УДК (519.6+004.05):004.075

Моделі та характеристики обчислювального кластера, які допомагають визначати напрямки його подальшого розвитку

В. О. Міщенко

Харківській національний університет імені В. Н. Каразіна, Україна

Розглянуто кластерну систему з неоднаковими у загальному випадку багатопроцесорними вузлами, яка потребує з часом послідовного поповнення або реконструкції за умови економічної доцільності. Проаналізовано математичні моделі функціонування такої системи, і розроблено відповідні метрики якості. Обґрунтовуються гіпотези про корисність цих метрик для осіб, що приймають рішення стосовно поповнення кластерних систем з метою підвищення якості їх роботи. Зокрема розроблено метод пошуку оптимальних у певному сенсі рішень за допомогою обчислювального експерименту. Реальним прикладом для ілюстрації прогнозів послугував міні-кластер факультету комп'ютерних наук Харківського національного університету ім. В. Н. Каразіна.

Ключові слова: *обчислювальний кластер, метрики якості, математична модель, оптимізація, обчислювальний експеримент.*

Рассмотрена кластерная система с разными, как правило, многопроцессорными узлами, которая требует со временем последовательного пополнения или реконструкции при условии экономической целесообразности. Проанализированы математические модели функционирования такой системы, и разработаны соответствующие метрики качества. Обосновываются гипотезы о полезности этих метрик для лиц, принимающих решения о пополнении кластерных систем с целью повышения качества их работы. В частности, разработан метод поиска оптимальных в определенном смысле решений с помощью вычислительного эксперимента. Реальным примером для иллюстрации прогнозов послужил мини-кластер факультета компьютерных наук Харьковского национального университета им. В. Н. Каразина.

Ключевые слова: *вычислительный кластер, метрики качества, математическая модель, оптимизация, вычислительный эксперимент.*

We considered cluster system with different, usually multiprocessor nodes, which require of replenishment or reconstruction if feasible. The mathematical models of the functioning of such a system were considered, and appropriate quality metrics were developed. We have argued the hypothesis of the usefulness of these metrics for decision-makers on the replenishment of cluster systems in order to improve the quality of their work. In particular, the method of finding optimal solutions by the help of computer simulation was developed. A real world example to illustrate the predictions was shown. It was the predictions for the mini-cluster of Computer Science faculty of V. N. Karazin Kharkiv National University.

Key words: *computing cluster, quality metrics, mathematical model, optimization, numerical experiment.*

1. Вступ

Кластери різного складу та архітектури за останнє десятиліття стали звичним полігоном для виконання обчислювальних експериментів, які до донедавна були неможливі на персональних комп'ютерах в наслідок браку пам'яті та часу [1].

Поперває коштовність їхнього облаштування і складність налаштування були настільки високими, що здебільшого ставали можливими лише під спеціальні проекти [2]. Згодом, зокрема в наслідок появи у складі комп'ютерів багатопроцесорних ядер, доступних звичайним фахівцям і у навчанні, ситуація змінилась. При цьому є *актуальним* питання про нарощування існуючих міні-кластерів до потужних ферм, модернізація існуючих учбових кластерів і т.п. [3]. Об'єктивною основою вирішення таких питань повинні слугувати показники потенціальної якості роботи кластера, залежні лише від його компоновки та режиму функціонування. Таким чином виникає *постановка задачі* про розробку метрик кластеру на основі обраної математичної моделі його роботи. Вибір моделі обумовлює конкретний варіант постановки. При цьому можна спиратись на доволі широкий спектр існуючих математичних моделей процесу обчислень у кластері, з яких нашу увагу звернули близькі до академічних інтересів типи моделей, розглянутих у [4-9]. Дві різні моделі, одна, що відбиває класичні уявлення про паралельну обробку даних, та друга, що акцентує увагу на обробці потоків завдань, які надходять із зовні, послугували базою дослідження нашої роботи. Вони відповідають наступним, де в чому протилежним, точкам зору:

- робота кластера тим краща, чим якісніше відбувається обчислень користувачів, тобто більш рівномірно завантажуються всі процесори обчислювальної системи (хоч як мало чи багато завдань обслуговується);
- робота кластера тим краща, чим повніше забезпечується зовнішній запит на проходження через кластер доволі щільного потоку обчислювальних завдань користувачів (без відносно того, яка б не була якість розпаралелювання обчислень цих завдань, що є турботою самих користувачів).

2. Моделювання якості розподілених обчислень у кластері

Дотримуючись першої точки зору, доцільно надати відомій формулі прискорення обчислень системою функціональних пристроїв (ФП) [4] вигляд:

$$R=\frac{\sum\limits_{k=1}^{N}p_k\pi_k}{max\,\{\pi_k\}_k}=\left(\sum_{k=1}^{N}p_k\frac{\pi_k}{\sum\limits_{l}\pi_l}\right)\cdot\frac{\sum\limits_{k=1}^{N}\pi_k}{max\,\{\pi_k\}_k}=p\cdot\frac{\sum\limits_{k=1}^{N}\pi_k}{max\,\{\pi_k\}_k} \qquad (1)$$

де R- міра прискорення обчислення відносно відповідного простого пристрою;

p_k - завантаженість k-го з N пристоїв системи;

π_k - пікова продуктивність k-го з пристоїв системи;

p - так звана завантаженість системи (див. (2.2) у [4]).

Оскільки для системи з даними піковими продуктивностями ФП величина

$$\Pi=\frac{\sum\limits_{k=1}^{N}\pi_k}{N\,max\,\{\pi_k\}_k}\ ,\quad \left(0<\Pi\le 1\right)\quad , \qquad (2)$$

відіграє роль *середньої відносної продуктивністі* (СВП), то з (1) отримуємо

$$R=N\cdot p\cdot\Pi \qquad (3)$$

де N – число ФП, p - завантаженість системи при обчисленні [4], а – СВП (2).

Ми можемо розглядати прискорення обчислень математичного алгоритму за рахунок розподілення обчислень (1), як головну якість кластерного обчислення. Тоді (3) дозволяє трактувати цю якість, як результат рівної дії одного фактору, залежного від якості розподілення обчислень по процесорних ядрах, і двох факторів, залежних тільки від кластеру – загальної кількості ядер і СВП, яка відбиває неоднорідність системи процесорів.

Нехай роль ФП відіграють процесорні вузли кластеру. Тоді у типовій для невеликих навчальних підрозділів ситуації (факультети, кафедри, лабораторії), коли кластерна система складалась поступово, залученням до неї комп'ютерів різних поколінь, СВП стає важливою метрикою кластеру. Вона потребує розрахунку, в різних варіантах поповнення кластеру приймає різні значення і може грати нетривіальну роль при виробці рішень щодо поповнення.

Поряд з режимами, для яких характерним є розпаралелювання програм лише у межах одного вузла, кластер може працювати в режимах, коли суттєвим є розподіл обчислень однієї програми між вузлами. Тоді роль ФП відіграватимуть вузли. Якщо тактові частоти процесорних ядер можна вважати хоча б приблизно однаковими, то СВП практично визначається розподілом вузлів за кількістю їхніх ядер, тобто формується дуже наглядно для осіб, що приймають рішення.

Перевіримо якісну адекватність метрики (2) на практичному прикладі. З міркувань здорового глузду, доти, доки ми додаємо новий ФП, продуктивность якого π_{N+1} не є малою та не перевищує попередніх, додавання збільшує якість кластеру. Але, якщо ми додамо ФП із набагато більшим значенням пікової продуктивності, ніж у попередніх ФП, то за принципом «не варто на старе ліпити заплат із нової тканини», якість варто вважати погіршеною. Формально

$$\pi_{N+1} \le \pi_{max} \quad \Rightarrow \quad \Pi_{new} = \Pi + (\theta - \Pi)/(N+1) \quad ,$$
$$\pi_{N+1} > \pi_{max} \quad \Rightarrow \quad \Pi_{new} = \Pi - \Pi \cdot \theta^{-1}/(N+1) - \left(\Pi \cdot (1-\theta^{-1}) - (N+1)^{-1}\right) \quad (4)$$

де $\theta = \pi_{N+1}/\pi_{max}$ -відношення пікових продуктивностей.

Звідси випливає твердження, цілком узгоджене з наведеними міркуваннями:

Лема 1. Якщо відношення пікової продуктивності доданого вузла до максимуму пікової продуктивності попередніх вузлів не більше від 1 та достатньо велике, то числове значення СВП зросте. Та якщо це відношення більше за 1 і має достатньо велике значення, то значення СВП зменшиться.

Доведення першого твердження очевидне. Для доведення другого достатньо додатково зауважити, що $\Pi > N^{-1}$. Тоді для останньої складової формули (4):

$$\Pi \cdot (1-\theta^{-1}) - (N+1)^{-1} > \theta^{-1} \cdot N^{-1} \cdot (1-\theta^{-1}) - (N+1)^{-1} > 0 \text{, якщо}$$
$$\pi_{N+1} > (N+1)\pi_{max} \quad .$$

На відміну від якісних міркувань, СВП дозволяє знайти точний рубіж для відношення продуктивностей θ, за яким додавання вузла погіршує якість!

Формула (3) вказує також на доцільність метрики режиму обчислень, основаної на завантаженості системи. Зазначимо, що безпосередньо p для цього непридатна, оскільки її визначено як миттєву величину [4], байдуже, що із словами «в середньому» (за припущеннями моделі питання взаємодії ФП ігноруються). Доповнимо класичну модель системи ФП припущенням про те, що для запобігання мертвих блокувань і перманентного гальмування окремих обчислень (їх «підвисання») завдання на обчислення у кластері об'єднуються в пакети. Байдуже зараз, чи об'єднує пакет програми різних користувачів, чи паралельні процеси однієї програми. Важливо лише, щоб завдання у межах пакету у кожний момент із значною ймовірністю потребували різних ресурсів та якомога повніше їх використовували! Кілька пакетів можуть виконуватись у кластері водночас, та це посилює ризик їх небажаної взаємодії.

Нехай на відрізку часу від T до $T+\Delta$ виконувалось S пакетів завдань, причому для кожного, s–го пакета ($s = 1 .. S$) у моменти часу τ_k вимірювалась завантаженість кластера p. Позначимо початок виконання s–го пакету як t_0^s, а кінець, як t_n. Надалі вважатимемо, що

$$t_0^s < .. < t_n^s \quad \left(n = n\,(s)\right), \tag{5}$$

$$\tau_k^s = \frac{t_k^s + t_{k+1}^s}{2} \quad \left(k = 1 .. n-1\right), \quad p_k^s = p(\tau_k^s) \tag{6}$$

і введемо *середню завантаженість кластера* (СЗК) у даний період як величину

$$p_{cl}\,\left(T, T+\Delta\right) = \left(\sum_s \left(t_{n\,(s)}^s - t_0^s\right)\right)^{-1} \cdot \sum_s \sum_{i=1}^{n\,(s)} p_i^s \cdot \left(t_i^s - t_{i-1}^s\right) \tag{7}$$

причому перекриття процесів виконання пакетів, може призвести до того, що,

$$\sum_s \left(t_{n\,(s)}^s - t_0^s\right) > \Delta \quad . \tag{8}$$

Демонстрацію адекватності обмежимо простим, але вважливим прикладом автоматичного врахування ризиків. Розглянемо 2 варіанти завантаження кластера протягом часу Δ=3 двома пакетами. Нехай

$$t_0^s = 0.0\,, \quad t_1^s = 0.5\,, \quad t_2^s = 1.5\,, \quad t_3^s = 2.5\,, \quad t_3^s = 3.0 \quad (s{=}1{,}2) \tag{9}$$

Приклад 1. У першому варіанті виконання кластер у перші 1.5 одиниць часу завантажено на 1.0 першим пакетом, а у другі 1.5 одиниць – тільки другим. У другому варіанті перший та другий пакети монопольно і повністю завантажують кластер у першу та відповідно – у останню одиницю часу, а у середню – кожний завантажує на 0.5 . В обох варіантах кластер увесь час повністю завантажено, але перший варіант кращий за другий, оскільки не несе ризику неконтрольованої взаємодії пакетів. Розрахунок СЗК узгоджується з цим міркуванням :

$$p_{cl}^1\left(0.0, 3.0\right) = \;\; 1.0 \;\; > \;\; p_{cl}^2\,\left(0.0, 3.0\right) = \;\; 0.75\,. \tag{10}$$

Залишається заперечення, що мовляв, оскільки міра СЗК означена як апостеріорна, а виконання пакетів успішно відбулось, то різниці в результатах оцінки СЗК для обох варіантів не повинно бути. Та ця аргументація має логічну прогалину. А звідки відомо, що «успішно»? Очевидно, що на інтервалі часу від 1.5 до 2.5 кожний з них потребував би на своє виконання у режимі монопольного завантаження лише по 0.5 одиниць часу, а у наслідок конкуренції за ресурси фактичне виконання тих самих обчислень затягнулось на 1.5 часу!

3. Якість GRID орієнтованого обслуговування

Тепер звернімось до моделі багатоканальної системи масового обслуговування (СМО) з обмеженою чергою [9], у яку в режимі випадкового процесу надходять заявки на виконання обчислювальних завдань. До таких моделі нашу увагу привернули робота [6], в якій за її допомогою вирішено іншу задачу. Зауважимо, що так само, як у попередньому розділі можливість оцінювання та змістовної інтерпретації показників якості кластера на основі формальної моделі, передбачає певний режим роботи і потребує ідеалізуючих припущень. По перше, поняття «завдання» матиме сенс програми (або її частини, або комплексу програм), що виконується на одному вузлі, причому монопольно, протягом випадкового часу з певним математичним очікуванням. Тобто програми, які вірогідно виконуються швидко, не потребуючи великого об’єму оперативної пам’яті, треба об’єднувати в одне завдання, а програми, що потребують розподілення по кількох вузлах, і виконуються довго, треба представляти частинами, що розглядатимуться як окремі завдання. Далі зважимо на те, що прості, зручні для математичного аналізу моделі СМО потребують припущення про статистичну незалежність термінів виконання завдань, а також наявності експоненційного закону розподілу для термінів надходження заявок і для термінів їх виконання. Послаблення цих вимог можливе [5]. Однак, ми приймемо для моделі СМО кластера припущення про марківський характер [9].

Нехай на вхід СМО (тобто кластера), котрий має m каналів обслуговування (тобто вузлів), надходить пуасонівський потік заявок з інтенсивністю λ, інтенсивність обслуговування заявок кожним каналом, дорівнює μ, а максимальне число місць очікування у черзі це n (всі заявки, котрі надходить у СМО при черзі з n заявок, не приймаються і надалі ніколи не обслуговується).

Параметри навантаження кластера та його довільного вузла, обчислюватимемо як відповідні навантаження на СМО за формулами [9]:

$$\rho = \frac{\lambda}{\mu}\ , \qquad \rho_m = \frac{\lambda}{m \cdot \mu} \tag{11}$$

Ми використовуватимемо наступні відомі формули для розрахунку параметрів усталеного режиму функціонування СМО типу M|M|m|n (математично доведено, що в такій моделі цей режим існує [9]).

Ймовірність простою системи P_0 – складає

$$P_0 = \left[\sum_{i=0}^{m} \frac{(\rho)^i}{i!} + \frac{(\rho)^{m+1}}{m!} \cdot \frac{1-\rho^n}{1-\rho} \right]^{-1} . \tag{12}$$

Тоді ймовірність P_+ обслуговування є

$$P_+ = 1 - \frac{1}{m^n \cdot m!} \cdot P_0 , \tag{13}$$

а пропускна здатність системи γ складає

$$\gamma = \lambda \cdot P_+ \quad . \tag{14}$$

Середній час очікування у черзі W дорівнює

$$W = \frac{m\pi_m}{\mu(m-\rho)^2} \cdot \left(1-(n+1)(\rho_m)^n + n(\rho_m)^{n+1}\right). \tag{15}$$

Введемо міри для характеристики кластера з зовнішнього боку і з середини.

Дефект допустимого простою (ДДП) вказує на відхилення стаціонарного режиму експлуатації кластера в складі грід від тієї вимоги, що середній час очікування завдань у черзі на виконання повинен складати задану долю («відсоток») від середнього часу виконання завдань:

$$\Delta = \vartheta/\mu - W , \tag{16}$$

де Δ - числове значення метрики ДДП;

ϑ – задана доля очікування у черзі від середнього часу виконання завдань;

μ та W – вище означена інтенсивність обслуговування заявок вузлами та середній час очікування (15).

Відносна продуктивність розширення (ВПР) показує наскільки виросте пропускна здатність стаціонарного режиму при поповненні кластеру на 1 вузол при плановій оптимізації черг:

$$E = (\gamma - \gamma_0)/\gamma_0 , \tag{17}$$

де E - числове значення метрики ВПР;

γ – пропускна здатність кластера після його розширення та оптимізації довжини черги;

γ_0 – поточна пропускна здатність кластера (наявна або очікувана за планом).

Для пошуку узгодженого оптимального розширення кластера, що відповідає простій марківській СМО, створено Ада програму Markov_Claster. Запустивши її, вводимо очікувані інтенсивності завдань та обслуговування одним вузлом, існуючу кількість вузлів, апріорно задану мінімальну довжину черги очікування завдань, максимум можливої кількості вузлів, максимально можливу довжину черги та рекомендовану границю часу простою у черзі відносно середнього терміну виконання. На виході буде кількість вузлів, на якій досягається максимум приросту ВПР, орієнтовна границя часу простою, відповідна оптимальна відносно ДДТ довжина черги та навантаження на кластер (очікуване у разі визначеного розширення).

4. Апробація СВП як засобу підготовки прийняття рішень

Проаналізуємо варіанти поповнення мини-кластеру ФКН ХНУ ім. В.Н. Каразіна за станом на 2010 р. [3]. Він складався з двох груп вузлів (група 3 ядерного комп’ютера і група двох 4 ядерних). Для планування можливого поповнення міні-кластеру на основі середньої відносної продуктивності (2) було намічено 4 варіанти придбання комп’ютерів приблизно з однаковою сумарною пам’яттю. Варіанти вважались приблизно економічно рівноцінними.

Варіанти поповнення визначались кількістю однорідних груп вузлів, кількістю ядер у вузлах по групах та кількісним складом груп (табл. 3.1).

Таблиця 3.1. Оцінка варіантів, що відрізняються за групами вузлів

№	Кількість груп	Ядерність вузлів	Кількість у групах	СВП
1	3	2, 3, 4	4, 1, 1	0.68
2	2	3, 4	4, 2	0.83
3	2	3, 4	1, 4	0.95
4	3	3, 4, 8	1, 2, 1	0.59

Висновок полягав у тому, що нарощування існуючої РС-ферми із збереженням існуючих вузлів забезпечить вищу якість у розумінні СВП, якщо всі додані вузли будуть 4 ядерними. Звертає увагу те, що найбільш привабливий, з огляду на новизну, варіант придбання 8 ядерного комп’ютеру отримав найгірший прогноз щодо впливу на якість кластеру. Це демонструє обмеженість СВП рамками зроблених припущень. Наприклад, цей варіант може здатися кращим з точки зору доступу більшості процесорів до пам’яті у ході обчислень.

На практиці у академічних кластерах водночас виконується кілька завдань і, навіть, кілька пакетів завдань. Зокрема так буває, якщо це учбові завдання, або якщо вони надходять з грід. Ми продовжимо аналіз наприкінці наступного розділу, використавши відповідні міри ефективності роботи кластера.

5. Прогноз ефекту нарощування РС-ферми, що працює у режимі СМО

Є сенс віртуально нарощувати кластер, додаючи черговий вузол, оптимально плануючи за мінімумом ДДТ нову максимальну довжину черги завдань, розраховуючи після цього ВПР та знов додаючи вузол. При цьому значення метрики ВПР з кожним доданням чергового вузла змінюватиметься, причому спочатку ці значення зростатимуть, а надалі зменшуватимуться. Звісно, що кількість вузлів, яка відповідає максимумові, об’єктивно є границею поповнення. Цю границю знаходить і рекомендує за найякісніший варіант поповнення програма Markov_Claster, вхідний інтерфейс якої ми вище описали.

Приклад 2 (суто демонстраційний). Нехай інтенсивності потоків завдань та термінів виконання однакові (1.0 та 1.0), існують 2 вузли, встановлено довжину черги 1. Запустимо Markov_Claster і введемо ці та подальші дані відповідно до інтерфейсу програми, описаному у розділі 3. З рис. 1 бачимо, що оптимальна кількість вузлів 3 (якщо більше, то приріст ВПР в результаті додання вузла різко зменшується), оптимальна довжина черги була б 6. В такому разі буде отримано приріст пропускної здатності від 0.917 практично до 1.0 (і нема чого далі поповнювати кластер при збереженні прийнятих умов).

Приклад 3 (демонстраційний). Змінивши умови, отримаємо оптимум, що відповідає поповненню на 2 вузли. Результати і вихідний інтерфейс - на рис. 2.

```
 ***  Markov_Claster  ***
Put Lambda (jobs): 1.0
Put   Mu  (terms): 1.0
Put   M0  (nodes): 2
Put   N0  (queue): 1
Put   M2  (.. M2): 6
Put   N2  (.. N2): 12
Put max of W/mu**(-1): 0.1
It's recomended queue length to be 1 -- dW =  1.667E-02
     * Max_Prodactivity *
m= 2 => 9.16667E-01
    M    Wmin   N  (G-G0)/G0
    2  8.33E-02  1 0.00000E+00
    3  4.17E-02  6 9.08282E-02
    4  4.24E-02  6 7.06963E-05
    5  5.97E-02  6 3.27827E-06
    6  7.10E-02  6 1.78814E-07
    *end Max_Prodactivity*
It's recomended nodes number to be 3 -- dG/G =  9.083E-02
Then the prodactivity will be  9.999E-01
 ***End of Markov_Claster***
```

Рис. 1. Результати автоматичного пошуку програмою Markov_Claster оптимальної кількості вузлів для поповнення кластера

```
 ***  Markov_Claster  ***
Put Lambda (jobs): 0.1
Put   Mu  (terms): 0.01
Put   M0  (nodes): 5
Put   N0  (queue): 1
Put   M2  (.. M2): 8
Put   N2  (.. N2): 12
Put max of W/mu**(-1): 0.05
It's recomended queue length to be 2 -- dW = -8.560E+01
     * Max_Prodactivity *
m= 5 => 9.99998E-02
    M    Wmin   N  (G-G0)/G0
    5  4.43E+00  2 0.00000E+00
    6  2.22E+00  1 5.21542E-07
    7  2.70E+00  1 1.11759E-06
    8  2.79E+00  1 1.49012E-07
    *end Max_Prodactivity*
It's recomended nodes number to be 7 -- dG/G =  1.118E-06
Then the prodactivity will be  1.000E-01
 ***End of Markov Claster***
```

Рис. 2. Приклад оптимального поповнення кластеру з 5 до 7 вузлів. При цьому його максимальна пропускна здатність збільшується у 10 разів.

Приклад 3 (практичний). Тепер розглянемо перспективи розширення міні-кластера ФКН за умови підключення до грід за різних варіантів вимог. Врахуємо, що наш мини кластер має невелику потужність, і розберемо приклад, коли навантаження на нього складає 0.2. Показові результати вміщено у табл.2. Бачимо, що потреба у максимальному поповненні проявляє себе тільки тоді, коли існуючий кластер працює з оптимальною чергою. При порівняно невеликій варіації даних можна отримати рекомендацію і на мінімальне поповнення (до 4 вузлів), і на значніше (до 5), і на максимальне (у нас 6). З цього виводимо, що для великих кластерів оптимум поповнення треба шукати засобами багатовимірної оптимізації (у 3-D просторі, враховуючи зміну навантаження).

Таблиця 2. Розширення до границі максимізації приросту ВПР за різних умов взаємодії з грід.

№ варіанту	Наванта-ження ρ	Спершу вузлів m_0	Довж. черги n	Очікування $W(15)$	Треба вузлів m
1	0.2	3	0	0.2	4
2	0.2	3	0	0.25	5
3	0.2	3	1	0.25	6
4	0.2	3	2	0.2	6

Зіставлення табл. 2 з результатом розділу 4, показує, що нарощування розглянутого мини-кластеру ФКН від 3 до 6 вузлів, додаючи однакові 4-ядерні, забезпечить при кожному нарощуванні максимальний приріст метрик якості СЗК і ВПР. Цей план був обраний за робочий, одначе відповідно до спеціалізації кластеру на експериментах з математичного моделювання складних фізичних процесів і обробці даних наукових експериментів при модернізації кластеру було обрано напрямок оснащення існуючих комп'ютерів картками CUDA [10]. З великою натяжкою це можна розглядати як додавання у вузли ядер великої пікової продуктивності або як певне підвищення продуктивності вузлів. Якщо можна, то неважко упевнитись, що цей варіант також призводить до суттєвого приросту метрики якості ВПР.

5. Висновки

Строго поставлено і розв'язано для різних режимів експлуатації задачу про підвищення якості неоднорідного обчислювального кластера при його поповненні. Ключові результати стосуються визначення і обґрунтування на основі моделей та прикладів наступних метрик якості кластерної системи: середня відносна продуктивність (СВП), середня завантаженість кластера (СЗК) у заданий період часу, дефект допустимого простою (ДДП), відносна продуктивность розширення (ВПР). Метод використання передбачає автоматичний аналіз варіантів відносно ВПР. Застосування розглянуто на прикладі міні-кластера факультету комп'ютерних наук Харківського національного університету ім. В. Н. Каразіна.

Важливим напрямком розвитку моделей роботи обчислювальних кластерів і створення відповідних метрик якості, є врахування у складі комп'ютерів спеціальних, насамперед графічних, процесорів.

Інший напрямок розвитку результатів роботи полягае у створенні інформаційних технологій підготовки рішень щодо способів вдосконалення кластер них систем на основі системи метрик.

ЛИТЕРАТУРА

1. Воеводин Вл.В., Жуматий С.А. "Вычислительное дело и кластерные системы".- М.: Изд-во МГУ, 2007. - 150 с..
2. Voronko M. V. LCG Middleware at the KIPT CMS Linux Cluster / M. V. Voronko, S. S. Zub, L. G. Levchuk, D. V. Soroka // Bulletin of V. Karazin Kharkiv National University. – 2005. – № 703. – P. 74-86. – (Series: Mathematical Modelling. Information Technology. Automated Control Systems; no. 5).
3. Mishchenko V.O. Towards the questions on planning the development of the Department computer cluster / Victor O. Mishchenko, Oleksandr U. Baiev, Ievgen V. Didenko, Valentine T. Lazurik // CONFERENCE PROCEEDINGS ICT in Education, Research, and Industrial Applications: Integration, Harmonization, and Knowledge Transfer (4-8 may 2011)», - Kherson, 2011, P. 27.
4. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления / БХВ-Петербург, 2002. – С-Пб. – 608 с.
5. Морозов Е. В., Румянцев А. С. Вероятностные модели многопроцессорных систем: стационарность и моментные свойства // Информатика и ее применения. 2012. Т. 6, № 3. С. 99–106.
6. Куланов С. А. Применение математического аппарата теории систем массового обслуживания для оценки стоимостных показателей grid-систем / С. А. Куланов, В. С Харченко. // Вісник Харківського національного університету: Зб. наук. праць. – Х., – 2007. – № 780. – С. 143-150. – (Серія: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління; вип. 8).
7. Куланов С.А. Теоретико-множественная модель потока задач в grid-системах / С. А. Куланов // Радіоелектронні і комп'ютерні системи. – 2008. – № 5. – С. 32-36.
8. Куланов С.А. Метод планирования задач и распределения ресурсов в грид на основе процедур прогнозирования / С. А. Куланов // Радіоелектронні і комп'ютерні системи. – 2010. – № 6. – С. 68-72.
9. Дудин А. Н. Практикум на ЭВМ по теории массового обслуживания / А. Н. Дудин, Г. А. Медведев, Ю. В. Меленец [Электронный ресур]: Учебное пособие. – Мн.: “Электронная книга БГУ”, 2003. – Режим доступа: http://anubis.bsu.by/publications/elresources/AppliedMathematics/dudin.pdf . — Электрон. версия печ. публикации, 2000. – PDF формат, версия 1.4.
10. Боресков А. В., Харламов А. А. Основы работы с технологией CUDA. М.: ДМК Пресс, 2010. 232 с.

Надійшла у першій редакції 22.04.2013, в останній - 01.05.2013