

Метод кратковременного использования простаивающих компьютеров корпоративной сети для решения ресурсоемкой задачи

С.В. Востокин, М.А. Русин

Аннотация—В статье описаны результаты экспериментального исследования метода агрегации и использования временно простаивающих вычислительных мощностей настольных компьютеров корпоративной сети. Особенностью метода является применение парадигмы журнала событий для синхронизации состояний рабочих процессов распределенного приложения. Использование парадигмы журнала событий вместо традиционной парадигмы задач обеспечивает следующие преимущества. Более гибкое взаимодействие между частями распределенного приложения расширяет класс задач, решаемых на временно простаивающем оборудовании. Перенос функции планирования в приложение позволяет учесть особенности решаемой задачи для повышения эффективности вычислений. Ключевое преимущество исследуемого метода вычислений – это низкие накладные расходы при миграции нагрузки в случаях перехода простаивающего ресурса к использованию по основному назначению. В статье детально рассмотрены характеристики производительности тестового распределенного приложения, демонстрирующие данное преимущество. Исследование выполнено на базе приложения поиска членов последовательности A022008 OEIS. Для типовой конфигурации корпоративной сети и тестового приложения исследован режим вычислений при максимальной интенсивности обращений к процессу журнала событий. В данном режиме экспериментально оценены значения максимального ускорения, минимального времени автономного счета рабочего процесса, минимального времени решения задачи при максимальном ускорении. Результаты измерения перечисленных параметров позволяют сделать положительное заключение о применимости метода. Отмечены направления дальнейшего совершенствования метода для прикладного применения.

Ключевые слова—Гибридные вычисления, грид-вычисления, высокопроизводительные вычисления, периферийные вычисления, журнал событий, синхронизация глобального состояния.

I. ВВЕДЕНИЕ

Одним из следствий глобальной цифровизации и перехода к моделям организации бизнеса в форме цифровых предприятий является рост потребности в вычислительных ресурсах для решения

производственных задач. Значительные вычислительные ресурсы требуются для обработки операционных данных, обучения моделей искусственного интеллекта, управления производством и его бизнес-процессами, выполнения сложных численных расчетов с целью компьютерного моделирования цифровых двойников.

Экстенсивными подходами удовлетворения потребности в вычислительных ресурсах для решения задач крупной производственной или образовательной организации являются развертывание дополнительных аппаратных ресурсов и покупка виртуальных ресурсов у облачных провайдеров. Новым подходом, отмеченным как одна из технологических тенденций 2025 года в материалах исследовательской и консалтинговой компании Gartner, является развитие технологий гибридных вычислений, которые понимаются как пересмотр методов организации вычислений с целью комбинации различных вычислительных ресурсов, сетевых ресурсов и ресурсов хранения для решения вычислительных проблем [1].

Методы комбинации ресурсов, как составная часть гибридных вычислений, могут использоваться для временной агрегации простаивающих по своему основному назначению ресурсов с целью выполнения полезных вычислений. В крупных организациях, обладающих большим парком вычислительной техники, есть несколько характерных типов не полностью загруженных вычислительных ресурсов. Например в частном облаке организации имеется резерв виртуальных машин, а при запуске высокопроизводительных вычислений на кластере обычно простаивает часть узлов.

Если измерять уровень простоя количеством простаивающих узлов, то самый высокий уровень простоя – у настольных компьютеров организации. Этот вид простоя ресурсов является естественным и связан с особенностями работы в интерактивном режиме, когда требуется кратковременно обрабатывать высокие нагрузки. Тем не менее данный вид простаивающих ресурсов обычно не используется из-за технических сложностей организации вычислений. Однако значительный объем потенциально доступных ресурсов и рост потребности на вычислительные мощности у цифровых предприятий делает актуальными исследования в области организации вычислений на простаивающих настольных компьютерах в корпоративных сетях.

В статье описываются результаты

Статья получена 24 декабря 2024.

Сергей Владимирович Востокин, Самарский национальный исследовательский университет имени академика С.П. Королева (e-mail: easts@mail.ru).

Максим Алексеевич Русин, Самарский национальный исследовательский университет имени академика С.П. Королева (e-mail: skiper30175@gmail.com).

экспериментального исследования метода, использующего журнал событий для синхронизации глобального состояния частей распределенного приложения в процессе вычислений на временно простаивающих настольных компьютерах. Исследование выполнено на базе вычислительных ресурсов Межвузовского медиацентра Самарского университета.

Результаты исследования изложены в следующем порядке. Вначале приведен краткий обзор исследований в смежных областях. Далее описан предлагаемый метод организации вычислений на основе парадигмы журнала событий и синхронизации глобального состояния. Метод проиллюстрирован тестовой задачей для нагрузочного тестирования, подобранной с учетом особенностей вычислений на простаивающих ресурсах. Дан обзор номенклатуры ресурсов Межвузовского медиацентра Самарского университета. Описаны результаты оценки производительности имеющихся компьютеров при решении тестовой задачи на одном компьютере. Поставлен эксперимент по измерению предельного достижимого ускорения в режиме максимальной интенсивности обращений рабочих процессов приложения к процессу журнала событий при решении тестовой задачи на всех доступных компьютерах. Приведены характеристики, достигнутые в этом режиме вычислений. В заключении сделаны выводы о применимости предложенного метода и намечены направления его совершенствования.

II. ИССЛЕДОВАНИЯ В СМЕЖНЫХ ОБЛАСТЯХ

Примерами классических платформ, реализующих функцию использования временно простаивающих ресурсов для полезных вычислений общего назначения, являются системы HTCondor [2] и BOINC [3]. Обеспечение возможности работы в произвольных ненадежных окружениях в этих системах реализовано на основе парадигмы задач [4]. Запущенная на выполнение задача гарантированно завершается, даже в случае наличия отказов.

Управление вычислениями на базе парадигмы задач используется также в грид-вычислениях. Это близкая по назначению и методу вычислений технология при реализации которой, в отличие от исследуемой области, используется специально подготовленное, обычно однородное оборудование. Примерами современных реализаций грид-платформ для вычислений в области физики атомного ядра [5] являются системы Dirac [6], PanDA [7]. Другими важными областями применения грид-систем являются материаловедение [8], молекулярная биология [9], прикладная и теоретическая математика [10], в частности, комбинаторика [11].

Рассматриваемая в работе область вычислений имеет особенности, которые делают неэффективным непосредственное применение технологий из области грид-вычислений и требуют пересмотра методов организации вычислений в целом. Отметим три такие особенности, учитываемые в исследовании.

Относительно просто загрузить временно простаивающие ресурсы выполнением независимых задач, не требующих взаимной синхронизации. Хотя

многозадачные вычисления и допускают упорядочение задач на основе графов зависимостей [12], мы исследуем более общий случай, когда требуется произвольное взаимодействие между частями распределенного приложения.

Управление планированием задач обычно реализуется централизованно на основе общих для разных приложений алгоритмов. Однако разработчик приложения может реализовать более эффективный алгоритм планирования на основе знания особенностей своего приложения. Перенос функции планирования в приложение упрощает и делает более универсальной организацию вычислений. В системах, основанных на парадигме задач, такой подход известен как двухуровневое планирование и пилотные задания [13]. Исследуемая технология, в силу своих особенностей, изначально ориентирована на распределенное планирование на уровне приложений.

Применение задач подразумевает нетривиальный алгоритм отслеживания текущего состояния задачи (например, см. диаграмму состояний задачи в системе PanDA [14]) и, как следствие, высокие накладные расходы на запуск отдельной задачи. Для компенсации накладных расходов и сохранения высокой эффективности вычислений увеличивают время автономного счета задачи. Это, в свою очередь, увеличивает накладные расходы на миграцию нагрузки при перезапуске задачи. Использование временно простаивающих ресурсов, напротив, требует коротких интервалов автономного счета, быстрой инициализации вычислений на свободном ресурсе, малых потерь при принудительном переходе ресурса в режим работы по основному назначению.

Традиционные технологии отказоустойчивых вычислений были изначально ориентированы на сети малой пропускной способности и высокой латентности и системы хранения на механических компонентах. Новые типы сетей 1-10 Гбит, а также твердотельные накопители данных повсеместно используются в сетях предприятий, что позволяет применить новые принципы управления вычислениями на временно простаивающих ресурсах. Далее рассматривается применение известной парадигмы журнала событий [15] и метода синхронизации глобального состояния с точки зрения возможности управления вычислениями на временно простаивающих ресурсах типовой корпоративной сети.

III. МЕТОД ОРГАНИЗАЦИИ ВЫЧИСЛЕНИЙ

Метод организации вычислений основан на синхронизации локальных состояний компонентов распределенного приложения, запускаемых на ресурсах во время простоя. Такая синхронизация выполняется посредством сервиса для записи-чтения локальных приращений состояния клиентами – рабочими процессами приложения. Если ведется упорядоченная запись приращений состояния и известно начальное состояние вычислений, приложение сможет восстановить всю последовательность состояний, обрабатывая по очереди приращения состояния.

В текущем состоянии планируются возможные переходы в новое состояние. Если таких переходов несколько, то их можно вычислять одновременно в различных рабочих процессах. Вычисление переходов в

новое состояние из текущего является аналогом выполнения задач. Результат выполнения такой “псевдо задачи” – это событие, помещаемое в журнал событий. Событиям присваиваются порядковые номера.

Для фактического перехода в новое состояние требуется считать из журнала очередное событие с результатом выполнения “псевдо задачи” и вычислить новое состояние как функцию от события и текущего состояния. В новом состоянии имеются новые потенциальные переходы в следующие состояния и так до тех пор, пока в текущем состоянии не будет возможности перехода в новое состояние.

На подобных принципах основана работа распределенных систем контроля версий (Git, Mercurial) и блокчейн-систем.

IV. ТЕСТОВАЯ ЗАДАЧА

Для нагрузочного тестирования использован алгоритм нахождения чисел p в группе из шести простых чисел p , $p+4$, $p+6$, $p+10$, $p+12$, $p+16$. Числа p образуют последовательность, классифицируемую в каталоге “Онлайн-энциклопедия целочисленных последовательностей” [16] под номером A022008 [17].

Используя описанный метод организации вычислений, члены последовательности A022008 можно искать следующим способом. Входными данными алгоритма являются интервал поиска членов последовательности и количество его разбиений.

В каждом рабочем процессе приложения ведется учет рассмотренных и не рассмотренных интервалов. Также ведется таблица простых чисел, используемая для проверки текущего числа на простоту и заполняемая по мере рассмотрения интервалов. В начальном состоянии каждого процесса все интервалы помечены как не рассмотренные, а таблица простых чисел пуста.

Все процессы выполняют следующий алгоритм. В текущем состоянии для возможного перехода в новое состояние выбирается произвольный, ранее не рассмотренный, интервал. Таблица простых чисел, если это необходимо, пополняется таким образом, чтобы можно было провести тест всех чисел внутри выбранного интервала на простоту.

Выполняется сканирование интервала для нахождения шестерок близкорасположенных простых чисел. Список первых чисел в найденных шестерках, а также номер рассмотренного интервала, помещаются в журнал событий.

Переход в новое состояние осуществляется путем запроса очередного выполненного интервала из журнала событий, выставления отметки “интервал рассмотрен” в новом состоянии. Чтение новых событий выполняется до тех пор пока: (1) все интервалы не будут отмечены как рассмотренные; (2) запрос очередного события не приведет к ошибке. В случае (1) вычисления завершаются, в случае (2) процесс возвращается к началу алгоритма и обрабатывает следующий интервал.

Задача построения последовательности A022008 выбрана для исследования предлагаемого метода вычислений в силу своей простоты, понятности, проверяемости; популярности проблемной области

поиска простых чисел для использования простаивающих компьютеров в научных целях (например, в проекте PrimeGrid [18]); а также в силу необходимости управления состоянием рабочих процессов.

V. ОБЗОР ВЫЧИСЛИТЕЛЬНЫХ РЕСУРСОВ

Исследования проводились на базе компьютеров корпоративной сети Межвузовского медиацентра Самарского университета. В табл. 1 показаны типы компьютеров, их основные характеристики и располагаемое количество.

Таблица 1 – Типы компьютеров, используемые в экспериментах.

Условное обозначение типа	Модель, производитель	Процессор, (ядер/логических процессоров)	Память, (емкость/тип)	Диск, (емкость/тип)	Кол-во
Type 1	PRO AP243T 12M, MSI	Intel(R) Core(TM) i5-12400 (6/12)	16 ГБ/DDR4	500 ГБ/SSD	20
Type 2	310H5, iRU	Intel(R) Core(TM) i5-10400 CPU (6/12)	8 ГБ/DDR4	440 ГБ/SSD	30
Type 3	Q190, Lenovo	Intel(R) Celeron(R) CPU 1017U (2/2)	4 ГБ/DDR3	500 ГБ/HDD	68

Компьютеры Медиацентра используются как терминалы для доступа к виртуальным машинам VMware vSphere в корпоративном облаке Самарского университета, работы с ресурсами сети Интернет (Type 3 в табл.1), а также как самостоятельные устройства, например, для работы с системами САПР (Type 1, 2 в табл.1). Компьютеры находятся в общем сегменте локальной сети с пропускной способностью 1 Гбит/с и оснащены сетевыми адаптерами Realtek PCIe GBE Family Controller. На компьютерах Type 3 установлена операционная система Windows 7 Pro SP1 32, на Type 1 и 2 – операционная система Windows 10 Pro 64.

Исследуемая вычислительная сеть обладает типовыми для корпоративных университетских сетей параметрами, а характер ее использования в учебном процессе допускает возможность выполнения дополнительных сторонних нагрузок. Далее рассматриваются технические возможности такого использования.

VI. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ ОТДЕЛЬНЫХ КОМПЬЮТЕРОВ

Производительность в режиме последовательного счета оценивалась по времени нахождения членов последовательности A022008 в интервале до 100.000.000 (первые 82 члена). Тест реализован в виде консольного приложения на языке C#. Среднее время счета на компьютере Type 1 составило (в секундах) ~106 с, Type 2 ~306 с, Type 3 ~824 с. Относительная производительность типов компьютеров в режиме

последовательного счета для рассматриваемой задачи представлена в табл. 2.

Таблица 2 – Относительная производительность типов компьютеров в режиме последовательного счета.

Производительность	Туре 1	Туре 2	Туре 3
В сравнении с Туре 1	1	0,347765628	0,1291833329
В сравнении с Туре 2	2,875499818	1	0,3714666502
В сравнении с Туре 3	7,74093668	2,692031706	1

Из табл. 2 видно, что наименее производительные компьютеры Туре 3 уступают самым мощным компьютерам в $\sim 7,7$ раз, а Туре 2 – в $\sim 2,7$; компьютеры типов Туре 1, 2 различаются по производительности в $\sim 2,9$ раз в режиме последовательного счета.

Далее задача нахождения членов последовательности A022008 в интервале до 100.000.000 решалась в режиме параллельного счета. Тест реализован в виде трех консольных компонент развертывания: рабочий процесс, журнал событий, управляющий процесс. Язык реализации – C#; для управления журналом событий использовалась библиотека SQLite; для организации сетевого взаимодействия – библиотека gRPC. Компоненты “журнал событий” и “управляющий процесс” были развернуты на компьютере Туре 1.

Вначале были найдены оптимальные количества рабочих процессов для компьютеров каждого типа, обеспечивающие наименьшее время счета (табл. 3). Затем на этой основе рассчитаны относительные производительности компьютеров в режиме параллельного счета (табл. 4).

Таблица 3 – Результаты экспериментов по определению оптимального количества рабочих процессов для типов компьютеров.

Условное обозначение типа	Лучшее время, с	Кол-во рабочих процессов для лучшего времени	Время при достижении 100% загрузки, с	Кол-во рабочих процессов при достижении 100% загрузки
Туре 1	20,5970187	7	20,71102467	8
Туре 2	39,9109708	13	47,23822103	9
Туре 3	431,3943321	2	431,3943321	2

Таблица 4 – Относительная производительность типов компьютеров в режиме параллельного счета.

Производительность	Туре 1	Туре 2	Туре 3
В сравнении с Туре 1	1	0,5160741091	0,04774522326
В сравнении с Туре 2	1,937706198	1	0,09251621505
В сравнении с Туре 3	20,94450359	10,80891603	1

В экспериментах табл. 3 и во всех последующих экспериментах для параллельных расчетов общее количество интервалов выбиралось таким образом, чтобы на один рабочий процесс приходилось по 30 интервалов. На основании предварительного имитационного моделирования в ранее выполненном исследовании [19] установлено, что это обеспечивает долю избыточных вычислений на уровне 7%, а дальнейшее увеличение числа разбиений на один

рабочий процесс не приводит к заметному уменьшению этой доли.

Из табл. 3 следует, что режим оптимальной производительности сильно зависит от особенностей компьютеров. У компьютеров Type 1 и Type 2 при одинаковом количестве ядер и логических процессоров он достигается при 7 и 13 рабочих процессах соответственно, а показатели загрузки процессора в диспетчере задач не свидетельствуют однозначно о выходе на режим максимальной производительности. Ускорение счета при решении задачи в параллельном режиме на компьютере Type 1 составило ~5,2 раза, Type 2 ~7,7 раз, Type 3 ~1,9 раз.

Из табл. 4 видно, что компьютеры Type 3 уступают самым мощным компьютерам Type 1 по производительности в режиме параллельного счета в почти в 21 раз, а Type 2 – более чем в 10 раз.

VII. ОЦЕНКА СОВОКУПНОЙ ПРОИЗВОДИТЕЛЬНОСТИ КОМПЬЮТЕРОВ

Проведем оценку совокупной производительности доступного для экспериментов парка компьютеров, приняв за единицу производительности производительность одного компьютера соответствующего типа в режиме параллельного счета. Используем данные из табл. 1. и табл. 4. Результаты оценки приведены в табл. 5.

Таблица 5 – Количество располагаемых компьютеров, приведенное в соответствии с производительностью их типов.

Количество	Type 1	Type 2	Type 3	Всего
Приведенное по производительности Type 1	20	15,48222327	3,246675182	38,72889846
Приведенное по производительности Type 2	38,75412396	30	6,291102623	75,04522659
Приведенное по производительности Type 3	418,8900718	324,2674809	68	811,1575527

Из табл. 5 видно, что совокупная производительность всех доступных компьютеров соответствует производительности 38 компьютеров Type 1, 75 компьютеров Type 2 или 811 компьютеров Type 3.

Оценка вклада компьютеров каждого типа в общую производительность в режиме параллельного счета приведена в табл. 6. Из табл. 6 следует, что наибольший вклад, превышающий 50%, вносят компьютеры Type 1, хотя их общее количество 20 штук из 118. А вклад компьютеров Type 3 – чуть более 8%, хотя их количество, 68 штук, наибольшее.

Таблица 6 – Вклад компьютеров в совокупную производительность с учетом производительности их типов.

	Type 1	Type 2	Type 3	Всего
Количество в типе	20	30	68	118
% производительности	51,64102465	39,97589369	8,383081655	100

Рассмотрим оценку ускорения счета на всех доступных компьютерах по сравнению со временем последовательного выполнения (табл. 7). Оценка приведена для каждого типа компьютеров, отдельно в столбцах с 2-4 приведена расшифровка вклада в ускорение компьютеров каждого типа.

Таблица 7 – Совокупное ускорение компьютеров, приведенное относительно времени счета в последовательном режиме для каждого типа.

Ускорение	Все из Type 1	Все из Type 2	Все из Type 3	Общее
Относительно Type 1	103,3267005	79,98635235	16,7734117	200,0864645
Относительно Type 2	297,1159084	230,0007416	48,2319423	575,3485924
Относительно Type 3	799,8454458	619,1692888	129,8419179	1548,856653

Исходя из найденных оценок производительности, в дальнейших экспериментах будем использовать только компьютеры Type 1 и Type 2, так как их ожидаемый совокупный вклад в общую производительность составляет более 90%. Ускорение при вычислении на всех доступных компьютерах будем оценивать относительно времени счета на самом производительном типе компьютеров Type 1. Тогда по нашим предварительным расчетам оценка максимального ускорения при использовании всех компьютеров Type 1 и Type 2, согласно табл. 7, составит 183 раза. Далее проводится серия экспериментов по исследованию режима вычислений с максимальной интенсивностью обращений к процессу журнала событий.

VIII. ПРОИЗВОДИТЕЛЬНОСТЬ В РЕЖИМЕ МАКСИМАЛЬНОЙ ИНТЕНСИВНОСТИ ОБРАЩЕНИЙ К ЖУРНАЛУ СОБЫТИЙ

С точки зрения использования неспециализированного, кратковременно доступного для сторонних вычислений оборудования, представляет интерес не только достижимое ускорение, но и возможность быстро занимать свободные ресурсы и быстро их освобождать. В случае принудительной приостановки вычислений (при возобновлении использования компьютеров по основному назначению) потери на пересчет прерванных вычислений должны быть минимальными. Это свойство определяется временем автономного счета рабочего процесса приложения (временем вычисления “псевдо задачи”). Чем меньше это время, тем оперативнее приложение занимает и освобождает простаивающие ресурсы. Это время будет тем меньше, чем быстрее обрабатывается поток запросов на запись приращений локальных состояний процессом журнала событий. Очевидно, что интенсивность записи имеет физический предел. Целью дальнейших экспериментов является нахождение данного предела и характеристик производительности на его границе.

Для нахождения параметров режима вычислений с максимальной интенсивностью обращений к процессу

журнала событий применялась следующая методика и настройки. Для экспериментов использовалось 20 компьютеров Type 1 и 30 компьютеров Type 2. На них запускались рабочие процессы приложения в количестве 140 процессов на Type 1 (по 7 на компьютер) и 390 процессов на Type 2 (по 13 процессов на компьютер), всего 530 рабочих процессов. С учетом обработки в среднем 30 подинтервалов одним процессом, общее число подинтервалов составило 15900 (15900 “псевдо задач”).

Для поиска границ исследуемого режима по очереди запускалось вычисление членов последовательности A022008 для диапазонов от 100 миллионов до 8 миллиардов. Проводился замер времени счета и загрузки процессоров у компьютеров Type 1. Результаты замеров представлены в табл. 8.

Таблица 8 – Определение границ режима вычислений с максимальной интенсивностью обращений к процессу журнала событий.

Граница диапазона поиска, миллионы	Время счета, с	Интенсивность потока обращений, записей/с	Загрузка Type 1 по данным диспетчера задач, %
100	109,673203	144,9761616	1
500	110,7727318	143,53713	2
1000	109,6077323	145,0627585	10
2000	112,9832738	140,7287952	30
3000	101,8924577	156,0468788	50
4000	127,7638989	124,4482998	80
5000	168,7200318	94,23895806	80
6000	218,9955606	72,6042115	90
7000	273,104526	58,21946722	90
8000	331,0230042	48,03291553	90

Из табл. 8 видно, что несмотря на повышение вычислительной сложности, время вычислений примерно одинаковое для диапазонов до 3 миллиардов, а нагрузка компьютеров Type 1 не достигает расчетных 80-90% (режима максимальной производительности). Это говорит о перегрузке процесса журнала событий при интенсивности потока обращений около 150 записей в секунду. Когда время счета начинает возрастать, компьютеры начинают работать в режиме полной загрузки (начиная от 80% и далее до 90% в табл. 8). Процесс журнала событий, напротив, работает с резервом по производительности.

Порог перехода в режим полной загрузки компьютеров достигается при поиске чисел последовательности A022008 в диапазоне до 4 миллиардов. Время счета на 50 компьютерах для этого диапазона составило около 128 секунд. Время последовательного счета этой же задачи, в силу его длительности, оценивалось по времени параллельного счета на одном компьютере Type 1. Оно составило около 1 часа и 10 минут (4229,34 с). Зная ускорение параллельного счета задачи на компьютере Type 1 (примерно 5,2 раза), ожидаемое время последовательного счета в диапазоне до 4 миллиардов на компьютере Type 1 можно оценить примерно в 6

часов. Итоговое измеренное ускорение относительно компьютера Type 1 в параллельном режиме составило 33,98; расчетное ускорение относительно компьютера Type 1 в последовательном режиме – 175,57, что близко к предварительной оценке ~183.

На пороге перехода в режим полной загрузки компьютеров среднее время автономного счета (время выполнения “псевдо задачи”) составило 4,25 секунды. Интенсивность обращений к журналу событий составила 124,45 записи в секунду, что соответствует времени одной транзакции около 8 миллисекунд.

IX. ЗАКЛЮЧЕНИЕ

Эксперименты показали, что даже стандартные технологии реализации журнала событий (gRPC для связи и SQLite для хранения) позволяют на типовом оборудовании корпоративной сети организовывать эффективное использование простаивающих ресурсов в течении коротких интервалов времени (около 2 минут) с высоким ускорением (около 175 раз). Также достижимы короткие интервалы автономного счета рабочих процессов (в пределах 5 секунд), следовательно, накладные расходы на миграцию рабочих процессов невелики.

В целом полученные результаты экспериментов свидетельствуют о применимости предлагаемого метода для полезного использования временно простаивающих ресурсов настольных компьютеров корпоративной сети. Дальнейшие исследования будут направлены на выявления факторов, ограничивающих производительность журнала событий, и на разработку оптимизированных реализаций метода.

БЛАГОДАРНОСТИ

Авторы выражают благодарность сотрудникам Межвузовского медиацентра Самарского университета, начальнику Центра телекоммуникаций Самарского университета Баскакову А.В. за предоставленную возможность проведения вычислительных экспериментов и технические консультации.

БИБЛИОГРАФИЯ

- [1] Gartner Top 10 Strategic Technology Trends for 2025. – URL: <https://www.gartner.com/en/articles/top-technology-trends-2025> (дата обращения: 20.12.2024). – Текст : электронный.
- [2] HTCondor Overview. – URL: <https://htcondor.org/htcondor/overview/> (дата обращения: 20.12.2024). – Текст : электронный.
- [3] BOINC. – URL: <https://boinc.berkeley.edu/> (дата обращения: 20.12.2024). – Текст : электронный.
- [4] Fundamentals of grid computing: theory, algorithms and technologies: Chapman & Hall/CRC numerical analysis and scientific computing. Fundamentals of grid computing / F. Magoulès. – Boca Raton: CRC Press, 2010. – 298 С.
- [5] The Worldwide LHC Computing Grid (WLCG). – URL: <https://home.cern/science/computing/grid> (дата обращения: 20.12.2024). – Текст : электронный.
- [6] Documentation — DIRAC Documentation. – URL: <https://dirac.readthedocs.io/en/latest/> (дата обращения: 20.12.2024). – Текст : электронный.
- [7] Welcome to PanDA documentation — PanDAWMS documentation. – URL: <https://panda-wms.readthedocs.io/en/latest/> (дата обращения: 20.12.2024). – Текст : электронный.
- [8] USPEX overview. – URL: <https://uspex-team.org/en/uspex/overview> (дата обращения: 20.12.2024). – Текст : электронный.

- [9] Widlak, W. High-Throughput Technologies in Molecular Biology / W. Widlak. – Текст: электронный // Molecular Biology / D. Hutchison; W. Widlak. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. – Т. 8248. – С. 139-153.
- [10] Prime numbers and random walks in a square grid / A. Fraile, O. Kinouchi, P. Dwivedi [et al.] // Physical Review E. – 2021. – Т. 104. – № 5. – С. 054114.
- [11] Albertian, A. M. The use of heterogeneous computing nodes in grid systems in solving combinatorial problems / A. M. Albertian, I. I. Kurochkin, E. I. Vatutin // Izvestiya SFedU. ENGINEERING SCIENCES. - 2022. - № 7. - С. 142-153
- [12] DAGMan Applications — HTCondor Manual 8.8.17 documentation. – URL: https://htcondor.readthedocs.io/en/v8_8/users-manual/dagman-applications.html (дата обращения: 20.12.2024). – Текст: электронный.
- [13] Turilli, Matteo, Mark Santcroos, and Shantenu Jha. "A comprehensive perspective on pilot-job systems." ACM Computing Surveys (CSUR) 51.2 (2018): 1-32.
- [14] Basic Concepts — PanDA WMS documentation. – URL: <https://panda-wms.readthedocs.io/en/latest/terminology/terminology.html#task> (дата обращения: 22.12.2024). – Текст: электронный.
- [15] What is an Event Log? Contents and Use | CrowdStrike. – URL: <https://www.crowdstrike.com/en-us/cybersecurity-101/next-generation/event-logs/> (дата обращения: 20.12.2024). – Текст: электронный.
- [16] The On-Line Encyclopedia of Integer Sequences (OEIS). – URL: <https://oeis.org/> (дата обращения: 20.12.2024). – Текст: электронный.
- [17] A022008 - OEIS. – URL: <https://oeis.org/A022008> (дата обращения: 20.12.2024). – Текст: электронный.
- [18] PrimeGrid. – URL: <https://www.primegrid.com/> (дата обращения: 20.12.2024). – Текст: электронный.
- [19] Vostokin, S., Rusin, M. Experiments with the A022008 Sequence Generator to Study Distributed Computing Based on State Synchronization Service / V. Voevodin et al. (Eds.): RuSCDays 2024, LNCS 15407, pp. 75–89, 2025. doi:10.1007/978-3-031-78462-0_6.

Method of Short-Term Use of Idle Corporate Network Computers to Solve a Resource-Intensive Problem

S. Vostokin, M. Rusin

Abstract— The article describes the results of an experimental study of utilizing temporarily idle computing power in a corporate network. A distinctive feature of the method is the implementation of the event log paradigm to synchronize the states of distributed worker processes. The paradigm provides the following advantages over ‘tasks’. As a general method it expands the class of problems solved on temporarily idle equipment. Transferring the scheduling function to the application allows one to improve the efficiency. The main advantage of the method is the low overhead of load migration. This paper provides a detailed discussion of metrics that demonstrate this advantage when searching for members of the OEIS A022008 sequence. For a typical corporate network, the mode with the maximum intensity of calls to the event log process was studied. In this mode, the values of maximum speedup, minimum time of autonomous calculation of the worker processes, minimum time of solving the problem at maximum speedup are experimentally estimated. The results allow us to make a positive conclusion about the applicability of the method. Directions for further improvement of the method are noted.

Keywords—Hybrid computing, grid computing, high-performance computing, edge computing, event log, global state synchronization.

REFERENCES

- [1] Gartner Top 10 Strategic Technology Trends for 2025. – URL: <https://www.gartner.com/en/articles/top-technology-trends-2025> (date accessed: 20.12.2024). – Text : electronic.
- [2] HTCondor Overview. – URL: <https://htcondor.org/htcondor/overview/> (date accessed: 20.12.2024). – Text : electronic.
- [3] BOINC. – URL: <https://boinc.berkeley.edu/> (date accessed: 20.12.2024). – Text : electronic.
- [4] Fundamentals of grid computing: theory, algorithms and technologies: Chapman & Hall/CRC numerical analysis and scientific computing. Fundamentals of grid computing / F. Magoulès. – Boca Raton: CRC Press, 2010. – 298 C.
- [5] The Worldwide LHC Computing Grid (WLCG). – URL: <https://home.cern/science/computing/grid> (date accessed: 20.12.2024). – Text : electronic.
- [6] Documentation — DIRAC Documentation. – URL: <https://dirac.readthedocs.io/en/latest/> (date accessed: 20.12.2024). – Text : electronic.
- [7] Welcome to PanDA documentation — PanDAWMS documentation. – URL: <https://panda-wms.readthedocs.io/en/latest/> (date accessed: 20.12.2024). – Text : electronic.
- [8] USPEX overview. – URL: <https://uspex-team.org/en/uspex/overview> (date accessed: 20.12.2024). – Text : electronic.
- [9] Widlak, W. High-Throughput Technologies in Molecular Biology / W. Widlak. – Text: electronic // Molecular Biology / D. Hutchison; W. Widlak. – Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. – T. 8248. – C. 139-153.
- [10] Prime numbers and random walks in a square grid / A. Fraile, O. Kinouchi, P. Dwivedi [et al.] // Physical Review E. – 2021. – T. 104. – № 5. – C. 054114.
- [11] Albertian, A. M. The use of heterogeneous computing nodes in grid systems in solving combinatorial problems / A. M. Albertian, I. I. Kurochkin, E. I. Vatutin // Izvestiya SFedU. ENGINEERING SCIENCES. - 2022. - № 7. - C. 142-153
- [12] DAGMan Applications — HTCondor Manual 8.8.17 documentation. – URL: https://htcondor.readthedocs.io/en/v8_8/users-manual/dagman-applications.html (date accessed: 20.12.2024). – Text : electronic.
- [13] Turilli, Matteo, Mark Santcroos, and Shantenu Jha. "A comprehensive perspective on pilot-job systems." ACM Computing Surveys (CSUR) 51.2 (2018): 1-32.
- [14] Basic Concepts — PanDA WMS documentation. – URL: <https://panda-wms.readthedocs.io/en/latest/terminology/terminology.html#task> (date accessed: 22.12.2024). – Text : electronic.
- [15] What is an Event Log? Contents and Use | CrowdStrike. – URL: <https://www.crowdstrike.com/en-us/cybersecurity-101/next-gen-siem/event-logs/> (date accessed: 20.12.2024). – Text : electronic.
- [16] The On-Line Encyclopedia of Integer Sequences (OEIS). – URL: <https://oeis.org/> (date accessed: 20.12.2024). – Text : electronic.
- [17] A022008 - OEIS. – URL: <https://oeis.org/A022008> (date accessed: 20.12.2024). – Text : electronic.
- [18] PrimeGrid. – URL: <https://www.primegrid.com/> (date accessed: 20.12.2024). – Text : electronic.
- [19] Vostokin, S., Rusin, M. Experiments with the A022008 Sequence Generator to Study Distributed Computing Based on State Synchronization Service / V. Voevodin et al. (Eds.): RuSCDays 2024, LNCS 15407, pp. 75–89, 2025. doi:10.1007/978-3-031-78462-0_6.