DOI https://doi.org/10.15407/usim.2019.01.076
УДК 303.721;004.03142

Н.А. РИБАЧОК, канд. техн. наук, старший викладач, кафедра ПЗКС ФПМ,
Нац. техн. ун-т України «Київський політехнічний інститут імені Ігоря Сікорського»
НТУУ «КПІ ім. І. Сікорського», просп. Перемоги, 37, Київ, 03056, Україна,
rybachok@pzks.fpm.kpi.ua

МОДЕЛЮВАННЯ ПОВЕДІНКИ СИСТЕМ ОРГАНІЗАЦІЇ ВОЛОНТЕРСКИХ ОБЧИСЛЕНЬ У БРАУЗЕРІ З ВИКОРИСТАННЯМ WEB WORKERS

Шляхом моделювання поведінки систем організації волонтерських обчислень у браузері із WebWorkers встановлено взаємозв'язок способів реалізації операцій серверу та властивостей системи браузерних волонтерських обчислень. Описано два набори реалізацій операцій серверу: один дає найшвидші обчислення, інший — мінімізує проблеми цих систем. Сформульовано список функцій системи, наявність яких необхідна для вирішення проблем волонтерів. Для вирішення задачі зручного керування системою наведено список функцій, необхідних адміністратору.

Ключові слова: *Web Workers, волонтерські обчислення, волонтерські обчислення у браузері, моделювання поведінки системи.*

Вступ

Волонтерські обчислення (ВО) є ідеєю розподілених обчислень, в якій користувачі (волонтери) віддають («жертвують») частину своїх обчислювальних ресурсів задля масштабних обчислювальних проектів, зменшуючи їх вартість та прискорюючи отримання результату [1].

ВО-системи є «місцем зустрічі» *Замовників*, які зацікавлені у отриманні результатів та *Волонтерів*, які хочуть вносити посильний вклад у проведення зазвичай неприбуткових ресурсномістких обчислень [2].

Системи для волонтерських обчислень допомогли отримати результати численних наукових досліджень з біології, математики, фізики, астрономії [3]. Такі обчислення також широко використовуються для обробки великих обсягів даних в *Internet*: у соціальних мережах для розпізнавання облич, логотипів та зображень, масштабування зображення, аналізу настроїв, вмісту веб-сторінок та ін. [1].

Одним з перших успішних великомасштабних проектів був *GIMPS* (*Great Internet Mersenne Prime Search*) — пошук простих чисел Мерсенна. Починаючи з 1996 р. знайдено 17 таких чисел [4].

Наступним проектом 1996 року, який здобув величезну підтримку, став *SETI@home* (*Search or Extra-Terrestrial IntelligenceatHome*) — виявлення ознак позаземного життя [5]. В результаті обчислень знайдено декілька незвичних сигналів, а сам проект став підтвердженням ідей ВО та тестовою площадкою для їх реалізації.

Ще одним з відомих проектів є *Folding@Home*. Це проект для проведення комп'ютерного моделювання згортання молекул білка [6] задля покращення розуміння виникнення захворювань, які викликаються дефектними білками (Альцгеймер, Паркін-

сон, діабет другого типу, склероз, онкологічні хвороби).

Наразі однією з найпопулярніших обчислювальних волонтерських платформ є *BOINC* (*Berkeley Open Infrastructure for Network Computing*). На сьогоднішній день *BOINC* має біля 4,5 мільйонів реєстрантів, 870,748 хостів і забезпечує середню продуктивність 27,288.517 *TeraFLOPS* [7]. *SETI@home* та *Folding@Home* є проектами на базі *BOINC*. Кількість результатів наукових досліджень, які отримані при використанні *BOINC*, просто вражають [8].

Історія розвитку систем волонтерських обчислень та класифікація існуючих рішень

Ідея ВО з'явилася близько двадцяти років тому. Термін *волонтерські обчислення* вперше був введений *Luis F.G. Sarmenta* [9] при розробці волонтерської комп'ютерної системи на базі *Java*-аплетів.

Розрізняють два підходи до реалізації ВО-систем:

- з інсталяцією спеціального програмного забезпечення (ПЗ);
- з використанням браузера (БВО — браузерні волонтерські обчислення).

Перший підхід вимагає від волонтера реєстрації на веб-сторінці, завантаження і встановлення спеціалізованого ПЗ, яке є залежним від операційної системи [10]. *BOINC* є саме такою платформою, що дещо обмежує популярність його проектів. Інсталяції додаткового програмного забезпечення можна уникнути за допомогою використання веб-браузера. У цьому випадку веб-сторінка, яка відкрита на вкладці браузера, виконує завдання, що надходять від сервера. Перевагами використання веб-браузера для волонтерів є простота, незалежність від платформи та безпека [10].

Історію розвитку БВО-систем розділяють на *три* покоління, кожне з яких відрізняється своїм набором характеристик [10].

У *першому* поколінні завдання реалізовані у вигляді *Java*-аплетів [9]. Щоб обрати завдання для обчислень зі списку на веб-сторінці, волонтеру необхідно було завантажити відповідний аплет. Іноді це спричиняло появу додаткових виринаючих вікон, що не подобалося користувачам і зменшувало популярність волонтерських обчислень.

Друге покоління БВО-систем базується на *JavaScript* і *AJAX*, але без багатопотокової підтримки. Волонтер може ініціювати обчислення, просто відвідавши сторінку, на цей раз без появи виринаючого вікна. Але ці системи мали проблеми з ефективністю та «політикою одного походження» [10]. Складні розрахунки блокували відображення веб-сторінки і уповільнювали взаємодію з нею.

Поява нових стандартів, таких як *Web-Workers*, представлених в *HTML5* дозволила проводити розподілені обчислення волонтерам у фоновому потоці браузера, використовуючи *Javascript*-код. Тепер відображення сторінки браузеру та взаємодія з користувачем може виконуватися в окремому потоці, дозволяючи користувачу швидко та ефективно взаємодіяти з системою [10]. У *третьому* поколінні користувач, обираючи завдання, завантажує сторінку з *JavaScript*, яка потім «спілкується» з сервером за допомогою техніки *AJAX*. Все, що потрібно робити волонтерам для обчислень — тримати відповідні веб-сторінки своїх браузерів відкритими протягом тривалого періоду часу.

Постановка задачі

Відносна простота реалізації БВО-систем, їх властивості та можливості сприяють створенню та дослідженню таких систем. Більшість наукових статей за цією тематикою стосується конкретної розробленої БВО-системи: описують її архітектуру та реалізацію, використання конкретних алгоритмів у відповідній предметній області та результати тестування створеного програмного продукту. Інші автори проводять загальні дослідження властивостей БВО-систем, зосереджуються на оцінці продуктивності, безпеки та вартості

виконання розподіленої обробки даних, беручи до уваги поведінку користувача та враховуючи її вплив на характеристики системи.

RomanDebski, *TomaszKrupa*, *PrzemyslawMajewski* в [11] представляють *Comcute JS* — веб-платформу для масштабних обчислень. Вони описують архітектуру, деталі реалізації і експериментальні результати пошуку простих чисел. Авторами наведено *UML*-діаграму послідовності процесу обчислення завдання, яка дозволяє прослідити передачу повідомлень між сервером, диспетчерами та клієнтами.

У [2] представлено «полегшену» БВО-систему, описано її архітектуру, функціональність і сценарій взаємодії між браузером волонтера і сервером, а також надано експериментальні результати пошуку регулярного виразу на сторінках, що завантажені з Вікіпедії.

В [10] обговорюються питання довіри та безпеки БВО, залучення та мотивування волонтерів, точності обчислень в *JavaScript*, підходи до планування завдань. Також представлено платформу *PovoCop*: її архітектуру, експериментальні результати для знаходження числа *PI* з використанням методу Монте-Карло та матричного множення.

У [1] розглянуто різні підходи до попередньої обробки вхідних даних, планування завдань, затвердження результатів обчислення завдань. Запропоновано алгоритм адаптивного планування для «непостійних» клієнтів, описано сценарій приєднання волонтера до роботи та обрахунок завдань.

На основі проведеного аналізу встановлено, що відсутній опис узагальненої моделі поведінки БВО-систем. Створення такої узагальненої моделі поведінки БВО-систем є актуальною задачею тому, що всі системи волонтерських обчислень є складними клієнт-серверними застосунками, які мають значні складнощі, як в процесі їх розробки, так і в процесі організації та управління обчисленнями. При їх розробленні важливо розуміти, які функції виконує кожен учасник взаємодії та способи реалізації цих функцій. Для моделювання поведінки таких систем необхідно вирішити наступні задачі:

- провести аналіз та узагальнення проблем БВО-систем;
- сформулювати високорівневий сценарій процесу виконання роботи при проведенні волонтерських обчислень;
- описати способи реалізації операцій сервера та їх вплив на властивості БВО-системи;
- описати функціональні можливості, які надає БВО-система волонтеру та адміністратору;
- запропонувати варіанти реалізації серверних операцій для БВО-систем та описати властивості отриманих систем.

Моделювання поведінки систем організації волонтерських обчислень у браузері

В процесі проведення БВО виконується *робота* — обчислюється результат виконання *Javascript*-функції над вхідними даними [2]. Для цього дані діляться сервером на проміжки та разом з функцією передаються сторінці браузеру волонтера, утворюючи *завдання* для обчислення (рис. 1). Щоб взяти участь у обчисленнях, волонтер повинен відкрити відповідну сторінку, обрати роботу та залишити сторінку відкритою. Створений при цьому фоновий потік сторінки браузера волонтера буде “спілкуватися” з сервером, запитуючи завдання, проводячи обчислення на наданих волонтером ресурсах та повертаючи результат. Всі отримані результати об’єднуються сервером в єдиний результат, що має відношення до виконуваної роботи.

Адміністратор системи проводить заходи щодо створення робіт, переглядає стан та результати їх виконання, налаштовує параметри системи та робіт.

Всі ВО-системи є клієнт-серверними додатками, де сервер виконує такі дії:

- попередньо обробляє вхідні дані роботи (ділить їх на проміжки);
- планує завдання між клієнтами;
- видає завдання клієнту;

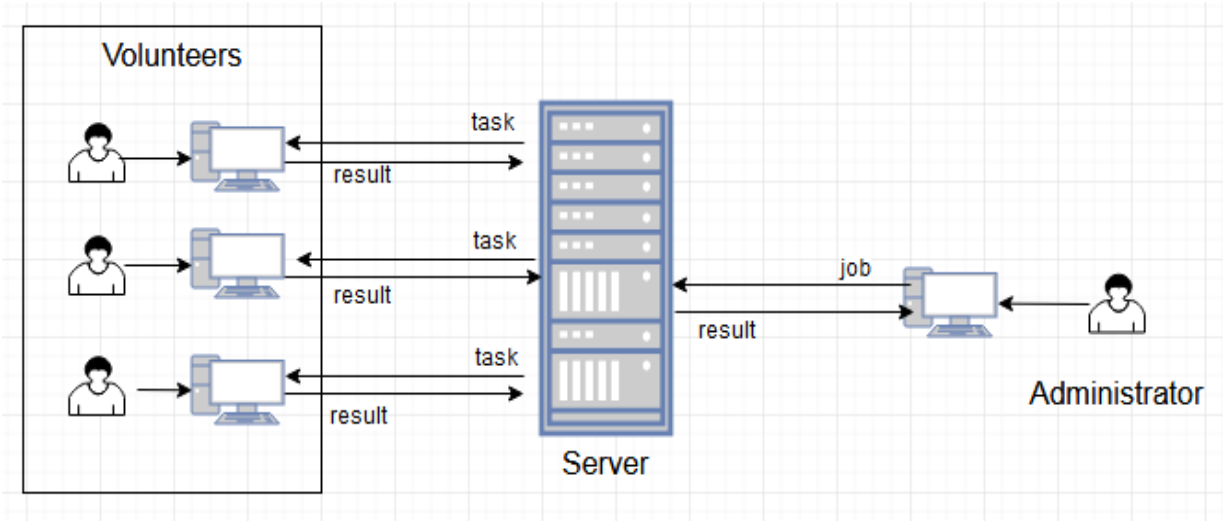

Рис. 1. Спрощена схема волонтерських обчислень

- затверджує результат обчислення завдання;
- агрегує результати роботи.

Всі необхідні обчислення над завданнями виконуються на стороні клієнтів-волонтерів. Обчислення відбуваються в браузері у фоновому режимі, тому основний потік сторінки, який відповідає за взаємодію з волонтером, не навантажується і відбувається *комфортний* діалог.

При моделюванні поведінки БВО-систем прийняті наступні припущення та обмеження:

- система не потребує реєстрації та авторизації волонтерів;
- системою не враховуються апаратні можливості пристроїв волонтерів.

Проблеми БВО-систем

При проведенні досліджень волонтерських обчислень через браузер виявлено ряд проблем. Подамо їхню класифікацію за учасниками взаємодії.

Проблеми рівня сервера

1. «Непостійність» волонтерів.

Як було зазначено вище, для проведення обчислень завдань волонтеру досить обрати роботу та тримати сторінку браузера відкритою. Волонтери можуть припинити обчислення в будь-який момент, просто закривши відповідну сторінку. Тривалість перегляду сторінки волонтером невідома і непрогнозована.

2. Підробка результатів обчислення завдань.

Існує ймовірність підробки результатів обчислення завдань. При браузерних ВО реєстрація і аутентифікація волонтерів не проводиться.

3. Затримка у поверненні результатів.

Час, за який волонтер повертає результат обчислень завдання, може бути занадто великим. Це призводить до зниження швидкості проведення обчислень.

4. Помилки у функції чи вхідних даних.

Такі проблеми можуть виникати як при неправильному внесенні даних в систему, так і через помилки в самих вхідних даних.

Проблеми рівня волонтера

5. Контроль над процесом обчислень.

5.1. Необхідність отримати згоду волонтера.

Код у браузері можна виконувати і без згоди волонтера. Волонтер повинен мати можливість ознайомитися з наявними роботами і приєднатися до обчислення обраних робіт.

5.2. Можливість відмовитися від обчислень.

Волонтер має змогу відмовитися від обчислень, просто закривши відповідну сторінку браузеру. Система повинна надавати можли-

вість волонтеру припинити обчислення, залишаючи активною сторінку для взаємодії з нею. Довірі волонтера до процесу БВО сприяє наявність засобів контролю над процесом обчислень.

6. Оцінка та контроль навантаження на систему волонтера.

Важливо, щоб обчислення контрольовано використовували ресурси системи волонтера.

6.1. Оцінка навантаження.

Волонтер повинен мати можливість оцінити використання ресурсів своєї системи.

6.2. Наявність засобів для контролю навантаження. Волонтер повинен мати можливість налаштувати використання ресурсів своєї системи. Рішення цих проблем також підвищує довіру волонтера до процесу БВО.

7. Заохочення волонтерів до участі в обчисленнях.

7.1. Оцінка вкладу у обчислення.

Для заохочення волонтерів до участі в обчисленнях можна оцінювати вклад кожного у виконання роботи.

7.2. Преміювання волонтерів.

Такі заохочення-бонуси можуть бути матеріальними чи віртуальними. Їх реалізація залежить від типу діяльності, якою займається організатор ВО: знижки на товари, якщо це комерційні сайти; бонусні нарахування, якщо це онлайн ігри; обробка ві-ео, пошук облич на фото, якщо це соціальні мережі і т.ін.

Обов'язки адміністратора

Як відзначалося вище, адміністратор системи проводить заходи щодо створення робіт, переглядає стан та результати їх виконання, налаштовує параметри системи та робіт.

Більшість з цих проблем можуть бути вирішенні за рахунок розробки ефективного сценарію процесу виконання роботи при проведенні волонтерських обчислень.

Високорівневий сценарій процесу виконання роботи при проведенні ВО

Сценарій процесу виконання роботи при проведенні волонтерських обчислень включає такі кроки:

К р о к 1. сервер розбиває вхідні дані роботи на проміжки;

К р о к 2. волонтер приєднується до обчислень роботи;

поки робота не завершена (для волонтера є необчисленні завдання) або волонтер не вирішить припинити обчислення, виконуються кроки 3–7:

К р о к 3. браузер запитує завдання для обчислень;

К р о к 4. сервер планує завдання клієнтам;

К р о к 5. сервер видає завдання клієнтам;

К р о к 6. браузер виконує обрахунок та повертає результат завдання;

К р о к 7. сервер затверджує результат обчислення завдання;

К р о к 8. сервер агрегує результати роботи.

В ньому бере учать сервер (кроки 1, 4, 5, 7, 8), волонтер (крок 2) та браузер (кроки 3, 6).

Операції сервера

Вибір способів реалізації операцій сервера допоможе у вирішенні проблем 1–4, тому розглянемо їх детальніше.

Кроки основного сценарію процесу виконання роботи при проведенні ВО

К р о к 1. Розбиття вхідних даних на проміжки.

Найбільш поширеними підходами до реалізації цього кроку є:

Таблиця 1. **Властивості системи при кроках 1 та 4**

Спосіб реалізації	Властивості
Р1П1	Дані розбиваються на однакові проміжки, довільний із необчислених передається клієнту Найпростіша реалізація, обчислення відбуваються швидко
Р2П2	Розмір проміжків поступово збільшується, перший із необчислених передається клієнту Врахування "непостійності" волонтерів (рішення проблеми 1)

▪ рівномірний (Р1) — поділ даних на проміжки однакового розміру;

▪ з поступовим збільшенням (Р2) — розмір проміжків даних починається від деякого мінімального і поступово збільшується до деякого максимального значення) [1].

К р о к 4. Планування завдань.

Планування передбачає вибір завдання, яке буде передано волонтеру.

Завдання для видачі можна обирати двома способами:

▪ вибір довільного незатвердженого і невиданого даному волонтеру завдання (П1);

▪ вибір першого незатвердженого і невиданого даному волонтеру завдання (П2).

В табл. 1 наведено властивості системи в залежності від обраних способів реалізації кроків 1 та 4.

В [1] запропоновано видавати волонтеру спочатку найменші за розміром завдання, далі поступово збільшуючи їх розмір (спосіб Р2П2). Тоді «непостійні» волонтери (які недовго знаходяться на сторінці), будуть отримувати тільки малі завдання. Чим більше часу перебуватиме волонтер на сторінці, виконуючи завдання, тим більш «відповідальні» (більші за розміром) завдання він буде отримувати. Такий підхід є вирішенням проблеми 1.

К р о к 5. Видача завдання.

Видача завдання волонтерам можлива теж двома способами:

▪ без повторень (В1);

▪ з дублюванням (В2).

У випадку В1 лише один волонтер прийме участь у обчисленні завдання, при В2 — кожне завдання для обчислень буде передано декільком волонтерам (продубльоване).

Відмітимо, що розглянуті вище варіанти розбиття та планування не впливають на способи видачі завдання клієнтам: так у випадку Р2П2 видача завдань може бути як без повторень, так і з дублюванням.

К р о к 6. Затвердження результатів завдання.

Є декілька варіантів реалізації затвердження результатів:

▪ без перевірки (З1),

▪ на підставі кворуму (З2).

У випадку З1 результат обчислень завдання затверджуються системою без перевірки правдивості.

У випадку З2 результат обчислень завдання затверджуються системою, якщо $m/n > q$, де m — кількість еквівалентних результатів обчислень проміжку, отриманих від волонтерів, n — кількість дублювань проміжку, q — поріг затвердження результатів. Чим точнішим має бути результат, тим вище значення q необхідно вказувати.

Спосіб видачі завдання клієнтам (крок 5) має вплив на вибір реалізації затвердження результатів завдання (крок 6). Очевидно, що існує залежність між способами видачі завдання та затвердженням його результатів (табл. 2)

В табл. 2 наведено властивості системи в залежності від обраних способів реалізації кроків 5 та 6.

Розширення основного сценарію процесу виконання роботи при проведенні волонтерських обчислень

Основний сценарій виконання роботи при проведенні ВО можна доповнити необов'язковими кроками, які покращують властивості системи ВО.

Таблиця 2. **Властивості системи при кроках 5 та 6**

Спосіб реалізації	Властивості
В131	Завдання обчислюється лише одним волонтером, отриманий результат затверджується як правильний Найпростіша реалізація, обчислення відбуваються швидко
В132	Неможливо реалізувати: оскільки завдання видано одному волонтеру, кворум організувати не можна
В133	Реалізація не має сенсу: завдання видано декільком волонтерам, перший результат затверджується без перевірки, а інші ігноруються
В134	Завдання видано декільком волонтерам, рішення про затвердження приймається на підставі кворуму Врахування можливості підробки результатів (рішення проблеми 2)

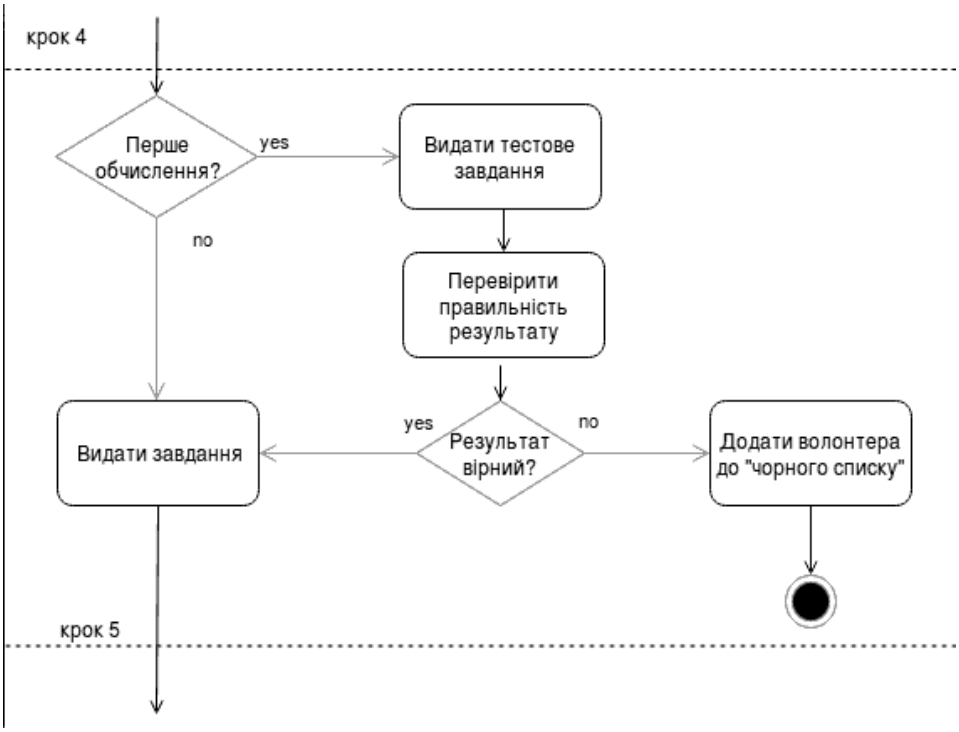

Рис. 2. Діаграма активності додавання волонтера до «чорного списку» способом Т1

Ведення «чорного списку» волонтерів. Знизити ризик підробки результатів обчислень можна за допомогою додаткових перевірок та внесення недоброчесних волонтерів до «чорного списку». Коли волонтер потрапляє до цього списку, сервер припиняє взаємодію з волонтером (завершує сесію браузера волонтера) та видаляє із системи результати обчислень даного волонтера.

Приймати рішення про занесення до «чорного списку» можна двома способами:

- на підставі тестових результатів (Т1);
- з урахуванням історії сесії волонтера (Т2).

При першому способі основний сценарій обчислення роботи буде мати розширення на кроці 5:

К р о к 5, *а*. Для першого обчислення сервер надсилає тестове завдання, відповідь на яке наперед відомо системі. Якщо повернений клієнтом результат є правильним — відбувається перехід до кроку 5. Якщо «ні» – клієнт додається до «чорного списку».

Такий підхід покращить властивості системи: при В131 він підвищить надійність результатів, а для В232 можна зменшити кількість дублювань при незмінно високому показнику надійності.

На рис. 3 наведено діаграма активності, що ілюструє крок 5.

При другому способі необхідно мати значення, яке визначатиме максимальну кількість неправильних результатів, які можуть бути в одного клієнта. Тоді основний сценарій обчислення роботи буде мати розширення на кроці 7:

К р о к 7. сервер затверджує результат обчислення завдання;

К р о к 7, *а*. якщо результат виконання волонтером завдання визнано неправильним, значення відповідного лічильника збільшується;

К р о к 7, *б*. якщо кількість обчислених волонтером результатів, які визнані сис-

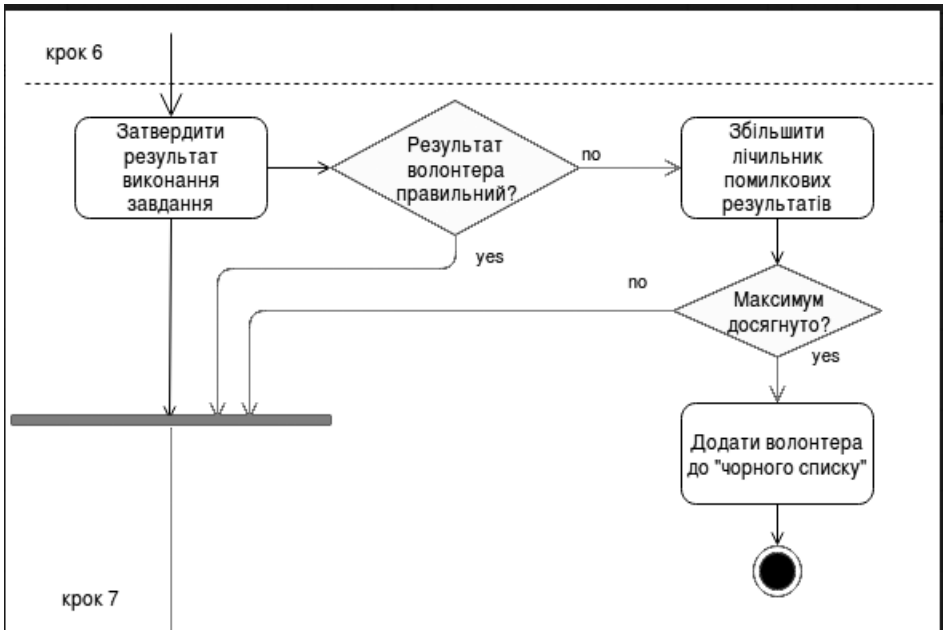

Рис. 3. Діаграма активності додавання волонтера до «чорного списку» способом Т2

темою неправильними, досягли вказаного максимального значення, система додає його до «чорного списку».

Такий спосіб перевірки покращить властивості системи: коли волонтер потрапляє в *чорний* список, його результати видаляються із системи.

Тому кількість неправильних результатів, які будуть перевірені на правдивість зменшується, що пришвидшує затвердження результатів і дає високий по-казник їх надійності (проблема 2).

На рис. 3 наведено діаграму активності, що ілюструє крок 7.

Врахування часу «старіння результатів завдання». Під часом «старіння результатів завдання» мається на увазі час, по завершенню якого переданий волонтером результат вже не враховується і попередньо видане завдання вважається невиданим.

Таке врахування буде розширенням кроку 5, яке додається до основного сценарію обчислення роботи:

К р о к 5, *б* при видачі завдання воно позначається «виданим» та для нього встановлюється час «старіння результатів». Якщо при настанні цього часу результат не був повернутий волонтером, завдання позначається як «невидане» (С1). Так можна уникнути довготривалих очікувань результатів, які можуть бути і не повернуті клієнтом (рішення проблеми 3).

Якщо за роботою видано набір завдань, але їх результати не були повернуті клієнтами вчасно (С2), можливо є помилки в даних (проблема 4), або час старіння обрано замалим і потрібно

Таблиця 3. **Функції системи та вирішувані проблеми**

Функція	Рішення проблеми
Обрати роботу для обрахунку	5.1
Зупинити обчислення поточного завдання	5.2
Переглянути завантаженість на систему	6.1
Конфігурувати параметри системи	6.2
Переглянути інформацію про стан обчислень та свій вклад у них	7.1
Отримати бонуси	7.2

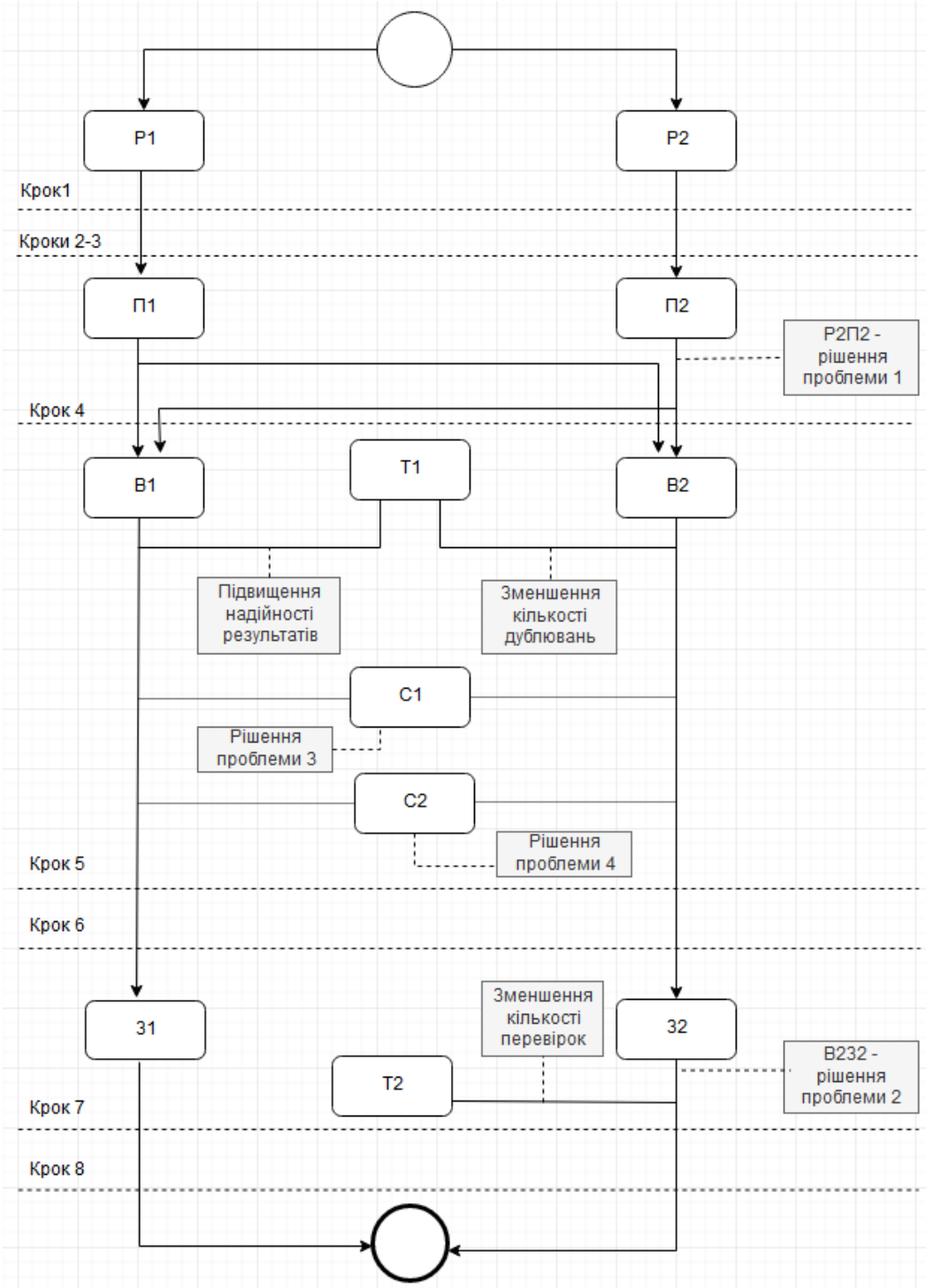

Рис. 4. Способи реалізації операцій сервера та властивості БВО-системи

корегувати його, або проводити розбиття даних на менші проміжки.

На рис. 4 наведено кроки розширеного сценарію процесу виконання роботи та їх зв'язок з рішеннями проблем, що властиві БВО-системам.

Зручність роботи волонтера

Для зручної роботи волонтера потрібно, щоб система надавала йому можливості, які пов'язані з вирішенням проблем 5–7. В табл. 3 наведено функції системи та проблеми, які ними вирішуються.

Зручність роботи адміністратора

Зручність роботи адміністратора визначається наявністю в системі функцій, які допоможуть у виконанні його обов'язків:

- керування роботами (створення, редагування, видалення);
- перегляд стану обчислень роботи та можливість отримувати повідомлення про помилки у функ-ції чи даних (операція сервера С2);
- перегляд результатів обчислень роботи;
- зміна параметрів роботи — налаштування параметрів обчислень (кількість проміжків, кількість дублювань, поріг затвердження ре-зультатів, максимальний час відповіді, максимальна кількість хибних результатів і т.д.);
- зміна параметрів системи (розміщення та параметрів БД, т.п.).

Обговорення

Очевидно, що найпростішою реалізацією БВО-системи з точки зору операцій сервера є випадок Р1П1В1З1: дані розбиваються на однакові проміжки, довільний з необчислених проміжків передається клієнту; завдання обчислюється лише одним волонтером, отриманий результат затверджується як правильний. Обчислення результатів у такій системі будуть відбуватися швидко, але їх результати будуть ненадійними.

Щоб врахувати перелічені вище проблеми 1–4 та мінімізувати їх вплив, необхідно обрати шлях Р2П2В2(Т1+С)32+Т2. При цьому вхідні дані розбиваються на проміжки з поступовим збільшенням; планування завдань відбувається шляхом вибору першого незатвердженого і невиданого даному волонтеру завдання; завдання видаються з дублюваннями (спершу видається тестовий результат, враховується час «старіння результатів завдання»); рішення про затвердження результату завдання приймається на підставі кворуму; на підставі врахування історії сесії волонтера приймається рішення про внесення його до «чорного списку». Обчислення результатів у такій системі сповільнюється за рахунок наявності дублювань та необхідності кворуму. Кількість дублювань можна зменшити за рахунок перевірок Т1 та Т2. Також перевірка Т2 сприяє швидшому досягненню кворуму, оскільки із системи видаляються неправдиві результати.

Висновки

Шляхом моделювання поведінки БВО-систем встановлено взаємозв'язок способів реалізації операцій сервера та властивостей такої системи.

Зменшити вплив "непостійності" волонтерів на процес обчислень (проблема 1) мож-на реалізувавши розподіл вхідних даних з поступовим збільшенням і почергове планування завдань.

Знизити ризик підробки результатів обчислень (проблема 2) можна за допомогою видачі завдання декільком волонтерам та затвердження результату на підставі кворуму, а також внесення недоброчесних волонтерів до «чорного списку".

Уникнути довготривалих очікувань результатів, які можуть бути і не повернуті клієнтом та виявити помилки у вхідних даних чи функції, можна за допомогою врахування часу "старіння результатів завдання".

Для забезпечення зручної роботи волонтерів та підвищення їх зацікавленості у проведенні обчислень сформульовано перелік рекомендованих до реалізації перелічених функції системи.

Для вирішення задачі зручного керування БВО-системою наведено список функцій, які потрібні адміністратору.

Матеріали можна використати на етапах проектування, документування, тестування систем проведення волонтерських обчислень у браузері.

REFERENCES

1. *Yao, Pan*. Gray Computing: A Framework for Distributed Computing with Web Browsers. PhD thesis. Vanderbilt University, 2017.
2. *http*://etd.library. vanderbilt.edu/available/etd-11192017-220210/ (visitedon 2018-03-23).
3. *Chorazyk, P.*, *Godzik, M.*, *Pietak, K.*, *Turek, W.*, *Kisiel-Dorohinicki, M.*, *Byrski, A.*, 2017. "Light weight Volunteer Computing Platform using Web Workers". Int. Conf. on Computational Science, ICCS 2017. Procedia Computer Science 108C, pp. 948–957.
4. *https*://web.archive.org/web/20100505064804/http://boincstats.boincstats.com/bam/
5. *Mersenne* Research, Inc. GIMPS history. http://www.mersenne.org/various/history.php, last check ed on 29.01.2017.
6. *http*://setiathome.berkeley.edu/
7. *https*://foldingathome.org/
8. *https*://boincstats.com/en/stats/-1/project/detail
9. *https*://boinc.berkeley.edu/wiki/Publications_by_BOINC_projects
10. *Luis, F.G. Sarmenta, 2001.* Volunteer computing. PhD thesis, Massachusetts Institute of Technology, 2001.
11. *Fabisiak, T.*, *Danilecki, A., 2017.* Browser-based Harnessing of Voluntary Computational PowerVolume/Issue: Volume 42: Issue 1 First Online: 04 Mar. 2017. Page Count: 3–42 DOI: https://doi.org/10.1515/fcds-2017-0001
12. *Debski, R.*, *Krupa, T.*, *Majewski, P., 2013.* Comcute JS: A Web Browser Based Platform for Large-scale Computations. Computer Science (AGH), 14(1).

Received 19.03.2019

N.A. Rybachok, PhD, Senior Lecturer,
Computer Systems Software Department of the Applied Mathematics Faculty,
National Technical University of Ukraine “Igor Sikorsky Kyiv Politechnic Institute”,
Peremohy Ave 37, Kyiv, Ukraine,
rybachok@pzks.fpm.kpi.ua

BROWSER-BASED VOLUNTEER
COMPUTING SYSTEMS’ BEHAVIOR MODELING USING WEB WORKERS

Introduction. The tasks of behavior modeling of Browser-Based Volunteer Computing Systems are of great practical importance for volunteers, system’s administrators, jobs owners, soft ware developers’ teams.

Purpose. The purpose of this research is behavior modeling of Browser-Based Volunteer Computing Systems, which can be applied to understanding what functions each participant performs and how these functions can be implemented. Creating such a generalized model of BBVC-systems’ behavior is an urgent task because all volunteer computing systems are client-server applications that have significant complexity, both in the process software development, and in the process of organizing and managing computations.

Methods. For behavior modeling of Browser-Based Volunteer Computing Systems, the in problems are generalized. A high-level scenario of the work calculating process is formulated. The effects of the server’s operations implementations on the system’s properties are analyzed. The opportunities provided by the BBVC systems for volunteers and administrators are described.

Results. Two sets of server’s operations implementations are described: the first provides the fastest computing, the second – minimizes the problems of BBVC systems.

The presented scheme of server’s operations methods realization and their effects on the systems’ properties can be used at different software development phases.

Conclusion. The functions of the systems for volunteers and administrators are listed, the methods of the server’s operations implementations for solving the BBVC system’s problems are formulated.

Keywords: *Web Workers, volunteer computing, Browser-Based Volunteer Computing, Browser-Based Volunteer Computing Systems, server, system’s behavior modeling.*

Н.А. Рыбачок, канд. техн. наук, старший преподаватель, кафедра ПОКС ФПМ,
Нац. техн. ун-т Украины «Киевский политехнический институт им. Игоря Сикорского»
НТУУ «КПИ им. И. Сикорского», просп. Победы, 37, Киев, 03056, Украина,
rybachok@pzks.fpm.kpi.ua

МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ СИСТЕМ ОРГАНИЗАЦИИ ВОЛОНТЕРСКИХ ВЫЧИСЛЕНИЙ В БРАУЗЕРЕ С ИСПОЛЬЗОВАНИЕМ *WEB WORKERS*

Введение. Организация волонтерских вычислений предполагает привлечение ресурсов добровольцев-волонтеров для решения трудоемких вычислительных задач. На сегодня собраны большие объемы информации, нуждающиеся в обработке. Значительно ускорить время получения результатов без привлечения дополнительных собственных ресурсов поможет разработка системы для организации волонтерских вычислений в браузере (БВО). Идеи организации таких систем описаны и доступны для применения.

Цель статьи — моделирование поведения систем организации волонтерских вычислений в браузере. Создание такой обобщенной модели поведения — актуальная задача, потому что все системы волонтерских вычислений — это сложные клиент-серверные приложения, имеющие значительные сложности как в процессе их разработки, так и в процессе организации и управления вычислениями. При разработке таких систем необходимо понимать, какие функции выполняет каждый участник взаимодействия, какие способы реализации этих функций существуют и как выбор реализаций будет влиять на свойства системы.

Методы. Проведен анализ и обобщение проблем БВО-систем. Сформулирован высокоуровневый сценарий процесса выполнения работы при проведении волонтерских вычислений. Описаны способы реализации операций сервера и определено их влияние на свойства БВО-системы. Описаны функциональные возможности, которые предоставляет БВО-система волонтеру и администратору.

Результат. Путем моделирования поведения БВО-систем установлена взаимосвязь способов реализации операций сервера и свойств этой системы. На основании полученных результатов описаны пути решения проблем, присущие БВО-системам.

Описано два набора реализаций операций сервера: первый обеспечивает самые быстрые вычисления, второй — минимизирует проблемы БВО-систем.

Выводы. Представленная схема реализации методов работы сервера и их влияния на свойства систем может быть использована на разных этапах разработки программного обеспечения.

***Ключевые слова**: Web Workers, волонтерские вычисления, волонтерские вычисления в браузере, моделирование поведения системы.*