

УДК 000.000.00

Національна академія наук України  
Назва інституту

ПРІЗВИЩЕ Ім'я По-батькові

Розвиток архітектури та методів автоматичної компіляції дисертацій  
у середовищі LaTeX

Дисертація  
на здобуття наукового ступеня кандидата технічних наук  
за спеціальністю 05.13.05 – комп'ютерні системи та компоненти

*Науковий керівник*  
доктор фізико-математичних наук  
професор  
ПІБ Керівника

Київ – 2018

На правах рукопису

## ЗМІСТ

	С.
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ . . . . .	3
ВСТУП . . . . .	5
1 АНАЛІЗ АРХІТЕКТУР ТА БАЗОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕН- НЯ ВИСОКОПРОДУКТИВНИХ ОБЧИСЛЮВАЛЬНИХ КОМПЛЕКСІВ .	8
1.1 Архітектура багатопроцесорних обчислювальних систем . . . . .	8
1.1.1 Системи з масовим паралелізмом (МРР) . . . . .	10
1.1.2 Кластерні системи . . . . .	11
1.1.3 Архітектура типу сузір'я . . . . .	13
1.2 Висновки . . . . .	14
ЗАГАЛЬНІ ВИСНОВКИ . . . . .	15
ПЕРЕЛІК ПОСИЛАНЬ . . . . .	16
А РЕАЛІЗАЦІЯ МОДУЛЯ ЗБОРУ СТАТИСТИКИ . . . . .	19
Б АКТИ ВПРОВАДЖЕННЯ . . . . .	22

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

**AMD64/EM64T** — 64-розрядна мікропроцесорна архітектура

**BIOS** — базова система вводу/виводу

**ccNUMA** (англ. cache coherent Non Uniform Memory Access) — системи з неоднорідним доступом до пам'яті та когерентним кешем)

**CPU** — центральний процесор

**DDR** — подвійна швидкість передачі даних мережі Infiniband

**GPU** — графічний процесор

**IBM** — американська корпорація, виробник обчислювальної техніки

**Infiniband, Myrinet, Quadrix** — високопродуктивні мережі передачі даних

**IP** (англ. Internet protocol) — мережевий протокол

**IPMI** — інтелектуальний інтерфейс управління платформою

**ISA** — шина вводу/виводу IBM персональних комп'ютерів

**LDAP** — протокол доступу до каталогів

**Linpack** — бібліотека програм для тестування комп'ютерів

**Linux** — UNIX- подібна операційна система

**Lustre** — розподілена файлова система

**MPI** (англ. Message Passing Interface) — інтерфейс обміну даними між обчислювальними процесами

**MPP** (англ. Massivlly Parallel Processing) — паралельна обчислювальна система з масивом процесорів

**NFS** — мережева файлова система

**NUMA** (англ. Non Uniform Memory Access) — системи з неоднорідним доступом до пам'яті

**OpenMP** — мова та система паралельного програмування із застосуванням мов FORTRAN (77, 90 і 95), C та C++

**PAM** — модулі аутентифікації операційної системи Linux

**RAID** — надлишковий масив незалежних дисків

**RAM** — запам'ятовуючий пристрій з довільним доступом

**SLURM** — менеджер ресурсів кластера

**UID** — числовий ідентифікатор користувача операційної системи

**БПЗ** — базове програмне забезпечення

**ЕОМ** — електронна обчислювальна машина

**ІК НАНУ** — Інститут кібернетики ім. В.М. Глушкова НАН України

**МОС** — мультипроцесорна обчислювальна система

**ОС** — операційна система

**ПЕ** — процесорний елемент

**ПЗ** — програмне забезпечення

**СКЗ** — система керування задачами

**СКІТ** — суперкомп'ютер інформаційних технологій

**суперкомп'ютер** — обчислювальна система, яка має визначні характеристики продуктивності у порівнянні з комп'ютерами загального вжитку, у роботі терміни “суперкомп'ютер”, “обчислювальний кластер”, “обчислювальний комплекс” використовуються як синоніми

**гібридний вузол** — обчислювальний вузол, в якому для обрахунків використовуються як універсальні процесори, так і спеціалізовані акселератори

## ВСТУП

**Актуальність теми.** Суперкомп'ютерні технології та системи є одним з головних чинників, що визначають конкурентоспроможність економіки держави та рівень її технологічного розвитку. Без їх використання неможливо створювати сучасну техніку, нові матеріали, лікарські засоби, моделювати та прогнозувати екологічні, технологічні та інші загрози. Понад 60% світових високопродуктивних обчислювальних ресурсів використовують для розв'язання складних задач економіки, управління фінансами, промисловості, нафтогазової та інженерно-геологічної галузей, решта – орієнтовані на задачі обороноздатності та науки.

Аналіз світових тенденцій застосування супер-ЕОМ у Росії, США, ЄС та інших країнах свідчить, що цей напрямок є одним з пріоритетних.

В Україні, яка має високий науково-технічний потенціал у галузі створення сучасних високопродуктивних інтелектуальних інформаційних технологій, створено ряд кластерних комплексів, які застосовуються при розв'язанні складних задач господарського комплексу, науки та промисловості.

**Зв'язок роботи з науковими програмами, планами, темами.** Дисертаційна робота виконувалась у рамках наступних держбюджетних дослідницьких тем:

**Мета і завдання дослідження.** Метою роботи є подальший розвиток архітектури та базового програмного забезпечення високопродуктивних обчислювальних комплексів за рахунок використання сучасних технологій віртуалізації обчислювальних систем та застосування графічних прискорювачів.

Основні задачі дослідження:

- розробка архітектури;
- побудова методики;
- побудова моделі;
- розробка методів;

*Об'єкт дослідження* – процес проектування високопродуктивних обчислювальних комплексів з кластерною архітектурою.

*Предмет дослідження* – архітектура та базове програмне забезпечення високопродуктивних обчислювальних комплексів.

*Методи дослідження.* Для розв'язання сформульованих задач у дисертаційній

роботі використовуються результати та методи загальної теорії систем, теорії інформації, теорії алгоритмів, теорії ймовірностей, математичної статистики та математичного аналізу.

**Наукова новизна отриманих результатів.** Основними результатами, які визначають наукову новизну і виносяться на захист, є наступні:

- удосконалено;
- вперше запропоновано;
- розвинуто методи.

**Практичне значення отриманих результатів.** Отримані в дисертаційній роботі результати мають практичне значення, і є внеском у ... Розв’язані задачі в області архітектури дозволяють зменшити вартість побудови, знизити енергоспоживання, збільшити надійність та продуктивність СКА.

Практичне значення визначають наступні результати:

- розроблено;
- розвинуто;
- досліджено.

Результати роботи використані при ....

**Особистий внесок здобувача.** Усі наукові результати дисертаційної роботи отримані автором особисто, вони опубліковані в 9 статтях у фахових журналах. У працях, виконаних у співавторстві, здобувачу належать такі результати: [1, 2] – методика побудови системи керування гетерогенним кластерним комплексом.

**Апробація результатів.** Основні результати досліджень доповідались на міжнародних наукових конференціях і обговорювались на наукових семінарах:

12, 13 Міжнародній науковій конференції “Knowledge-Dialogue-Solution” (Варна, Болгарія, 2006, 2007); 4, 6 Міжнародній науковій конференції “Искусственный интеллект, Интеллектуальные многопроцессорные системы” (смт. Кацивелі, Крим, Україна, 2006, 2008); науковій молодіжній міжнародній школі “Высокопроизводительные вычислительные системы” (смт. Кацивелі, Крим, Україна, 2006);

**Публікації.** Основні результати дисертації опубліковано в 20 наукових працях. З них 9 – у фахових виданнях та одинадцять тез доповідей міжнародних наукових конференціях.

**Структура та обсяг роботи.** Дисертаційна робота складається із вступу, чотирьох розділів, розбитих на підрозділи, висновків, 4-х додатків (додаток Г містить акти впровадження дисертаційної роботи) та списку використаних джерел, що містить

112 найменувань. Повний обсяг роботи – 183 сторінки.

## **ОСНОВНИЙ ЗМІСТ РОБОТИ**

Далі короткий текст по основним результатам.

Автор висловлює щиру вдячність Важливим помічникам за постановку задачі та постійну увагу до роботи.

# 1 АНАЛІЗ АРХІТЕКТУР ТА БАЗОВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВИСОКОПРОДУКТИВНИХ ОБЧИСЛЮВАЛЬНИХ КОМПЛЕКСІВ

## 1.1 Архітектура багатопроцесорних обчислювальних систем

У процесі розвитку суперкомп'ютерних технологій ідею підвищення продуктивності обчислювальної системи за рахунок збільшення кількості процесорів використовували неодноразово.

Експериментальні розробки зі створення багатопроцесорних обчислювальних систем почалися в 70-х роках XX століття. Однією з перших таких систем стала розроблена в університеті Іллінойса багатопроцесорна обчислювальна система ILLIAC IV [21], яка включала 64 (у проекті до 256) процесорних елементів (ПЕ), що працюють за єдиною програмою, яка застосовується до вмісту власної оперативної пам'яті кожного ПЕ.

Обмін даними між процесорами здійснювався через спеціальну матрицю комунікаційних каналів. Вказана особливість комунікаційної системи дала назву "Матричні суперкомп'ютери" відповідному класу мультипроцесорних обчислювальних систем (МОС). Ширший клас МОС з розподіленою пам'яттю і з довільною комунікаційною системою отримав надалі назву "Багатопроцесорні системи з масовим паралелізмом", або МОС з MPP-архітектурою. При цьому, як правило, кожен з ПЕ MPP системи є універсальним процесором, що діє за своєю власною програмою (на відміну від спільної програми для всіх ПЕ матричної МОС).

Перші матричні МОС випускалися буквально поштучно, тому їх вартість була надзвичайно високою. Серійні ж зразки подібних систем, такі як ICL DAP, що включали до 8192 ПЕ, з'явилися значно пізніше, проте не набули широкого поширення, зважаючи на складність програмування МОС з одним потоком управління (з однією програмою, спільною для всіх ПЕ). Перші промислові зразки мультипроцесорних систем з'явилися на базі векторно-конвеєрних комп'ютерів у середині 80-х років XX ст.

Найбільш поширеними МОС такого типу були суперкомп'ютери фірми Cray. Проте такі системи були надзвичайно дорогими і виготовлялися невеликими серіями.



Як правило, в подібних комп'ютерах об'єднувалося від 2 до 16 процесорів, які мали рівноправний (симетричний) доступ до спільної оперативної пам'яті. У зв'язку з цим вони отримали назву симетричні мультипроцесорні системи (Symmetric Multi-Processing – SMP).

Як альтернатива таким дорогим мультипроцесорним системам на базі векторно-конвеєрних процесорів була запропонована ідея будувати еквівалентні за потужністю багатопроцесорні системи з великої кількості дешевих мікропроцесорів, що серійно випускалися.

Проте дуже скоро виявилось, що SMP архітектура має досить обмежені можливості з нарощування кількості процесорів в системі через різке збільшення числа конфліктів при зверненні до спільної шини пам'яті. У зв'язку з цим виправданою представлялася ідея забезпечити кожен процесор власною оперативною пам'яттю, перетворюючи комп'ютер на об'єднання незалежних обчислювальних вузлів.

Такий підхід значно збільшив міру масштабованості багатопроцесорних систем, але у свою чергу почав вимагати розробки спеціального способу обміну даними між обчислювальними вузлами, що реалізовується зазвичай у вигляді механізму передачі повідомлень (Message Passing).

Комп'ютери з такою архітектурою є найбільш яскравими представниками MPP систем. У даний час ці два напрямки (або деякі їх комбінації) є домінуючими в розвитку суперкомп'ютерних технологій.

Чимось середнім між SMP і MPP є NUMA-архітектури, в яких пам'ять фізично розділена, але логічно загальнодоступна. При цьому час доступу до різних блоків пам'яті стає неоднаковим. У одній з перших систем цього типу Cray T3D час доступу до пам'яті іншого процесора був у 6 разів більше, ніж до своєї власної.

На даний час розвиток суперкомп'ютерних технологій йде чотирма основними напрямками:

- MPP системи;
- кластерні системи;
- архітектура типу сузір'я.

Розглянемо основні особливості перерахованих архітектур.

### 1.1.1 Системи з масовим паралелізмом (MPP)

Проблеми, притаманні мультипроцесорним системам зі спільною пам'яттю, природним чином усуваються в системах з масовим паралелізмом. Комп'ютери цього типу є багатопроцесорними системами з розподіленою пам'яттю, в яких, за допомогою деякого комунікаційного середовища, об'єднуються однорідні обчислювальні вузли.

Кожен з вузлів складається з одного або кількох процесорів, власної оперативної пам'яті, комунікаційного обладнання, підсистеми введення/виведення, тобто має все необхідне для незалежного функціонування.

При цьому на кожному вузлі може функціонувати або повноцінна операційна система (як в системі RS/6000 SP2), або урізаний варіант, що підтримує лише базові функції ядра, а повноцінна ОС працює на спеціальному керівному комп'ютері (як в системах Cray T3E, nCUBE2).

Процесори в таких системах мають прямий доступ лише до своєї локальної пам'яті. Доступ до пам'яті інших вузлів реалізується, зазвичай, за допомогою механізму передачі повідомлень. Така архітектура обчислювальної системи усуває одночасно як проблему конфліктів при зверненні до пам'яті, так і проблему когерентності кеш-пам'яті.

Це дає можливість практично необмеженого нарощування кількості процесорів в системі, збільшуючи тим самим її продуктивність. Успішно функціонують MPP з сотням і тисячами процесорів (ASCI White – 8192, Blue Mountain – 6144).

Важливою властивістю MPP систем є їх висока міра масштабованості. Залежно від обчислювальних потреб для досягнення необхідної продуктивності потрібно просто зібрати систему з потрібною кількістю вузлів.

На практиці все, звичайно, набагато складніше. Усунення одних ускладнень, як це зазвичай буває, породжує інші. Для MPP систем насамперед виникає питання ефективності комунікаційного середовища. Різні виробники MPP систем використовували різні топології. У комп'ютерах Intel Paragon процесори утворювали прямокутну двовимірну сітку. Для цього в кожному вузлі досить чотирьох комунікаційних каналів. У комп'ютерах Cray T3D/T3E використовувалася топологія тривимірного тора. Відповідно, у вузлах цього комп'ютера було шість комунікаційних каналів. Фірма nCUBE використовувала в своїх комп'ютерах топологію n-вимірного гіперкуба.

Кожна з розглянутих топологій має свої переваги і недоліки. Зазначимо, що при обміні даними між процесорами, які не є найближчими сусідами, відбувається трансляція даних через проміжні вузли. Очевидно, що у вузлах мають бути передбачені якісь апаратні засоби, які звільняли б центральний процесор від участі в трансляції даних.

Останнім часом для з'єднання обчислювальних вузлів частіше використовується ієрархічна система високошвидкісних комутаторів, як це вперше було реалізовано в комп'ютерах IBM SP2. Така топологія дає можливість прямого обміну даними між будь-якими вузлами, без участі в цьому проміжних вузлів.

Системи з розподіленою пам'яттю ідеально підходять для паралельного виконання незалежних програм, оскільки при цьому кожна програма виконується на своєму вузлі і жодним чином не впливає на виконання інших програм. Проте при розробці паралельних програм доводиться враховувати складнішу, ніж в SMP системах, організацію пам'яті. Оперативна пам'ять в MPP системах має 3-х рівневу структуру:

- кеш-пам'ять процесорів;
- локальна оперативна пам'ять;
- оперативна пам'ять інших вузлів.

При цьому відсутня можливість прямого доступу до даних, розташованих в інших вузлах. Для їх використання ці дані мають бути заздалегідь передані в той вузол, який в даний момент їх потребує. Це значно ускладнює програмування. Крім того, обмін даними між вузлами виконується значно повільніше, ніж обробка даних в локальній оперативній пам'яті вузлів. Тому написання ефективних паралельних програм для таких комп'ютерів є складнішим завданням, ніж для SMP систем.

### 1.1.2 Кластерні системи

Кластерні технології стали логічним продовженням розвитку ідей, закладених в архітектурі MPP систем.

Якщо процесорний модуль в MPP системі є закінченою обчислювальною системою, то наступний крок напрошується сам собою: чому б в якості обчислювальних вузлів не використовувати звичайні комп'ютери, що серійно випускаються [22].

Розвиток комунікаційних технологій, а саме, поява високошвидкісного мережевого обладнання і спеціального програмного забезпечення, такого як система MPI, що реалізовує механізм передачі повідомлень над стандартними мережевими протоколами, зробили кластерні технології загальнодоступними.

Сьогодні не складає великих труднощів створити невелику кластерну систему, об'єднавши обчислювальні потужності комп'ютерів окремої лабораторії або учбового класу. Привабливою рисою кластерних технологій є те, що вони дозволяють для досягнення необхідної продуктивності, об'єднувати в єдині обчислювальні системи комп'ютери різного типу, починаючи від персональних комп'ютерів і закінчуючи потужними суперкомп'ютерами.

Широкого поширення кластерні технології набули як засіб створення систем суперкомп'ютерного класу зі складових частин масового виробництва, що значно знижує вартість обчислювальної системи. Звичайно, про повну еквівалентність цих систем говорити не доводиться. Як вказувалося в попередньому розділі, продуктивність системи з розподіленою пам'яттю дуже сильно залежить від продуктивності комунікаційного середовища.

Комунікаційне середовище можна досить повно охарактеризувати двома параметрами: латентністю – часом затримки при відсиланні повідомлення та пропускну здатністю – швидкістю передачі інформації. Так для комп'ютера Cray T3D ці параметри складають відповідно 1 мкс і 480 мб/с, а для кластера, у якому в якості комунікаційного середовища використана мережа Fast Ethernet, – 100 мкс і 10 мб/с відповідно. Це частково пояснює дуже високу вартість суперкомп'ютера. При таких параметрах, як у розглянутого кластера, знайдеться не так багато задач, які можна ефективно вирішувати на досить великій кількості процесорів.

Коротко кажучи, кластер – це зв'язаний набір повноцінних комп'ютерів, що використовується як єдиний обчислювальний ресурс. Переваги кластерної системи перед набором незалежних комп'ютерів очевидні.

По-перше, розроблено безліч диспетчерських систем пакетної обробки завдань, що дозволяють відправити завдання на обробку кластера вцілому, а не якомусь окремому комп'ютеру. Ці диспетчерські системи автоматично розподіляють завдання за вільними обчислювальними вузлами або буферизують їх за відсутності таких, що дозволяє забезпечити більш рівномірне і ефективне завантаження комп'ютерів. По-друге, з'являється можливість спільного використання обчислювальних ресурсів декількох комп'ютерів для вирішення одного завдання. Для створення кластерів,

зазвичай, використовуються або прості однопроцесорні персональні комп'ютери, або двох- чи чотирипроцесорні SMP-сервери [10, 3].

При цьому не накладається жодних обмежень на склад і архітектуру вузлів. Кожен з вузлів може функціонувати під управлінням своєї власної операційної системи. Найчастіше використовуються стандартні ОС: Linux, FreeBSD, Solaris, Tru64 Unix, Windows.

У тих випадках, коли вузли кластера неоднорідні, то говорять про гетерогенні кластери [1, 2].

При створенні кластера можна виділити два підходи. Перший підхід застосовується при створенні невеликої кластерної системи. У кластер об'єднуються повнофункціональні комп'ютери, які продовжують працювати і як самостійна одиниця, наприклад, комп'ютери учбового класу або робочі станції лабораторії, як у навчальній системі UACLUSTR [23].

Другий підхід застосовується в тих випадках, коли цілеспрямовано створюється потужний обчислювальний ресурс. Тоді системні блоки комп'ютерів компактно розміщуються в спеціальній стійках, а для управління системою і для запуску задач виділяється один або декілька повнофункціональних комп'ютерів, так званих хост-комп'ютерів. У цьому випадку немає необхідності забезпечувати комп'ютери обчислювальних вузлів графічними картами, моніторами, дисковими накопичувачами та іншим периферійним обладнанням, що значно знижує вартість системи.

### 1.1.3 Архітектура типу сузір'я

Сузір'я (Constellation) – це кластер великих SMP вузлів, в якому кількість процесорів на вузол більше, ніж кількість вузлів.

Сьогодні для кластера стандарним є вузол з 16 ядрами. У системах типу сузір'я використовуються вузли із значно більшою кількістю ядер, SMP-системи з 100–1000 ядрами.

Також, згідно визначення, до архітектури сузір'я відносяться гібридні вузли з графічними процесорами, оскільки кожен такий вузол містить кілька сотень процесорних елементів.

Загальна пам'ять, доступна всім процесам задачі у великому SMP-вузлі, до-

зволяє ефективно вирішувати тісно пов'язані задачі з великою кількістю обмінів даними між процесами. Це не під силу звичайним кластерам, на яких такі завдання виконуються вкрай неефективно.

Прикладами задач, для яких кластери поступаються архітектурі Constellation, є прогнозування погоди, моделювання клімату, розрахунок щільності, задачі теплопереносу.

## 1.2 Висновки

Сучасні суперкомп'ютерні архітектури є основою для побудови нових поколінь обчислювальних систем, які дозволять вийти на нові рубежі продуктивності суперкомп'ютерів.

У розділі зроблено порівняльний аналіз основних різновидів архітектур суперкомп'ютерів, їх розвиток від матричних суперкомп'ютерів до сучасних кластерів та суперкомп'ютерів з масовим паралелізмом, показано, чому кластерна архітектура є на сьогодні домінуючою при побудові суперкомп'ютерів.

Також проаналізовано гібридні обчислювальні системи, нових перспективний тип архітектур обчислювальних вузлів, які дозволили суперкомп'ютерній індустрії подолати петафлопсний бар'єр, проаналізовані їх переваги та недоліки.

У розділі проаналізовані питання створення базового програмного забезпечення суперкомп'ютерів для систем з різними типами архітектури.

Основні наукові результати розділу опубліковані в працях ...

## ЗАГАЛЬНІ ВИСНОВКИ

Головним результатом дисертаційної роботи є розв’язання науково-технічної задачі проектування архітектури та розвитку базового програмного забезпечення високопродуктивних обчислювальних комплексів з кластерною архітектурою.

Проведено експериментальні та теоретичні дослідження, які дозволили зробити наступні висновки:

1. Розроблено архітектуру та розвинуто.
2. Удосконалено.
3. Запропоновано.
4. Побудовано моделі.
5. Розроблено методи.

Результати дисертації впровадженно в архітектурі та базовому програмному забезпеченні суперкомп’ютера. Широке впровадження результатів дисертаційного дослідження показало ефективність сформульованих науково-технічних положень.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Головинский А.Л. Гетерогенный кластерный комплекс Института кибернетики НАН Украины: средства построения / Головинский А.Л., Рябчун С.Г., Якуба А.А. // Искусственный интеллект. — 2006. — Т. 3. — С. 107–112.
2. Головинский А.Л. Средства построения гетерогенного кластерного комплекса / Головинский А.Л., Рябчун С.Г., Якуба А.А. // Изв. Таганрогского государственного радиотехнического университета. — 2006. — Т. 71, № 16. — С. 4–11.
3. Golovynskyi A.L. Cluster supercomputer architecture / Golovynskyi A.L., Ryabchun S.G., Yakuba A.A. // Information Theories and Applications. — 2007. — Vol. 14. — P. 374–381.
4. Головинский А.Л. Проблемы политики безопасности кластерного комплекса / Головинский А.Л., Рябчун С.Г., Якуба А.А. // Управляющие системы и механизмы. — 2007. — Т. 4. — С. 22–27.
5. Горенко С.А. Построение суперкомпьютера кластерной архитектуры без использования сети ethernet / Горенко С.А., Головинский А.Л., Рябчун С.Г. // Искусственный интеллект. — 2008. — Т. 4. — С. 643–647.
6. Бандура А.Ю. Построение высокопроизводительной дисковой подсистемы для суперкомпьютеров кластерной архитектуры / Бандура А.Ю., Головинский А.Л., Рябчун С.Г. // Искусственный интеллект. — 2008. — Т. 3. — С. 611–618.
7. Портал кластерных вычислений для управления вычислительными процессами на суперкомпьютерном комплексе / Якуба А.А., Головинский А.Л., Бандура А.Ю. и др. // Кибернетика и системный анализ. — 2009. — Т. 6. — С. 97–105.
8. Головинський А.Л. Графічний інтерфейс суперкомп'ютера на основі веб-технологій / Головинський А.Л. // Управляющие системы и механизмы. — 2010. — Т. 3. — С. 54–58.
9. Головинский А.Л. Веб-портал системы управления суперкомпьютером / Головинский А.Л., Белоус Л.Ф., Маленко А.Л. // Вычислительные методы и программирование. — 2010. — Т. 11/7.
10. Golovynskyi A.L. Cluster supercomputer architecture / Golovynskyi A.L., Ryabchun S.G., Yakuba A.A. // Proceedings of the Intern. Conf. KDS'2006. — 2006. — P. 112–118.



11. Golovynskyi A.L. Safety policy problems of cluster supercomputers / Golovynskyi A.L., Ryabchun S.G., Yakuba A.A. // Proceedings of the Intern. Conf. KDS'2007. — Varna, Bulgaria, 2007.
12. Рябчун С.Г. Перспективы развития суперкомпьютеров кластерной архитектуры в Украине / Рябчун С.Г., Головинский А.Л. // Праці конференції "50 років інституту кібернетики імені В.М. Глушкова НАН України". — 2008. — С. 236–241.
13. Компьютерное моделирование электронной и молекулярной структуры фрагментов М-ДНК: стекнинг димеров и тримеров / Рубин Ю.В., Белоус Л.Ф., Головинский А.Л. и др. // Труды международной научной конференции "Параллельные вычислительные технологии ПаВТ'2009". — Челябинск: Изд. ЮУрГУ, 2009.
14. Головинський А.Л. Система керування обчислювальним процесом суперкомп'ютера та ґриду / Головинський А.Л., Якуба А.О. // Тези доповідей міжнародної конференції "Теоретичні та прикладні аспекти побудови програмних систем ТАAPSD'2009". — 2009. — С. 81–82.
15. Головинський А.Л. Аналіз статистики використання суперкомп'ютера СКІТ Інституту кібернетики / Головинський А.Л., Бандура О.Ю., Якуба А.О. // Тези доповідей міжнародної конференції "Теоретичні та прикладні аспекти побудови програмних систем ТАAPSD'2009". — 2009. — С. 89–91.
16. Головинский А.Л. Применение суперкомпьютерных технологий / Головинский А.Л., Маленко А.Л. // Зб. наукових праць міжнародної конференції "Сучасні інформаційні та інноваційні технології на транспорті MINTT-2010". — Т. 1. — Херсон, 2010. — С. 42–46.
17. Головинский А.Л. Веб-портал системы управления суперкомпьютером / Головинский А.Л., Белоус Л.Ф. // Сб. трудов XII международной конференции серии "Научный сервис в сети Интернет". — п. Абрау-Дюрсо, Россия, 2010.
18. Головинський А.Л. Аналіз ефективності планувальників черги задач для суперкомп'ютера з кластерною архітектурою / Головинський А.Л., Маленко А.Л. // Матеріали конференції "Високопродуктивні обчислення НРС-UA 2011". — 2011. — С. 64–69.
19. Архітектура ґри вузлів обчислювального кластера / Головинський А.Л., Маленко А.Л., Горенко С.О., Бандура О.Ю. // Матеріали конференції "Високопродуктивні обчислення НРС-UA 2011". — 2011. — С. 70–75.
20. Головинський А.Л. Система керування суперкомп'ютером з підтримкою роботи

у гріді scms 4.0 / Головинський А.Л., Маленко А.Л., Черепинець В.В. // Матеріали міжнародного наукового конгресу з розвитку інформаційно-комунікаційних технологій та розбудови інформаційного суспільства в Україні. — 2011. — С. 28–29.

21. Іліас 4. — портал [Електронний ресурс], дата звертання: 10.11.2018. — Режим доступу: <http://ed-thelen.org/comp-hist/vs-illiac-iv.html>.
22. Кластерна архітектура beowulf. — портал [Електронний ресурс], дата звертання: 10.11.2018. — Режим доступу: <http://www.beowulf.org/>.
23. Программное обеспечение ucluster / Марьяновский В.А., Погорелый С.Д., Бойко Ю.В. и др. // Управляющие системы и механизмы. — 2009. — Т. 5. — С. 76–80.

## ДОДАТОК А РЕАЛІЗАЦІЯ МОДУЛЯ ЗБОРУ СТАТИСТИКИ

Для відображення даних з черги задач СКЗ та виконання розрахунку статистики роботи кластера розроблений Perl-модуль системи порталу кластерних обчислень. Цей модуль виконує обробку вхідного текстового лог-файлу менеджера ресурсів SLURM, а на виході формує оброблені дані черги задач, відфільтрованих за певном критерієм та обчислює статистичні показники, розглянуті вище. Модуль працює в автономному режимі (запуск за запитом з консолі) та доданий до функціоналу порталу кластерних обчислень.

Perl-модуль використовує файл конфігурації підсистеми hydra порталу кластерних обчислень, розміщену в `/opt/hydra/etc/hydra.conf`. Конфігураційні дані відображені на рис. А.1.

```
##Job-log section
#Path for job-log file or pipe command "cmd|"
JOB_LOG_FILE="ssh n1000 cat /var/log/slurm.job.log|"
#Identifie usernamemes from uid's in job-log, can by "yes" or "no",
LOG_USER_IDENT="yes"
```

Рисунок А.1 — Конфігурація Perl-модуля статистики роботи кластера

Параметр **JOB\_LOG\_FILE** вказує шлях до вхідного лог-файла SLURM або є конвеєрною командою, якщо лог-файл знаходиться на іншому сервері. Параметр **LOG\_USER\_IDENT** вказує ідентифікувати чи ні імена користувачів по їхнім UID. Керівний демон SLURM кластера SKIT `slurmctld` функціонує на сервері, який з метою безпеки не забезпечує аутентифікацію користувачів через LDAP. У наслідок чого `slurmctld` не може по UID користувача, що запустив задачу, визначити його імені. Через це в лог-файлі черги задач фігурують тільки UID користувачів, а ім'я ідентифікується як `Unknown`. Саме для таких випадків і потрібна ця опція. Perl-модуль напряду з'єднується з LDAP-сервером, та отримує від нього необхідну інформацію. Параметри підключення беруться з файла адміністративної конфігурації підсистеми hydra порталу кластерних обчислень, розміщену у `/opt/hydra/etc/hydra-adm.conf`. Параметри налаштування роботи з LDAP-сервером відображені на рис. А.2.

Додатково модуль виконує обрахунок тривалості кожної задачі за часом її запуску та завершення. Цей параметр використовується у розрахунках статистики, а також виводиться у форматі "дні – години : хвилини : секунди" при відображенні

інформації з черги задач. Модуль може відфільтровувати чергу задач за наступними параметрами:

- початкова та кінцева дата, в межах року, що задають період, за яким відображатиметься інформація про завершені задачі;
- ім'я користувача (у профілях користувачів порталу кластерних обчислень завжди виводиметься інформація, відфільтрована цим критерієм);
- статус завершення задачі.

Особливістю ведення журналу задач системи SLURM є відсутність значення року у датах запуску та завершення задач, тому необхідно розділяти лог-файли за роками.

Приклад результируючого виводу стану завершених задач, придатного для відображення в табличному вигляді системою порталу кластерних обчислень, зображений на рис. А.3.

Розрахована статистика відображається як для окремого користувача, так і в загальному по кластеру. Системним адміністраторам порталу доступне відображення статистики за всіма користувачами одночасно. Приклад відображення статистики використання кластера за січень поточного року зображено на рис. А.4.

```
# User database management system LDAP,PASSWD
USERDB=LDAP
# LDAP Parameters
LDAP_HOST=10.0.0.254
#LDAP_SEARCH_BASE="dc=cluster,dc=icyb"
LDAP_SEARCH_BASE="ou=People,dc=cluster,dc=icyb"
LDAP_ADM_DN="cn=ldapadm,dc=cluster,dc=icyb"
LDAP_ADM_PASSWD="password"
```

Рисунок А.2 — Конфігурація Perl-модуля статистики роботи кластера для роботи з базою LDAP

```
22988 solo(15026) run.ampi run.ampi COMPLETED scit3 20000 11/02-13:29:45
11/02-13:29:47 0-0:0:2 n3137 1 8
22799 ru(5005) part2 part2 COMPLETED scit3 20000 10/27-13:30:09
11/02-14:10:55 6-0:40:46 n[3106-3112,3143] 8 64
22990 yershov(5599) btree btree FAILED scit3 120 11/02-14:10:56
11/02-14:15:34 0-0:4:38 n[3106-3112,3143] 8 64
22991 vgor(1111) gms gms COMPLETED scit3 20000 11/02-14:15:36
11/02-14:38:43 0-0:23:7 n[3106-3109] 4 32
22987 engraver(1255) D78h26_searchField D78h26_searchField
CANCELLED scit3 300 11/02-12:43:48 11/02-15:43:04 0-2:59:16 n3119 1 8
22996 engraver(1255) vortexD78h26 vortexD78h26 CANCELLED scit3 600
11/02-15:43:16 11/02-16:01:55 0-0:18:39 n3119 1 8
```

Рисунок А.3 — Розпарсений вивід стану завершених задач

```
# ./ic_get_jobstat -s -b 01/01 -e 02/01
USER:JobCount,JobRatio(%),CD,CA,F,NF,T0,AvgJobTime(min),Load(TotalTime(min)*NCPU),LoadRatio(%)
Chobal: 30      1.35%  27      2      0      0      1      1552      1477135  27.41%
akn: 61      2.75%  50      5      6      0      0      5      2463      0.05%
alex: 124     5.58%  90     10     15     0      9      0      463      0.01%
belous: 43    1.94%  35      8      0      0      0      360     136676   2.54%
dept125:62    2.79%  27      9     26     0      0      0      3      0.00%
dept175:400   18.00% 193    145    62     0      0      9     25911   0.48%
glad: 7      0.32%  4       0      2      0      1      1      98      0.00%
ka: 55      2.48%  48      4      3      0      0      2     987      0.02%
kaa: 201     9.05% 188      7      6      0      0     24    742924  13.79%
karpman:13    0.59%  4       6      0      0      3      5     1064     0.02%
logvina:155   6.98%  82     19     54     0      0      0      4      0.00%
malyuga:1     0.05%  0       0      1      0      0      0      0      0.00%
pere: 5      0.23%  0       0      0      0      5    20000   800025  14.85%
koverov:8     0.36%  5       3      0      0      0    3379   149732  2.78%
root: 32     1.44%  28      0      3      1      0      0     3532     0.07%
ru: 24      1.08%  17      3      4      0      0     148   159688  2.96%
solo: 123    5.54%  70     44     9      0      0     168   1362447 25.28%
soro: 175    7.88% 112      9     54     0      0     39   223762  4.15%
tikus: 27    1.22%  22      4      1      0      0      3    1164     0.02%
vgor: 72     3.24%  61      1     10     0      0     44   204214  3.79%
xand: 5      0.23%  4       0      1      0      0      3     83      0.00%
yakim: 596   26.82% 357    107   132     0      0      7    96320   1.79%
yrubin: 3    0.14%  3       0      0      0      0      1     34      0.00%
-----
TOTAL: JobCount, CD,CA,F,NF,T0, AvgJobTime(min), Load(TotalTime(min)*NCPU)
TOTAL: 2222      1427      386      389      1      19      106      5388739
```

Рисунок А.4 — Відображення статистики використання кластера

## ДОДАТОК Б АКТИ ВПРОВАДЖЕННЯ



