

# Применение модели bag-of-tasks с централизованным хранилищем для распределенной сортировки большого массива данных

С.В. Востокин<sup>1</sup>, И.В. Бобылева<sup>1,2</sup>

<sup>1</sup>Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

<sup>2</sup>АО «Ракетно-космический центр «Прогресс», Земеца 18, Самара, Россия, 443009

**Аннотация.** В работе обсуждается применение модели программирования «портфель задач» (bag of tasks) для сортировки большого массива данных. Выбор объекта исследования определяется общностью его алгоритмической структуры с различными задачами из области анализа данных, включая корреляционный анализ, частотный анализ и индексацию данных. Алгоритм сортировки представляет собой поэтапную сортировку блоков массива, а затем их попарное слияние. В конце сортировки данные в блоках образуют упорядоченную последовательность. Порядок сортировки и слияния задается статическим ориентированным ациклическим графом. Алгоритм сортировки реализован с использованием библиотеки MPI на языке C++ с централизованным хранением блоков данных в управляющем процессе. Особенностью реализации является передача блоков между управляющим и рабочим MPI-процессами в каждой задаче. Проведенное экспериментальное исследование алгоритма подтвердило гипотезу о том, что интенсивный обмен данными, вызванный централизованной природой модели «портфель задач», не приводит к потере производительности. Применение рассмотренной модели обработки данных позволяет снизить технические требования к аппаратному и программному обеспечению.

## 1. Введение

В работе исследуется возможность применения модели управления распределенными вычислениями «портфель задач» (bag of tasks) для решения прикладных задач анализа и обработки больших информационных массивов.

Для решения задач, связанных с обработкой и анализом больших объемов данных, обычно используется дорогостоящее специализированное аппаратное и программное обеспечение. Однако в организациях, где ведется такая обработка, уже имеется большой парк настольных компьютеров, рабочих станций, ноутбуков сотрудников, а часто и кластерные системы, ресурсы которых в обычных условиях эксплуатации используются не полностью. Поэтому, с точки зрения возможного снижения затрат на аппаратное обеспечение, при решении задач обработки данных перспективным является применение всей имеющейся аппаратной инфраструктуры организации [1].

Для вычислений на неспециализированном программно-аппаратном обеспечении применяются особые модели программирования и поддерживающие их программные платформы [2]. Одной из таких моделей программирования является модель «портфель-задач» (bag of tasks). Эта

модель имеет коммуникационную топологию процессов типа «звезда» и состоит из одного управляющего процесса и группы рабочих процессов. Рабочие процессы обрабатывают задачи, поступающие от управляющего, и возвращают ему результат. Рабочие процессы не обладают состоянием. Общее состояние вычислений, так называемый «портфель задач», отслеживается управляющим процессом. Это состояние может обновляться по мере завершения текущих задач в рабочих процессах.

Представление параллельных вычислений в модели «портфель-задач» имеет ряд преимуществ при разработке приложений, работающих в гетерогенном ненадежном программно-аппаратном окружении.

Во-первых, приложение имеет единственную точку отказа – управляющий процесс. Отказ рабочих процессов не является критичным.

Во-вторых, только управляющий процесс обязан принимать соединения, рабочие процессы могут поддерживать только исходящие соединения.

В-третьих, для определения нового алгоритма в модели «портфель задач» достаточно определить 4 последовательные процедуры: проверка наличия задачи в текущем состоянии «портфеля задач» для передачи рабочему процессу; формирование новой задачи и обновление состояния «портфеля задач»; обработка задачи в рабочем процессе; прием результата выполнения задачи и обновление состояния «портфеля задач». Другие части кода являются универсальными.

В-четвертых, схема управления в модели «портфель задач» в большинстве случаев не требует специальных методов балансировки нагрузки. Это является удобным, так как производительность компьютеров в вычислительной сети предприятия может значительно отличаться.

Обладая перечисленными преимуществами, дизайн модели «портфель задач» имеет потенциальный недостаток, заключающийся в том, что управляющий процесс может оказаться узким местом при передаче данных. Поэтому применимость модели в конкретном случае определяется совместными характеристиками задачи и реализующего её оборудования, что требует специального исследования.

Целью данной работы являлась экспериментальная проверка применимости модели «портфель задач» для вычислений на кластерных системах без оптимизации трафика между управляющим и рабочими процессами в задаче блочной сортировки большого массива данных.

## **2. Метод экспериментального исследования**

В качестве модели характерной задачи анализа данных в исследовании рассматривается задача блочной сортировки большого массива данных. Выбор задачи обусловлен общностью её алгоритмической структуры с различными прикладными задачами анализа данных, включая корреляционный анализ, частотный анализ, индексацию данных.

Алгоритм сортировки представляет собой поблочную сортировку массива данных, за которой следует попарное слияние блоков методом «каждый с каждым». В конце сортировки данные в блоках образуют упорядоченную последовательность. В «портфель задач» помещаются задачи двух типов: сортировка блока и слияние двух блоков. Оба типа задач модифицируют содержимое блоков путем перестановки элементов, при этом общее количество блоков и их размер не изменяется. Порядок задач сортировок и слияний задан статическим графом зависимостей задач, который в процессе вычислений интерпретируется управляющим процессом.

Особенностью исследуемого алгоритма является то, что в нем происходит передача больших блоков данных от управляющего процесса к рабочим процессам и обратно к управляющему процессу при выполнении каждой задачи. В исследовании намеренно не применяются вспомогательные технологии, оптимизирующие трафик, например, кэширование блоков на рабочих процессах.

В качестве модели оборудования в вычислительных экспериментах применяется кластерная система коллективного использования «Сергей Королев» Суперкомпьютерного центра Самарского университета ([hpc.ssau.ru](http://pc.ssau.ru)). Выбор обусловлен простотой программирования

алгоритма распределенной сортировки в модели «портфель задач» с применением технологии MPI на языке C++. Кластерная система проста для развертывания компонентов приложения сортировки на вычислительные узлы с использованием пакетной системы Torque. Кластер является системой наибольшей пропускной способности в составе вычислительной инфраструктуры организации. Положительный результат тестирования сортировки в модели «портфель задач» на кластере открывает возможность дальнейших исследований для другого типа оборудования в составе вычислительной сети организации.

### 3. Результаты экспериментального исследования

Проведенное экспериментальное исследование алгоритма сортировки подтвердило гипотезу о том, что в современных кластерных системах интенсивный обмен данными, являющийся следствием реализации алгоритма обработки данных в модели «портфель задач», не приводит к потере производительности.

В эксперименте выполнялась сортировка массива, состоящего из 8 блоков по 50 миллионов 8-ми байтных целых чисел в каждом блоке. При поблочной обработке выполнялось 8 вызовов алгоритма стандартной библиотеки C++ `std::sort` для сортировки и 28 вызовов `std::merge` для слияния блоков. Ярусно-параллельная форма используемого при слиянии блоков графа зависимостей имела 10 ярусов, максимум 4 параллельные задачи `std::merge` в одном ярусе. Программирование и управление вычислительными экспериментами на кластере выполнялось в системе TempletWeb [3].

Время последовательной сортировки на одном узле кластера (<http://hpc.ssau.ru/node/6>) алгоритмом `std::sort` стандартной библиотеки C++, примененным для всего массива, составило ~110,8 секунд. Время последовательной сортировки на одном узле кластера для программы, приведенной к модели «портфель задач», составило ~118,9 секунд. В параллельном варианте исполнения при развертывании управляющего и 8 рабочих процессов на отдельных узлах кластера время счета составило ~23,6 секунд (лучший результат). Это соответствует примерно пятикратному ускорению.

Проведена серия экспериментов, в которой исследовались различные варианты развёртывания программы на узлы кластерной системы. Было обнаружено, что несмотря на то, что используемые узлы кластера поддерживали минимум 8 аппаратных потоков, развёртывание всех рабочих процессов на 1 узел дает худший результат (~35,6 секунд) по сравнению с распределением процессов приложения по схеме 1 процесс на 1 узел кластера. Накладные расходы из-за конкуренции процессов за общую шину данных на узле оказались даже больше, чем накладные расходы на передачу данных между узлами.

### 4. Заключение

Проведенные эксперименты подтвердили пригодность модели «портфель задач» для вычислений на кластерных системах даже без применения алгоритмических приёмов оптимизации трафика между управляющим и рабочими процессами в задаче блочной сортировки большого массива данных.

Таким образом, модель управления распределенными вычислениями «портфель задач» позволяет ослабить технические требования к аппаратным средствам и может применяться для решения прикладных задач анализа и обработки больших информационных массивов в вычислительных сетях организации.

### 5. Литература

- [1] Ivashko, E.E. Enterprise Desktop Grids // CEUR Workshop Proceedings. – 2015. – Vol. 1502. – P. 16-21.
- [2] Sukhoroslov, O. Web-Based Platform for Publication and Distributed Execution of Computing Applications / O. Sukhoroslov, S. Volkov, A.A. Afanasiev // 14th International Symposium on Parallel and Distributed Computing (ISPDC) IEEE, 2015. – P. 175-184.

- [3] Vostokin, S.V. Templet Web: the use of volunteer computing approach in PaaS-style cloud / S.V. Vostokin, Y.S. Artamonov, D.A. Tsaryov // Open Engineering. – 2018. – Vol. 8(1). – P. 50-56.

## Using the bag-of-tasks model with centralized storage for distributed sorting of large data array

S.V. Vostokin<sup>1</sup>, I.V. Bobyleva<sup>1,2</sup>

<sup>1</sup>Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

<sup>2</sup>Joint Stock Company Space Rocket Center Progress, Zemetsa str. 18, Samara, Russia, 443009

**Abstract.** The article discusses the application of the bag of tasks programming model for the problem of sorting a large data array. The choice is determined by the generality of its algorithmic structure with various problems from the field of data analysis including correlation analysis, frequency analysis, and data indexation. The sorting algorithm is a block-by-block sorting, followed by the pairwise merging of the blocks. At the end of the sorting, the data in the blocks form an ordered sequence. The order of sorting and merging tasks is set by a static directed acyclic graph. The sorting algorithm is implemented using MPI library in C++ language with centralized storing of data blocks on the manager process. A feature of the implementation is the transfer of blocks between the manager and the worker MPI processes for each task. Experimental study confirmed the hypothesis that the intensive data exchange resulting from the centralized nature of the bag of task model does not lead to a loss of performance. The data processing model makes it possible to weaken the technical requirements for the software and hardware.