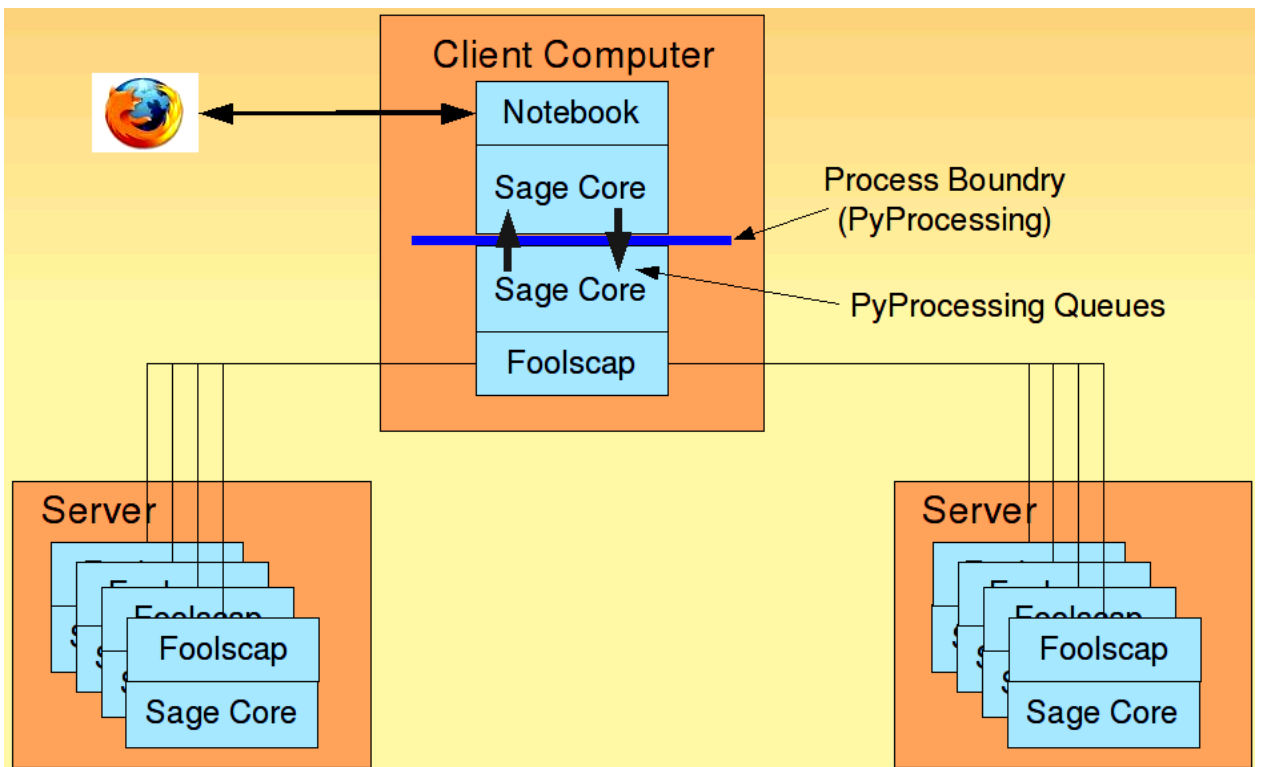


Рис. 1. Взаємодія процесів у DSage

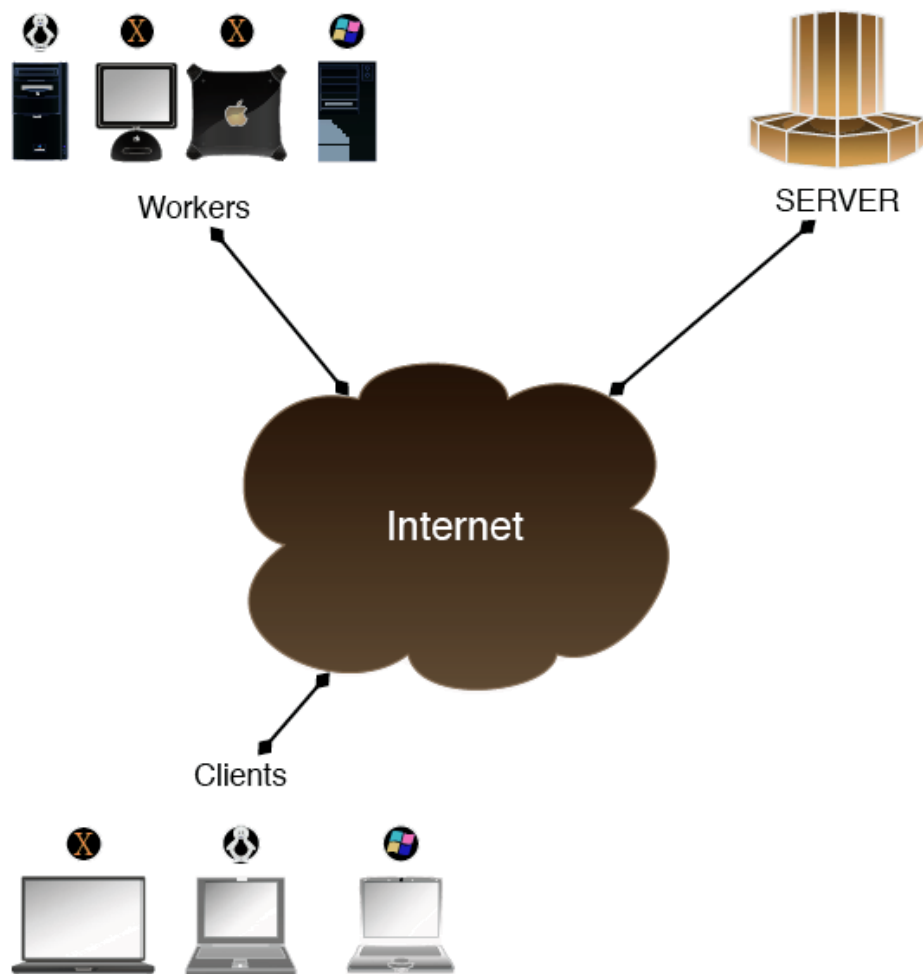


Рис. 2. Об'єкти DSage

Функціональність DSage реалізована в класі DistributedSage. Перед першим його застосуванням необхідно виконати конфігураційну утиліту, що створює бази даних, приватний та публічний ключі для аутентифікації та SSL-сертифікат для сервера. Це необхідно для створення надійного захищеного з'єднання.

DSage, на відміну від інших систем розподілених обчислень (BOINC, XGRID, Chainsaw), здатен ефективно розподіляти як довготривалі завдання, так і короткотривалі. DSage застосовує математичні бібліотеки Sage та є придатним для розв'язання різних типів задач (а не лише традиційних переборних задач).

Завантаження серверу DSage виконується шляхом створення об'єкту класу DistributedSage, до якого можна передати будь-яку команду чи функцію:

```
sage: d=dsage.start_all()
```

Приклади обчислень за допомогою стандартних функцій:

```
sage: res = d('factor(2^325-1)')
sage: res
31 * 601 * 1801 * 7151 * 8191 * 51879585551 * 145295143558111 *
4613679391936953610429590532014122532260339739644049093601
sage: print "Задачу виконано за", res.cpu_time, "секунд"
Задачу виконано за 8.37275099754 секунд
```

Наступний приклад показує, яким чином можна застосувати функції користувача з явним розпаралелюванням. Ми створили нащадок класу DistributedFunction, що містить більшість дій, необхідних для роботи розподіленої системи: постановку завдання у чергу, збір результатів, зберігання завдань з подальшим їх відновленням. Реалізація власної функції process\_result() дозволяє перевірити результат виконання завдання та позначити завдання як виконане чи незавершене. Реалізація власної функції next\_job() дозволяє створити нове завдання та відстежувати його параметри.

```
sage: class DistributedFunctionTest(DistributedFunction):
...     def __init__(self, dsage, n, name='DistributedTestFunction'):
...         DistributedFunction.__init__(self, dsage)
...         self.n = n
...         self.name = name
...         self.result = 0
```

```

...     self.results = []
...     self.jobs = ["print %s"%i for i in range(1,n+1)]
...
...     def process_result(self, job):
...         self.result += job.result
sage: s=DistributedFunctionTest(d,10)
sage: s.jobs
['print 1', 'print 2', 'print 3', 'print 4', 'print 5', 'print 6',
'print 7', 'print 8', 'print 9', 'print 10']
sage: s.results
[]

```

Для перевірки ефективності реалізації розподілених обчислень засобами DSage було використано клас `DistributedFactor`, що дозволяє виконати розкладання числа на прості співмножники (рис. 3). На рис. 4 показано залежність часу рендерингу зображення за допомогою POV-Ray в залежності від кількості використовуваних процесорів (клас `DistributedPOVRay`). Цей приклад показує можливість застосування DSage для розв'язання нематематичних задач.

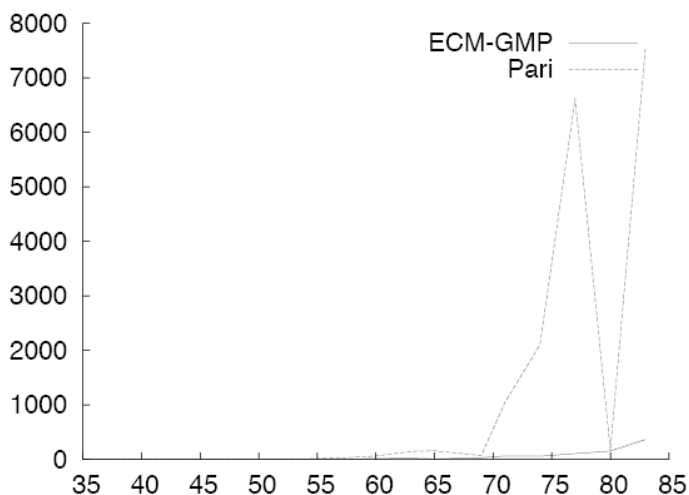


Рис. 3. Залежність часу розкладання від кількості десяткових розрядів числа розподіленіми (PARI) та нерозподіленіми (ECM-GMP) засобами Sage

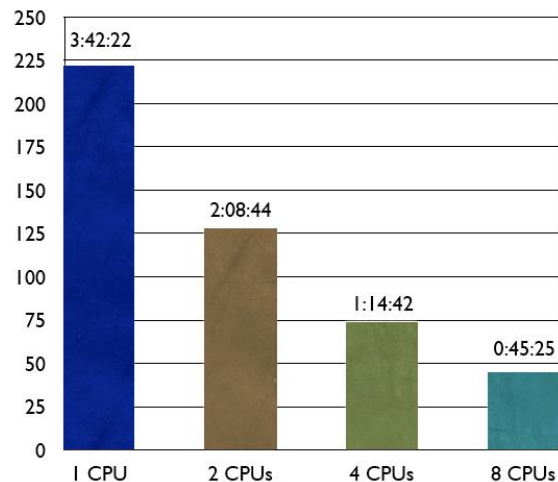


Рис. 4. Залежність часу рендерингу зображення за допомогою POV-Ray в залежності від кількості процесорів

### Висновки:

1. Аналіз можливостей різних розподілених систем показує, що вони спря-

мовані або на розв'язання вузькоспеціалізованих задач, або орієнтовані на програмуючого користувача. Застосування ММС Sage та її модуля для розподілених обчислень дозволяє будувати ефективні розподілені системи різного призначення користувачам, дозволяючи їм зосередитись на реалізації обчислювального алгоритму замість деталей реалізації процесу розподілу завдань.

2. Застосування DSage дозволяє виконати ефективні обчислення як паралельно (в межах багатопроцесорної системи), так і розподілено. Інтерпретована природа коду Sage дозволяє в процесі обчислень брати участь серверам, клієнтам та виконавцям, що працюють на різних програмно-апаратних платформах без будь-якої модифікації програмного коду.

Виконана робота може бути основою для подальших розподілених досліджень з використанням Web-СКМ Sage:

1. Моделювання роботи системи BOINC засобами DSage.
2. Застосування алгебраїчних методів в криптоаналізі блокових шифрів.
3. Організація розподілених матричних обчислень.

#### ЛІТЕРАТУРА

1. Інноваційні інформаційно-комунікаційні технології навчання математики: навчальний посібник / В. В. Корольський, Т. Г. Крамаренко, С. О. Семеріков, С. В. Шокалюк; науковий редактор академік АПН України, д.пед.н., проф. М. І. Жалдак. – Кривий Ріг: Книжкове видавництво Киреєвського, 2009. – 2009. – 316 с.

2. Макаренко Е. В. Использование метакомпьютинга для решения теоретико-числовых проблем / Макаренко Е. В., Семериков С. А. // Збірник праць III Всеукраїнської конференції «Сучасні технології в науці та освіті». – Кривий Ріг : Видавничий відділ КДПУ, 2003. – С. 24–27.

3. Чумак Д. О. Розробка програмного комплексу для метакомп'ютерних обчислень / Чумак Д. О., Семеріков С. О. // Проблеми підготовки та перепідготовки фахівців у сфері інформаційних технологій : матеріали V Міжнародної науково-технічної конференції «Комп'ютерні технології в будівництві» : Київ–Севастополь, 18–21 вересня 2007 р. – Кривий Ріг, 2008. – С. 102–103.

4. Шокалюк С. В. Основи роботи в SAGE / Шокалюк С. В. ; за ред. академіка АПН України М. І. Жалдака. – К. : НПУ імені М.П. Драгоманова, 2008. – 64 с.