

Павлов В.Г.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

ДИСПЕТЧЕРИЗАЦІЯ ДІНАМІЧНОГО БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ОБЧИСЛЮВАЛЬНИХ ПРИСТРОЇВ НА ОСНОВІ ОЦІНЮВАННЯ СТАНУ ПОТОЧНИХ ОБЧИСЛЕНЬ

У статті проаналізовані методи статичного та динамічного балансування навантаження обчислювальних пристроїв в умовах, коли ці пристрої відрізняються за своєю обчислювальною потужністю. Запропоновано спосіб динамічного балансування на основі універсального критерію, а саме – мінімального часу виконання усіх завдань. В якості показника завантаженості обчислювального пристрою використовується його часові характеристики. Покрокова послідовність динамічного балансування розглянута на прикладі. Визначені перспективи подальшого вдосконалення.

Ключові слова: розподілені обчислення, балансування навантаження, динамічне балансування, оцінювання навантаження, критерії балансування, диспетчер навантаження.

Постановка проблеми. Розробка систем штучного інтелекту та пов'язана з цим необхідність обробки величезних обсягів даних вимагає різкого підвищення ефективності обчислень. Одна з найбільш відомих міжнародних дослідницьких компаній **International Data Corporation** ще у 2011 році надала результати своїх досліджень, згідно з якими обсяг обчислень зростає експонентно та подвоюється кожні 3,5 місяці, тобто у п'ять разів швидше, ніж вважалося раніше за відомим законом Мура [1]. Наступні роки лише підтвердили цей прогноз.

Звичайно, підвищити ефективність обчислень можна шляхом застосування більш потужних комп'ютерів, але цей шлях інтенсивного технологічного розвитку має свої межі. Тому вже давно вихід знайдений у поєднанні обчислювальних потужностей окремих комп'ютерів, між якими розподіляються або окремі частки однієї задачі, або взагалі окремі задачі. При цьому вкрай важливою стає проблема оптимального розподілу задач (або окремих частин однієї задачі) між усіма обчислювальними пристроями, що виконують ці розподілені обчислення, тобто балансування навантаження.

Аналіз останніх досліджень і публікацій. Оскільки проблема балансування навантаження обчислювальних пристроїв існує досить давно, то шляхи та методи її вирішення обговорюються у численних публікаціях на цю тему. Класифікацію та загальний огляд методів балансування здійснено у роботах V. Thakur, S. Kumar [2], Debankur Mukherjee, Sem Borst, Johan van Leeuwen, Philip Whiting [3], B. Hendrickson, K. Devine [4],

Y.F. Hu, R.J. Blake, D.R. Emerson [5], Бо Тянь, М.А. Посыпкина, И.Х. Сигала [6], Д. В. Белькова [7], А. Емельянова [8] та А.С. Хританкова [9]. Статті Hong, J.H. No, S.Y. Kim [10], Sarhan Ebada, Ghalwash Atif, Mohamed Khafagy [11], Saeed Sharifana, Seyed A. Motamedia, Mohammad K. Akbari [12], Zhang Lin, Li Xiao-ping [13] та Neeraj Rathore [14] присвячено балансуванню навантаження WEB-серверів. У публікації Е.Г. Игнатенко [15] більш детально розглянуто балансування у кластерних системах, а В.А. Райхлин, та Р.К. Классен [16] дослідили балансування у розподілених базах даних. Однак у наведених роботах під час балансування обчислювальних пристроїв майже не приділяється уваги дослідженню часових характеристик та балансуванню на їх основі.

Постановка завдання. Як відомо, балансування поділяється на статичне та динамічне. Перший вид балансування виконується перед початком обчислень та дозволяє заздалегідь оптимально розподілити навантаження між обчислювальними пристроями за заданими критеріями. Але ця перевага зводиться нанівець, коли нові завдання виникають вже під час обчислень, та на них треба оперативно реагувати. В цьому випадку використовуються методи динамічного балансування, яке в більшості методів зводиться до перерозподілу завдань між обчислювальними пристроями (ОП) з урахуванням оновлених даних. Цей перерозподіл виконується або на основі обміну інформацією між самим обчислювальними пристроями, тобто децентралізовано, або під управлінням диспетчера навантаження, який виконує роль єдиного центра керування. Найчастіше диспетчер наван-

таження здійснює пошук оптимального варіанта балансу навантаження обчислювальних пристроїв згідно з заданими критеріями в умовах стислого часу, оскільки за появи нових завдань знайдений варіант балансування перестає бути актуальним. В якості подібних критеріїв виступають наступні [7, с. 72]:

1. максимально рівномірний розподіл завдань за усіма ОП;
2. максимальна продуктивність системи для усіх додатків;
3. мінімальний час відповіді системи на кожний запит;
4. максимально швидке виконання завдань згідно з розкладом.

Оскільки безпосередньо або опосередковано у кожному з цих критеріїв йдеться про часові характеристики, то пропонується їх звести до одного універсального: **мінімальний час виконання усіх завдань на наявних ОП**. Цей критерій найбільшою мірою відповідає головній цілі використання розподілених обчислень – досягнення максимальної їх продуктивності.

Другим фактором, який необхідно враховувати під час балансування навантаження ОП, є їх обчислювальна потужність. У більшості методів балансування вона або не враховується, або вважається однаковою у всіх ОП. Це, звичайно, створює суттєві обмеження на застосування цих методів в умовах, коли виконується балансування між ОП, які мають різну швидкість обчислень, хоча й забезпечують функціональну однорідність [9, с. 68].

Введемо також третє обмеження: завдання обробляються диспетчером у тій же послідовності, в якій вони поступають, тобто підтримується дисципліна FIFO, що не потребує зміни порядку обробки завдань та виконання будь-якого їх попереднього упорядкування. Також ці завдання повинні бути незалежними одне від одного, а саме результати виконання одного з них не є входними даними для будь-якого наступного в межах часового інтервалу балансування. Але для реалізації балансування завдань висувається необхідна умова: для кожного завдання повинна бути відома метрика, яка характеризує його обчислювальну складність. Ця або схожі метрики використовуються авторами під час розгляду статичних методів балансування, наприклад [7, с. 73] та [6, с. 13–14], але не знайдено прикладів її застосування у методах динамічного балансування.

Виклад основного матеріалу дослідження. Розглянемо спосіб балансування послідовності з M завдань, які створюють певну послідовність, та

кожне з яких має власну відмінну обчислювальну складність Z_i , де $i = 1, 2 \dots M$. Балансування здійснюється у системі, яка складається з N обчислювальних пристроїв, які в загальному випадку мають різні обчислювальні потужності: P_j , де $j = 1, 2 \dots N$. Тоді час виконання i -того завдання на j -том обчислювальному пристрої T_{ij} визначається як:

$$T_{ij} = \frac{Z_i}{P_j}$$

Звичайно, вважається, що $M \gg N$, інакше балансування навантаження ОП зводиться до тривіальної статичної задачі.

Введемо поняття «поточний час роботи ОП τ_j », значення якого буде відбивати час, який минув з початку роботи ОП j . На першому етапі, коли ОП вільні та знаходяться в режимі очікування, $\tau_1 = \tau_2 = \dots \tau_j = \dots \tau_N = 0$.

Згідно з третім обмеженням, яке було введено раніше, спочатку розподіляється завдання Z_1 , потім Z_2 , ... Z_N . Критерієм розподілу пропонується вважати найменший час роботи τ_j , який буде мати j -те ОП порівняно з усіма іншими після виконання завдання Z_1 . Оскільки до розподілу завдання Z_1 усі ОП знаходилися у режимі очікування ($\tau_1 = \tau_2 = \dots \tau_j = \dots \tau_N = 0$), то найменший час роботи τ_j забезпечить те j -те ОП, яке швидше за інших зможе виконати завдання Z_1 , тобто

$$\min \{T_{1j}\} \text{ для } j \in 1, 2, \dots, N$$

В умовах, коли ОП мають різну обчислювальну потужність, звичайно за найменший час T_{1j} завдання Z_1 виконає ОП з найбільшою обчислювальною потужністю. Нехай це ОП має номер k , де $k \in 1, 2, \dots, N$. Тому це завдання буде розподілено диспетчером саме йому, а час τ_k буде збільшений на час T_{1k} та буде становити $\tau_k = T_{1k}$. Під час розподілу диспетчером завдання Z_2 критерієм його призначення для виконання якомусь ОП знову ж таки буде найменше значення τ_j , але вже за інших початкових умов:

$$\tau_1 = \tau_2 = \dots \tau_j = \dots \tau_N = 0 \text{ для } j \in 1, 2, \dots, N, \text{ але } j \neq k \text{ та } \tau_k = T_{1k}$$

Тобто диспетчер знову повинен «приміряти» нове завдання до кожного ОП, але ті ОП, які вже виконують попередні завдання, будуть мати інші початкові значення поточного часу роботи.

До речі, якщо деякий ОП має суттєво більшу обчислювальну потужність порівняно з усіма іншими, то за цією умовою завдання Z_2 може бути розподілено диспетчером також йому, оскільки це ОП виконає два завдання (Z_1 та Z_2) за менший час, ніж будь-яке інше ОП тільки завдання Z_2 :

$$\tau_k = T_{1k} + T_{2k} < \tau_j \text{ для } j \in 1, 2, \dots, N, \text{ але } j \neq k$$

Доведемо, що послідовне застосування для кожного з M завдань подібного способу розподілу

навантаження серед N обчислюваних пристроїв забезпечить **мінімальний час виконання усіх завдань на наявних ОП** – Θ^M .

Нехай на кроці i розподіляється завдання Z_i . При цьому попередні $i-1$ завдання вже розподілені за критерієм мінімального часу їх виконання, який визначається наступним чином:

$$\Theta^{i-1} = \max \{\tau_j\} \text{ для } j \in 1, 2, \dots, N$$

Нехай k – номер ОП, яке буде працювати найбільш тривалий час, та, відповідно, буде визначати Θ^{i-1} , що буде дорівнювати τ_k .

За підсумками розподілу завдання Z_i , можливо усього три варіанти:

1. Завдання Z_i буде розподілено якомусь іншому s -тому ОП ($s \neq k$), **поточний час роботи** τ_s , якого навіть з урахуванням T_{is} не перевищить Θ^{i-1} . В цьому випадку значення мінімальний час виконання усіх i завдань не зміниться, тобто $\Theta^i = \Theta^{i-1}$.

2. Завдання Z_i буде розподілено також якомусь іншому s -тому ОП ($s \neq k$), але **поточний час роботи** τ_s , якого з урахуванням T_{is} перевищує Θ^{i-1} , тобто нове значення $\Theta^i > \Theta^{i-1}$. Оскільки Θ^i тепер буде визначати саме τ_s , а на попередньому кроці найбільше значення мав τ_k , то буде мати місце нерівність:

$$\tau_s - T_{is} < \tau_k < \tau_s$$

Тобто у результаті розподілу завдання Z_i відбулося збільшення **мінімального часу виконання i завдань** Θ^i , але це збільшення найменше з усіх можливих та не перевищує T_{is} .

3. Завдання Z_i буде розподілено також k -тому ОП, оскільки збільшення мінімального часу виконання буде в цьому випадку найменше порівняно з тим, якщо б завдання Z_i було б розподілено до якогось іншого ОП. В цьому випадку мінімальний час виконання збільшиться саме на величину T_{ik} :

$$\Theta^i = \Theta^{i-1} + T_{ik}$$

Таким чином, за підсумками розгляду усіх трьох варіантів подій доведено, що розподіл завдання Z_i призведе до мінімально можливого збільшення **мінімальний час виконання**, яке буде знаходитися у межах від 0 до T_{ik} , де k – номер ОП, яке визначало **мінімальний час виконання** на попередньому кроці.

Таким чином, послідовне застосування подібного способу розподілу для кожного з M завдань забезпечує найменший час їх виконання за допомогою N обчислювальних пристроїв, що й відповідає поставленим умовам.

Для ілюстрації запропонованого способу балансування навантаження розглянемо приклад.

Приклад.

Нехай виконується балансування 8 завдань, кожне з яких має свою відмінну складність:

$$\begin{array}{lll} Z_1 = 5; & Z_2 = 3; & Z_3 = 2; Z_4 = 1; \\ Z_5 = 7; & Z_6 = 4; & Z_7 = 8; Z_8 = 9. \end{array}$$

Балансування відбувається для 4 обчислювальних пристроїв, обчислювальна потужність яких визначається наступним чином:

$$P_1 = 2; P_2 = 1; P_3 = 1; P_4 = 3.$$

Зазначимо, що цифри у прикладах мають умовні значення, які допомагають зрозуміти хід балансування та легко перевірити результати обчислень. Але навіть через умовність самих значень ці цифри добре ілюструють ситуацію, коли i завдання, і обчислювальні пристрої мають досить різні характеристики, що відповідає постановці завдання.

Згідно зі способом балансування, що пропонується, **поточний час** роботи для кожного ОП має нульові значення, тобто $\tau_1 = \tau_2 = \tau_3 = \tau_4 = 0$

Крок 1 (розподіл завдання Z_1)

Спочатку диспетчер визначає час виконання завдання Z_1 для кожного з чотирьох ОП:

$$\begin{array}{lll} T_{11} = \frac{Z_1}{P_1} = \frac{5}{2} & T_{12} = \frac{Z_1}{P_2} = \frac{5}{1} & T_{13} = \frac{Z_1}{P_3} = \frac{5}{1} \\ & T_{14} = \frac{Z_1}{P_4} = \frac{5}{3} \end{array}$$

Оскільки T_{14} має найменше значення, то диспетчер направляє Z_1 на ОП 4.

Крок 2 (розподіл завдання Z_2)

Оскільки на попередньому кроці Z_1 виконувалася на ОП 4, то його поточний час роботи $\tau_4 = 5/3$, поточний час роботи для інших ОП мають знову ж таки нульові значення: $\tau_1 = \tau_2 = \tau_3 = 0$. Знову визначаємо час виконання завдання Z_2 для кожного з чотирьох ОП:

$$\begin{array}{lll} T_{21} = \frac{Z_2}{P_1} = \frac{3}{2} & T_{22} = \frac{Z_2}{P_2} = \frac{3}{1} & T_{23} = \frac{Z_2}{P_3} = \frac{3}{1} \\ & T_{24} = \frac{Z_2}{P_4} = \frac{3}{3} \end{array}$$

Але тепер шукаємо найменше значення серед значень **поточного часу роботи** для кожного з ОП з урахуванням їх початкових значень:

$$\begin{array}{ll} \tau_1^{(2)} = 0 + T_{12} = \frac{3}{2} & \tau_2^{(2)} = 0 + T_{22} = \frac{3}{1} \\ \tau_3^{(2)} = 0 + T_{23} = \frac{3}{1} & \tau_4^{(2)} = \frac{5}{3} + T_{12} = \frac{8}{3} \end{array}$$

Оскільки найменше значення має τ_1 , то диспетчер направляє Z_2 на ОП 1.

Крок 3 (розподіл завдання Z_3)

Маємо наступні значення показників **поточного часу роботи**:

$$\tau_1 = 3/2; \quad \tau_2 = 0; \quad \tau_3 = 0 \quad \tau_4 = 5/3$$

Визначаємо час виконання завдання Z_3 та можливі нові значення показників τ_j :

$$\tau_1^{(3)} = \frac{3}{2} + \frac{2}{2} = \frac{5}{2} \quad \tau_2^{(3)} = 0 + \frac{2}{1} = \frac{2}{1}$$

$$\tau_3^{(3)} = 0 + \frac{2}{1} = \frac{2}{1} \quad \tau_4^{(3)} = \frac{5}{3} + \frac{2}{3} = \frac{7}{3}$$

Найменше значення мають τ_2 та τ_3 , тому диспетчер направляє Z_3 на ОП 2.

Крок 4 (розподіл завдання Z_4)

Значення показників **поточного часу роботи**:

$$\tau_1 = 3/2; \quad \tau_2 = 2/1; \quad \tau_3 = 0 \quad \tau_4 = 5/3$$

Визначаємо час виконання завдання Z_4 та можливі нові значення показників τ_j :

$$\tau_1^{(4)} = \frac{3}{2} + \frac{1}{2} = \frac{4}{2} \quad \tau_2^{(4)} = \frac{2}{1} + \frac{1}{1} = \frac{3}{1}$$

$$\tau_3^{(4)} = 0 + \frac{1}{1} = \frac{1}{1} \quad \tau_4^{(4)} = \frac{5}{3} + \frac{1}{3} = \frac{6}{3}$$

Найменше значення має τ_3 , тому диспетчер направляє Z_4 на ОП 3.

Оскільки дії, які здійснюються на наступних кроках (5–8), вже зрозумілі, то немає сенсу розглядати їх у подробицях, а просто продемонструємо результати у вигляді зведеної таблиці.

Зведена таблиця балансування на кроках 5–8

		ОП 1	ОП 2	ОП 3	ОП 4	
Крок 5	$Z_5 = 7$	$\tau_1 = 3/2$	$\tau_2 = 2/1$	$\tau_3 = 1/1$	$\tau_4 = 5/3$	$\Theta^5 = 12/3$
		$\tau_1^{(5)} = \frac{3}{2} + \frac{7}{2} = \frac{10}{2}$	$\tau_2^{(5)} = \frac{2}{1} + \frac{7}{1} = \frac{9}{1}$	$\tau_3^{(5)} = \frac{1}{1} + \frac{7}{1} = \frac{8}{1}$	$\tau_4^{(5)} = \frac{5}{3} + \frac{7}{3} = \frac{12}{3}$	
Крок 6	$Z_6 = 4$	$\tau_1 = 3/2$	$\tau_2 = 2/1$	$\tau_3 = 1/1$	$\tau_4 = 12/3$	$\Theta^6 = 12/3$
		$\tau_1^{(6)} = \frac{3}{2} + \frac{4}{2} = \frac{7}{2}$	$\tau_2^{(6)} = \frac{2}{1} + \frac{4}{1} = \frac{6}{1}$	$\tau_3^{(6)} = \frac{1}{1} + \frac{4}{1} = \frac{5}{1}$	$\tau_4^{(6)} = \frac{12}{3} + \frac{4}{3} = \frac{16}{3}$	
Крок 7	$Z_7 = 8$	$\tau_1 = 7/2$	$\tau_2 = 2/1$	$\tau_3 = 1/1$	$\tau_4 = 12/3$	$\Theta^7 = 20/3$
		$\tau_1^{(7)} = \frac{7}{2} + \frac{8}{2} = \frac{15}{2}$	$\tau_2^{(7)} = \frac{2}{1} + \frac{8}{1} = \frac{10}{1}$	$\tau_3^{(7)} = \frac{1}{1} + \frac{8}{1} = \frac{9}{1}$	$\tau_4^{(7)} = \frac{12}{3} + \frac{8}{3} = \frac{20}{3}$	
Крок 8	$Z_8 = 9$	$\tau_1 = 7/2$	$\tau_2 = 2/1$	$\tau_3 = 1/1$	$\tau_4 = 20/3$	$\Theta^8 = 16/2$
		$\tau_1^{(8)} = \frac{7}{2} + \frac{9}{2} = \frac{16}{2}$	$\tau_2^{(8)} = \frac{2}{1} + \frac{9}{1} = \frac{11}{1}$	$\tau_3^{(8)} = \frac{1}{1} + \frac{9}{1} = \frac{10}{1}$	$\tau_4^{(8)} = \frac{20}{3} + \frac{9}{3} = \frac{29}{3}$	
Підсумк.		$\tau_1 = 16/2$	$\tau_2 = 2/1$	$\tau_3 = 1/1$	$\tau_4 = 20/3$	

Таким чином, остаточний результат балансування у розглянутому прикладі:

ОП 1: виконуються завдання Z_2 , Z_6 та Z_8 , $\tau_1 = 16/2 = 8$

ОП 2: виконується завдання Z_3 , $\tau_2 = 2/1 = 2$

ОП 3: виконується завдання Z_4 , $\tau_3 = 1/1 = 1$

ОП 4: виконуються завдання Z_1 , Z_5 та Z_7 , $\tau_4 = 20/3 = 7,67$

Мінімальний час виконання усіх завдань: $\Theta^8 = \max \{\tau_1, \tau_2, \tau_3, \tau_4\} = 8$

Рішення може бути масштабованим для будь-якої кількості завдань та ОП, оскільки не вводяться будь-які обмеження.

Висновки.

1. У статті проаналізовані методи статичного та динамічного балансування навантаження обчислювальних пристроїв в умовах, коли ці пристрої відрізняються за своєю обчислювальною потужністю.

2. Запропоновано спосіб динамічного балансування на основі універсального критерію, а саме – мінімального часу виконання усіх завдань.

3. Запропоновано в якості показника завантаженості обчислювального пристрою використовувати поточний час його роботи, тобто загальний час, що минув з початку роботи й до моменту здійснення балансування та коли ОП був завантажений виконанням завдань.

4. Доведено, що застосування цього способу дійсно забезпечує мінімальний час виконання усіх завдань на наявних ОП.

5. Наведено приклад покрокової послідовності динамічного балансування.

6. Перевірена масштабованість запропонованого способу динамічного балансування на будь-яку кількість завдань та довільну кількість обчислювальних пристроїв.

7. Визначені основні напрями розвитку запропонованого способу балансування, а саме:

– поширення способу диспетчеризації на випадки, коли загальний потік завдання не є безперервним, та виникають паузи;

– поширення способу диспетчеризації на випадок зміни кількості обчислювальних пристроїв під час балансування;

– вирішення питань надійності та захищеності диспетчеризації.

Список літератури:

1. Gantz John, Reinsel David. Extracting Value from Chaos. IDC Analyze the Future, Sponsored by EMC Corporation. June 2011. URL: <https://www.emc.com/collateral/demos/microsites/emc-digital-universe-2011/index.htm>. (дата звернення: 01.2011).
2. Thakur V., Kumar S. Load Balancing Algorithm: An Analytical Study. *The IUP Journal of Computer Sciences*. 2014. Vol. VIII. № 2. P. 25–34.
3. Mukherjee, D., Borst, S. C., van Leeuwen, J. S. H., & Whiting, P. A. Universality of Power-of- d Load Balancing Schemes. *ACM SIGMETRICS Performance Evaluation Review*. 2016. Issue 44(2). P. 36–38.
4. Hendrickson B., Devine K. Dynamic load balancing in computational mechanics. *Computer Methods in Applied Mechanics and Engineering*. 2000. Vol. 184. P. 485–500.
5. Hu Y. F., Blake R.J., Emerson D.R. An optimal dynamic load balancing algorithm. *Concurrency – Practice and Experience*. 1998. Vol. 10. P. 467–483.
6. Бо Тянь, Посыпкин М.А., Сигал И.Х. Балансировка нагрузки на основе оценок алгоритмической сложности подзадач. *Информационные технологии и вычислительные системы*. 2015. №1. С. 10–18.

7. Бельков Д.В. Алгоритмы балансировки загрузки процессоров параллельной вычислительной системы. *Наукові праці ДонНТУ*. 2006. випуск 106. С. 72–76.
8. Емельянов А. Балансировка нагрузки: основные алгоритмы и методы. *blog.selectel.ru*. 2015. URL: <http://blog.selectel.ru/balansirovka-nagruzkiosnovnye-algoritmy-i-metody>. (дата звернення: 05.02.2015).
9. Хританков А.С. Модели и алгоритмы распределения нагрузки. Алгоритмы на основе сетей СМО. *Информационные технологии и вычислительные системы*. 2009. №2. С. 65–79.
10. Hong Y.S., No J.H., Kim S.Y. DNS-based load-balancing in distributed web- server systems. *Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*. 2006. P. 4.
11. Ebada Sarhan, Atif Ghalwash, Khafagy Mohamed. Queue Weighting Load-Balancing Technique for Database Replication in Dynamic Content Web Sites. *Proceedings of the 9th WSEAS International Conference on Applied Computer Science*. 2009. P. 50–55.
12. Sharifiana Saeed, Seyed A. Motamedia, Mohammad K. Akbari. A predictive and probabilistic load-balancing algorithm for clusterbased web servers. *Applied Soft Computing*. 2011. Volume 11. Issue 1. P. 970–981.
13. Lin Zhang, Xiao-ping Li. A content-based dynamic load-balancing algorithm for heterogeneous Web server cluster. *Computer Science and Information Systems*. 2010. Vol.7(1). P. 153–162.
14. Rathore Neeraj. Performance of Hybrid Load Balancing Algorithm in Distributed Web Server System. *Wireless Personal Communications*. 2018. Volume 101. Issue 3. P. 1233–1246.
15. Игнатенко Е.Г., Бессараб В.И., Терепалов В.В. Алгоритм адаптивной балансировки нагрузки в кластерных системах. *Моделювання та інформаційні технології: Зб. наук. пр.* Київ : ПІМЕ ім. Г.Є. Пухова НАН України. 2010. Вип. 58. С. 142–150.
16. Райхлин В. А., Классен Р.К. Моделирование процессов балансировки нагрузки в распределенных СУБД, использующих ресурсы сети RUNNet. *Научный вестник Новосибирского государственного технического университета*. 2015. том 61, № 4. С. 90–100.

ДИСПЕТЧЕРИЗАЦИЯ ДИНАМИЧЕСКОЙ БАЛАНСИРОВКИ НАГРУЗКИ ВЫЧИСЛИТЕЛЬНЫХ УСТРОЙСТВ НА ОСНОВЕ ОЦЕНИВАНИЯ СОСТОЯНИЯ ТЕКУЩИХ ВЫЧИСЛЕНИЙ

В статье проанализированы методы статической и динамической балансировки нагрузки вычислительных устройств в условиях, когда эти устройства отличаются своей вычислительной мощностью. Предложен способ динамической балансировки на основе универсального критерия, а именно – минимального времени выполнения всех заданий. В качестве показателя загруженности вычислительного устройства применяются его временные характеристики. Пошаговая последовательность динамической балансировки рассмотрена на примере. Определены перспективы дальнейшего усовершенствования.

Ключевые слова: распределенные вычисления, балансировка нагрузки, динамическая балансировка, оценка загруженности, критерии балансировки, диспетчер загрузки.

DISPATCHING OF DYNAMIC LOAD BALANCING OF COMPUTING DEVICES BASED ON EVALUATING STATE OF CURRENT CALCULATIONS

In the article it was investigated methods of static and dynamic load balancing of computing devices in cases when the devices have different computing performance. It was suggested a way of dynamic balancing based on universal criterion, namely – the minimal time of completion of all the tasks. As indicator of loading of computer device, it's used its timing characteristic. It was reviewed a step-by-step sequence of dynamic balancing by using example. Additionally, it was considered possibilities of future improvements.

Key words: distributed computing, load balancing, dynamic balancing, evaluating of loading, criteria of loading, dispatcher of loading.