УДК 004.75

ПЕТРЕНКО А.І.,
БУЛАХ Б.В.,
ЧЕКАЛЮК В.В.

АВТОМАТИЗАЦІЯ ВИКОНАННЯ ПОТОКІВ РОБІТ У ГРІДІ ІЗ ЗАЛУЧЕННЯМ ГРІД-СЕРВІСІВ

В даній статті розглядається архітектура системи автоматизованого виконання потоків робіт (англ. workflows) у гріді, складених з веб-сервісів та грід-сервісів. Описано механізм взаємодії компонентів системи, вказано деталі її практичної реалізації. Проведено оцінку часу виконання потоків робіт в рамках даної архітектури, наведено результати експериментів на тестовій інфраструктурі.

This article describes the architecture of the software system for automated workflow execution in grid environment for workflows composed of web services and grid services. The interactions between the components of this system are described and the details of its implementation are given. The workflow execution time is evaluated and the experimental results are provided.

1. Вступ

Грід-обчислення як технологія експлуатації віддалених обчислювальних ресурсів, наданих у спільне використання учасникам певних об'єднань користувачів (віртуальних організацій), вже давно успішно застосовуються для рішення різноманітних задач науки та інженерії високої обчислювальної складності [1]. Однак навіть сьогодні більшості користувачів (зокрема й українського гріду) доступна лише базова функціональність грід-середовища, що дозволяє віддалено запускати поодинокі обчислювальні задачі, контролювати стан їх виконання та вивантажувати результати обчислень. Для організації виконання складних сценаріїв обчислень, що вимагають скоординованого виконання у гріді десятків-сотень задач, такі можливості є, як правило, недостатніми. Можна стверджувати, що нині лишаються актуальними проблеми розробки і подальшого розвитку таких прикладних грід-середовищ, які б надавали достатньо гнучкості для організації власних обчислювальних сценаріїв користувачами, при цьому не вимагаючи від них додаткової підготовки, яка наразі необхідна при безпосередній роботі з базовими засобами службового програмного забезпечення проміжного рівня (ПЗПР).

Серед найбільш поширених дистрибутивів ПЗПР, що керують українськими грід-ресурсами, а саме – Nordugrid ARC та EGEE gLite різних версій, лише gLite надає можливість опису «композитних задач», складених з атомарних грід-задач. Натомість пропонується дослідити альтернативний підхід, що полягає у використанні принципів сервісно-орієнтованої архітектури (СОА [2]), зокрема – використати композицію грід-сервісів для організації гетерогенних обчислювальних сценаріїв.

2. Постановка задачі

Під обчислювальним сценарієм будемо далі розуміти попередньо визначену послідовність виконання обчислень, спрямовану на отримання певного визначеного результату. Під обчислювальним потоком робіт будемо розуміти такий обчислювальний сценарій, який можна представити у вигляді N відокремлених кроків обчислень $S_i \mid i = 1..N$, що виконуються у певній послідовності відносно один одного [3]. Якщо кроками потоку є грід-задачі, такий потік будемо називати потоком грід-задач [4]. Задачами даної статті є: а) дослідити архітектуру системи проектування та виконання потоків грід-задач, у якості кроків яких виступають грід-сервіси; б) оцінити часову ефективність виконання таких потоків відносно виконання обчислень на локальному ресурсі.

3. Організація потоків грід-задач

Як вже зазначалося вище, базових засобів таких ПЗПР, як Globus Toolkit чи ARC, не вистачає для повноцінного проектування та виконання потоків грід-задач. Однак, як правило, кожне відоме ПЗПР надає засоби для розробки власних клієнтів та планувальників у вигляді бібліотек та інструментарію розробника (SDK)

під різні мови програмування (напр. засоби libarcclient з проекту ARC).

Певним винятком є можливості ПЗПР gLite по запуску взаємозалежних грід-задач як єдиної «супер-задачі» засобами планувальника Condor DAGman. Такі сценарії описуються на мові опису задач Job Description Language (JDL), підтримуваній gLite. Втім, рішення, що базуються на даному функціоналі, дієві лише для gLite-інфраструктур.

З часів появи ПЗПР Globus Toolkit 3 чимале поширення дістала сервісно-орієнтована архітектура ПЗПР. Нині в тій чи іншій мірі усі дистрибутиви ПЗПР використовують веб-сервіси (або їх варіант – грід-сервіси) як програмний інтерфейс до своїх підсистем. Тож напрошується рішення, що використовуватиме як грід-сервіси, так і веб-сервіси як складові етапи (елементи) S_i потоку робіт. До переваг такого підходу порівняно з планувальником грід-задач типу DAGman можна віднести: можливість організації «змішаних» сценаріїв обчислень, складених не лише з грід-задач; сумісність з численними стандартами на веб-сервіси та їх композиціювання, і, відповідно, – з наявним інструментарієм.

4. Грід-сервіс

Згідно опису відкритої архітектури грід-сервісів OGSA [5], грід-сервісом може називатися лише такий програмний компонент, що відповідає усім вимогам саме цієї архітектури. Надалі дозволимо більш вільне трактування грід-сервісу як веб-сервісу, що надає доступ до грід-ресурсів (керує ними).

Характерною проблемою при реалізації грід-сервісів є довготривалі транзакції. Сервіси у гріді часто мають справу з тривалістю операцій, що перевищує допустимі межі для «таймауту» з'єднання. Серед можливих шляхів вирішення даної проблеми були розглянуті наступні: механізм оповіщень для асинхронного повідомлення клієнту про завершення операції, що дещо ускладнює архітектуру та підсилює її зв'язність, та циклічне опитування сервісу про стан виконання розпочатої операції.

Для практичної перевірки даного підходу було розроблено веб-сервіс, здатний ініціювати виконання обчислень у гріді. Його інтерфейс складають наступні операції (не включаючи службових операцій, що вирішують такі питання, як делегування прав тощо):

- *submitJob*: запуск обчислень. Повертає унікальний ідентифікатор задачі, який слугуватиме ключем для подальшої роботи із задачею;
- *cancelJob*: скасування виконання задачі;
- *getJobStatus*: отримання поточного статусу задачі;
- *getJobInfo:* отримання розширеної діагностичної інформації про задачу та грід-ресурс;
- *getJobOutput:* отримання результатів виконання грід-задачі.

Сам перелік операцій не є унікальним і співзвучний з інтерфейсами таких веб-сервісів, як WM-Proxy з ПЗПР gLite та A-REX з ARC. Розроблений сервіс відрізняє орієнтація на виконання конкретних обчислень з області моделювання електронних схем, що забезпечуються функціоналом комплексу NetALLTED [6]. Механізм роботи грід-сервісу спрощено проілюстровано на рис.1.

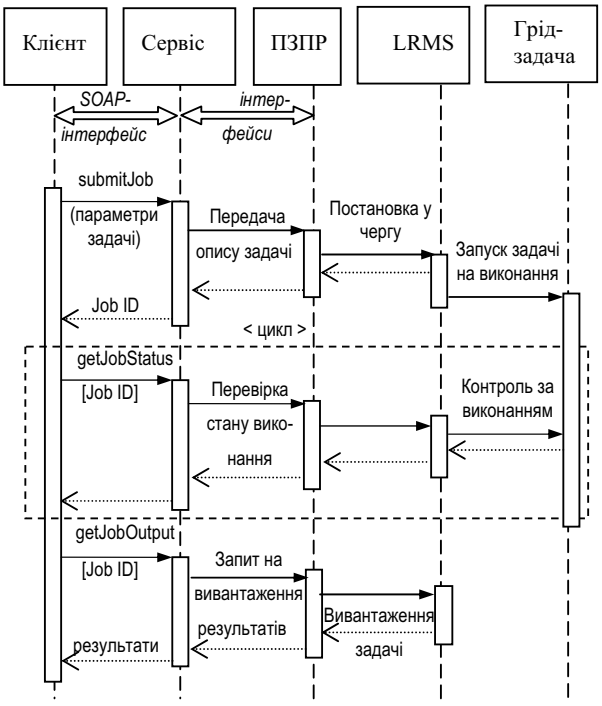


Рис.1. Спрощена діаграма послідовності взаємодії з грід-сервісом

Задля наочності дана діаграма не відображає усіх операцій, пов'язаних із запуском задачі.

Оцінимо час виконання задачі через грід-сервіс. Загальний час на отримання результатів від виконання обчислень на віддаленому ресурсі можна представити як:

$$T = \sum_{i=1}^{s} I_i + E + \sum_{j=1}^{d} O_j \qquad (1)$$

де I_i – час на i-ту транзакцію вхідних даних між хостами на шляху від джерела даних до обчислювального ресурсу (s – загальна кількість транзакцій), O_j – час на j-ту транзакцію вихідних даних між хостами на шляху від обчислювального ресурсу до сховища результатів (d – загальна кількість транзакцій), E – час на виконання обчислень на обчислювальному ресурсі.

У випадку, коли запуск обчислень передбачає взаємодію із додатковим службовим ПЗ (ПЗПР типу локальних менеджерів ресурсів кластерів або ПЗПР гріду), з'являються додаткові затримки. Окремо слід врахувати, що інформація про статус задачі оновлюється у інфосистемі ПЗПР з певним періодом τ (залежно від налаштувань ПЗПР). Тоді реально спостережений час виконання E^* можна оцінити як:

$$E \le E^* < E + \tau \qquad (2)$$

Тоді оцінка (1) приймає вигляд:

$$T = \sum_{i=1}^{s} I_i + P + E + \sum_{k=1}^{c} \tau_k + \sum_{j=1}^{d} O_j \qquad (3)$$

де P – час на підготовку до виконання обчислень конкретним ПЗПР, h – кількість рівнів ієрархії інформаційної системи ПЗПР, τ_k – період оновлення інформації про стан виконання обчислень на k-му рівні інфосистеми (оцінка є граничною, для найгіршого випадку співвідношення (2)).

Перейдемо до оцінки часу виконання грід-задачі. Для типового сценарію роботи в гріді маємо наступне.

1. Завантаження даних на віддалений ресурс (stage in). Час виконання i-тої транзакції:
$$I \approx D_I / W_I \qquad (4)$$
де D_I – об'єм вхідних даних, W_I – пропускна здатність мережі між джерелом даних та грід-ресурсом. Аналогічно, для вивантаження даних (stage out) час виконання j-тої транзакції:
$$O \approx D_O / W_O \qquad (5)$$
де D_O – об'єм вхідних даних, W_O – пропускна здатність мережі між ресурсом та кінцевим сховищем даних.
2. Накладні витрати на підготовку до виконання складають:
$$P = T_a + T_Q + T_g \qquad (6)$$
де T_a – часові витрати на авторизацію та делегування (клієнт делегує ПЗПР право діяти від його імені), вимірюються секундами, T_Q – час на постановку та очікування задачі у черзі локальної системи керування ресурсом (LRMS), T_g – інші накладні витрати, що не залежать від даних самої задачі (створення каталогів сесії тощо). T_Q для ресурсу зі спільним доступом – випадкова величина, що залежить від його поточної завантаженості. Обираючи для моделювання грід-ресурсу ту чи іншу модель СМО та її параметри згідно статистичних спостережень за ресурсом, можна наближено оцінити середній час очікування заявки у черзі.
3. Фактичний час виконання задачі визначається продуктивністю ресурсу R, алгоритмом виконання задачі A, причому може залежати і від розміру вхідних даних:
$$E = f(A, R, D_I) \qquad (7)$$
4. Інфосистема гріду, як правило, розрахована на тривалі задачі і має період оновлення τ_G ~30 с (залежно від налаштувань).

При оцінці часу виконання обчислень через грід-сервіс слід врахувати те, що недоліком SOAP-протоколу є зависоке відношення кількості службових символів до корисної інформації та можливість передачі лише символів, що відображаються (алфавіт, цифри, пунктуація та деякі спецсимволи). Зважаючи на це, бінарні дані передаються, як правило, у кодуванні Base64, що додатково збільшує розмір даних приблизно на третину. Також до витрат на кодування-декодування даних XML слід додати витрати на шифрування повідомлень при безпечному SSL-з'єднанні, що істотно зростуть у випадку використання шифрування самих даних повідомлення (WS-Security).

У випадку, коли дані прямо передаються сервісу через SOAP, слід скоригувати (4) та (5):

$$I \approx K_{WS} D_I / W_I, O \approx K_{WS} D_O / W_O \qquad (8)$$

де K_{WS} – коефіцієнт, що враховує накладні витрати на кодування/шифрування.

5. Потік робіт з грід-сервісів

Однією з переваг грід-сервісів над іншими інтерфейсами до ПЗПР є те, що перші краще придатні для композиції внаслідок відкритості

та стандартності своїх WSDL-описів, кращої стандартизації самої композиції (так зване «оркестрування»), наявності численних засобів розробки та розгортання, як комерційних, так і відкритих.

Архітектуру запропонованої системи виконання потоків задач складають:

– **сервіс керування потоками**, що має набір операцій, загалом аналогічних операціям грід-сервісу (за виключенням того, що сервіс керування потоками оперує описом та даними для усього потоку, а не окремого грід-сервісу): *submitTask, cancelTask, getTaskStatus, getTaskInfo, getTaskOutput*.

– **ПЗ автоматичного виконання потоків** грід- та веб-сервісів. Сервіс керування приймає опис потоку на певній мові та перекладає його на стандартну мову виконання потоку веб-сервісів, що підтримується цими засобами. Одним з таких стандартів є мова WS-BPEL 2.0. Ця мова описує так званий «бізнес-процес», що зовні поводить себе як «віртуальний» веб-сервіс. Таким чином, потік робіт представляється одним «композитним» веб-сервісом. Серед некомерційних BPEL-засобів нині доступні такі продукти, як Apache Ode, OW2 Orchestra та ін. Останній і було використано у практичній реалізації.

– **грід-сервіси**, з якими взаємодіє ПЗ автоматичного виконання потоку згідно отриманого BPEL-опису.

– **службові компоненти**: реєстр доступних грід-сервісів та їх метаданих, сховище грід-сертифікатів та ін.

Перші три компоненти розробленої архітектури виконання потоків робіт, складених з веб- та грід-сервісів, представлені на рис.2.

6. Оцінка ефективності рішення

«Чистий» (без урахування накладних витрат) час виконання потоку сервісів повністю визначається конфігурацією потоку G (що зазвичай задається графом):

$$E_W = f(T_i, I_i, O_i, G), i = 1..N \qquad (9)$$

де T_i – час виконання i-го етапу потоку робіт. Для оцінки E_W можна дотримуватись наступної процедури. Розглянемо довільний потік робіт з N кроків як послідовно сполучені M під-потоків, кожен з яких містить кілька паралельних гілок. Тривалість виконання кожного з M під-потоків визначається найбільшою тривалістю виконання з усіх паралельних гілок. Рекурсивно застосовуючи дане розбиття, можна отримати оцінку часу виконання для довільного потоку.

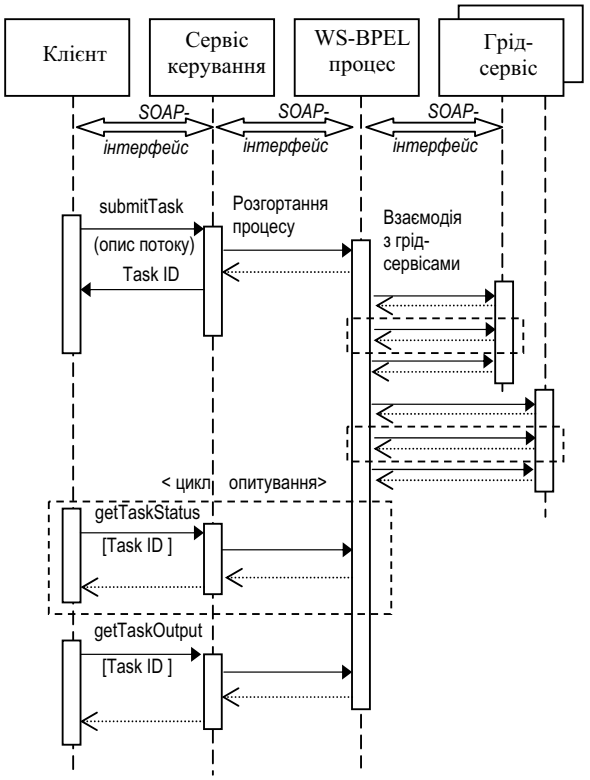


Рис.2. Взаємодія компонентів архітектури системи керування виконанням потоків робіт

Оцінимо тепер накладні витрати на виконання потоку. Витрати на пересилку даних визначаються аналогічно до (8), однак за випадку, коли використовується BPEL-процес, кількість пересилань може зрости приблизно у 2 рази. Аналогічно до міркувань щодо грід-сервісів, тут також слід урахувати інтервал опитування стану виконання потоку τ_W, а також інтервал опитування BPEL-процесом грід-сервісів τ_S. Накладні витрати також збільшаться за рахунок часу на розгортання BPEL-процесу T_{BP}.

Для експериментальної оцінки ефективності запропонованого підходу у гріді вирішувалась задача моделювання та оптимізації схемотехнічного рішення за допомогою функціоналу пакету NetALLTED. Подальші уточнення та припущення справедливі для задач моделювання на схемотехнічному рівні.

Експеримент проводився на виділеній тестовій грід-інфраструктурі з наступними характеристиками. Основним компонентом є тестовий

кластер, який складається з головного вузла та чотирьох робочих вузлів. Кожен робочий вузол має 2 чотириядерні ЦП Intel Xeon E5345 2,33 ГГц, 8 Гб оперативної пам'яті та жорсткий диск ємністю 500 Гб.

Оцінка часу виконання проводилася для потоків, складених з задач одного з кількох умовних класів: короткотривалі (E ~ 0,05 хв. на вищевказаних ресурсах), середньої тривалості (E ~ 7 хв.) та довготривалі (E ~ 35 хв.), коригування часу виконання відбувалось за рахунок зміни параметрів процедури оптимізації.

Грід-задачі було виділено в окрему високопріоритетну чергу (під час проведення експерименту у черзі не було сторонніх задач). Грід-ресурси керуються ПЗПР Nordugrid ARC 0.8.3.

Розглядався чи не найвигідніший у порівнянні з послідовним виконанням випадок з M = 1 та N паралельними гілками (тобто, одночасний запуск N задач моделювання електричної схеми з різними наборами вхідних параметрів; для оцінки часу при плануванні послідовного потоку робіт див. також [7]). Позначимо символом «′» параметри виконання потоку на одному локальному вузлі, а «′′» – параметри потоку грід-сервісів. Тоді з формул (3-9) для вищеописаного сценарію час виконання 1 етапу потоку:

$$T_1' = E' \tag{10}$$

$$T_1'' = E'' + P + \sum_{i=1}^{s} I_i + \sum_{j=1}^{d} O_j + \tau_G \tag{11}$$

Тоді при обмеженнях на ресурси у n ядер на локальному вузлі та n_G вільних ядер у гріді повний час виконання потоку:

$$T' = \begin{cases} E' \mid N \le n \\ E'N/n \mid N > n \end{cases} \tag{12}$$

$$T'' = (E'' + P + \tau_G + \sum_{i=1}^{s} I_i + \sum_{j=1}^{d} O_j)\lceil N/n_G \rceil + \\ + \sum_{i=1}^{s''} I_i'' + \sum_{j=1}^{d''} O_j'' + T_{BP} + \tau_S + \tau_W \tag{13}$$

Враховуючи, що об'єм вхідних та вихідних даних запущених задач є нехтовно малим відносно пропускної здатності мережі (що є поширеним сценарієм для задач схемотехнічного моделювання), а шифрування не застосовувалось, накладними витратами на пересилку даних можна знехтувати. Тоді з (12) та (13) слідує, що для такого типу задач виграш у часі від запуску потоків грід-сервісів порівняно з локальним виконанням можна отримати при $N > n$ за умови:

$$E' \frac{N}{n} > \left\lceil \frac{N}{n_G} \right\rceil (E'' + P + \tau_G) + T_{BP} + \tau_S + \tau_W \tag{14}$$

Експеримент проводився за наступних умов (дані отримані на основі налаштувань та попередніх тестів окремих компонентів системи): T_{BP} = 30 с, P = 1 хв., τ_G = 30 с, τ_S = 10 с, τ_W = 12 с. З (14) слідує, що при N = 16 виграш у часі можна отримати при E > 3 хв., при N = 32 та N = 64 маємо виграш для задач довше 1 хв. (мається на увазі, що $E' = E'' = E$).

Результати виконання потоків грід-задач, що загалом підтвердили ці оцінки, зведені у табл.1-3 та проілюстровані на графіках (рис.3). Слід зазначити, що теоретичні оцінки враховували максимально можливі затримки інфосистеми, а також не враховували випадкового характеру затримок P.

При $N > n$ = 8 для задач середньої та довгої тривалості виграти у часі відносно рішення на потоках грід-задач не дозволило навіть використання для локальних обчислень потужніших ЦП (Intel Xeon E5440 @ 2.83ГГц, також 2 ЦП по 4 ядра на вузол, локальне виконання пришвидшилось на ~25%).

Поряд із запуском потоків грід-сервісів проводились запуски тих самих наборів задач за допомогою bash-скриптів та утиліт командного рядка ПЗПР. Даний механізм запуску дозволяє скоротити накладні витрати на T_{BP}+ τ_S+τ_W ~ 0,5..1 хв., однак є заскладним для безпосереднього використання грід-користувачами, і програє у гнучкості запропоновану рішенню при побудові прикладних грід-середовищ, дружніх до користувача.

7. Висновки

Представлено архітектуру системи виконання потоків грід-задач, що використовує сервісно-орієнтований підхід та базується на окремих незалежних функціональних одиницях – грід-сервісах. Особливостями даного рішення є: використання мови WS-BPEL 2.0 у якості внутрішнього формату опису потоку, що дозволяє спиратися на існуючі стандартні BPEL-засоби при виконанні потоку робіт; використання грід-сервісів, що сумісні з WS-стандартами та дозволяють створення гібридних потоків робіт, складених як з грід-сервісів, так і звичайних

веб-сервісів. Було розроблено робочий прототип системи, що дозволив виявити деякі недоліки даного рішення, на які слід звернути увагу при подальшому вдосконаленні системи, а саме: надмірні об'єми внутрішнього інформаційного обміну, та, відповідно, суттєві накладні витрати при передачах великих об'ємів даних.

Було розроблено та інтегровано в дану архітектуру грід-сервіс для рішення задач оптимального моделювання схемотехнічних рішень, для яких вказані недоліки не є вагомими.

Хоча головними перевагами даної архітектури, що автоматизує виконання багатокрокових обчислень у гріді, є, в першу чергу, гнучкість, відповідність стандартам, використання існуючої інфраструктури, а не мінімізація часу розрахунків, було проведено оцінку часу виконання потоків робіт у даній архітектурі, додатково підтверджену експериментально. Слід зазначити, що порівняльні експерименти проводились за умов відсутності конкуруючих задач, що відрізняється від реальних умов динамічного грід-середовища, що слід брати до уваги при застосуванні отриманих оцінок.

Проведені дослідження дозволяють стверджувати, що дана архітектура найкраще підходить до вирішення довготривалих обчислювальних задач. Накладні витрати на організацію виконання потоку робіт роблять недоцільним запуск короткотривалих задач із тривалістю виконання порядку кількох хвилин (для яких, як правило, достатньо потужностей локальних ресурсів). Дане рішення з автоматизованого виконання складних обчислювальних сценаріїв може розглядатися як складова архітектури спеціалізованих високорівневих грід-середовищ та грід-порталів.

Табл. 1. Результати тестів для короткотривалих задач

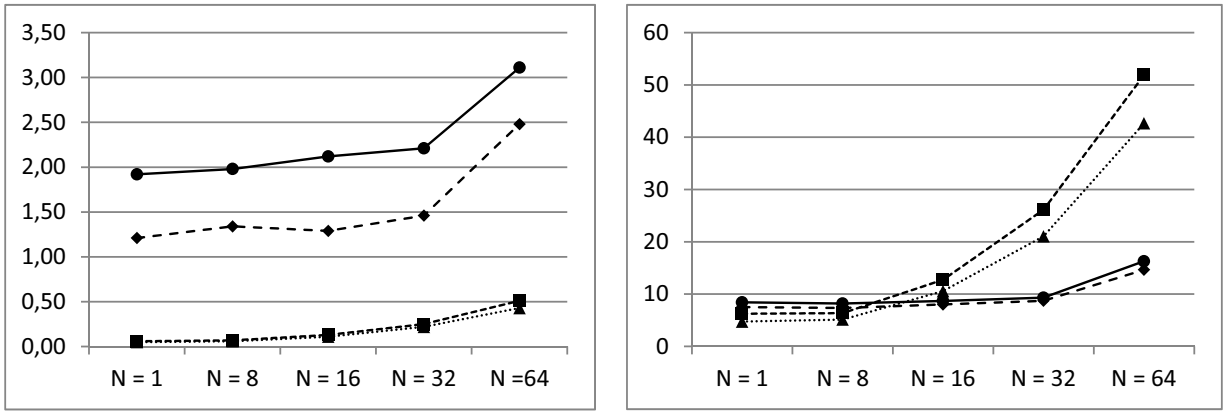
Спосіб запуску	**Середній час виконання потоку задач, хв.**				
	N = 1	N = 8	N = 16	N = 32	N =64
Локальне виконання @ 2.33 ГГц	0,06	0,07	0,13	0,25	0,51
Локальне виконання @ 2.83 ГГц	0,05	0,06	0,11	0,22	0,43
Запуск у грід (скрипт)	1,21	1,34	1,29	1,46	2,48
Запуск потоку сервісів	1,92	1,98	2,12	2,21	3,11

Табл. 2. Результати тестів для задач середньої тривалості

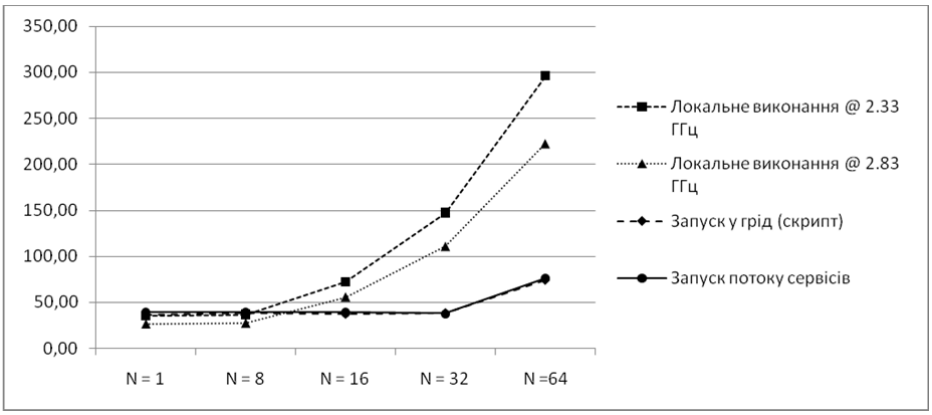
Спосіб запуску	**Середній час виконання потоку задач, хв.**				
	N = 1	N = 8	N = 16	N = 32	N = 64
Локальне виконання @ 2.33 ГГц	6,25	6,36	12,75	26,13	51,89
Локальне виконання @ 2.83 ГГц	4,73	5,12	10,58	21,04	42,66
Запуск у грід (скрипт)	7,51	7,32	8,01	8,72	14,64
Запуск потоку грід-сервісів	8,42	8,18	8,69	9,31	16,27

Табл. 3. Результати тестів для довготривалих задач

Спосіб запуску	**Середній час виконання потоку задач, хв.**				
	N = 1	N = 8	N = 16	N = 32	N =64
Локальне виконання @ 2.33 ГГц	35,33	36,58	71,97	147,15	295,73
Локальне виконання @ 2.83 ГГц	26,69	27,65	55,92	110,79	222,21
Запуск у грід (скрипт)	36,18	37,82	36,8	38,42	74,13
Запуск потоку сервісів	38,85	38,91	38,87	37,81	75,91



а *б*



в

Рис. 3. Час виконання потоку грід-задач (хвилин): а – короткотривалих, б – середньої тривалості, в – довготривалих

Список літератури

1. Згуровський М. З. Grid-технології для е-науки і освіти / Згуровський М. З., Петренко А. І. // Наукові вісті НТУУ «КПІ». – 2009. – №2. – С. 10–17 .
2. Erl T. Service-Oriented Architecture: Concepts, Technology & Design / Erl T. – New York : Prentice Hall/PearsonPTR, 2005. – 792 p.
3. Workflows for e-Science. Scientific Workflows for Grids / Edited by I.J. Taylor, E. Deelman, D.B. Gannon, M. Shields. – Guildford : Springer, 2007. – 530 p.
4. Yu J. A Taxonomy of Workflow Management Systems for Grid Computing / Yu J., Buyya R. // Journal of Grid Computing. – 2005. – Vol. 3, № 3. – P. 171–200.
5. Talia D. The Open Grid Services Architecture: Where the Grid Meets the Web // IEEE. Internet Computing. – 2002. – Vol. 6, № 6. – P. 67–71.
6. Petrenko A. ALLTED – a computer-aided engineering system for electronic circuit design / Petrenko A., Ladogubets V., Tchkalov V., Pudlowski Z. – Melbourne : UICEE, 1997. – 205 p.
7. Franz D. A Workflow Engine for Computing Clouds / Franz D., Tao J., Marten H., Streit A. // Cloud Computing, GRIDs, and Virtualization : 2nd International Conference «CLOUD COMPUTING 2011», 25-30 Sep. 2011, Rome, Italy : proc. – 2011. – P. 1–6. – ISBN : 9780470940105.