4. Sukhov A. M., Onoprienko A. V. Evaluating the effectiveness of geographic routing based on RIPE Atlas data [Text] //Telecommunications Forum Telfor (TELFOR), 2014 22nd. – IEEE, 2014. – P. 107-110.

5. Papadopoulos F. et al. Greedy forwarding in dynamic scale-free networks embedded in hyperbolic metric spaces [Text] //INFOCOM, 2010 Proceedings IEEE. – IEEE, 2010. – P. 1-9.

6. Sukhov A. M., Chemodanov D. Y. A metric for dynamic routing based on variational principles [Text] //Journal of High Speed Networks. – 2013. – Т. 19. – №. 2. – P. 155-163.

И.В. Казакова, С.Н. Попов, С.В. Востокин

# МИКРОСЕРВИСНОЕ ПРИЛОЖЕНИЕ ДЛЯ РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ НА ПРИМЕРЕ ЗАДАЧИ БЛОЧНОЙ СОРТИРОВКИ

(Самарский национальный исследовательский университет
имени академика С.П. Королёва)

При решении многих научных и технических задач возникает необходимость обработки больших массивов данных. Возможности последовательных приложений не позволяют эффективно использовать аппаратные ресурсы современных вычислительных систем, поэтому применение методов параллельного программирования вызывает активный интерес среди ученых и разработчиков программного обеспечения.

В работе на примере одной из актуальных задач распределенной обработки данных – блочной сортировки – рассматривается способ организации вычислений с применением акторной модели. Главными элементами модели являются акторы – активные агенты, выполняющие определенные действия согласно заданному сценарию [1]. Акторы были использованы для построения микросервисного приложения. Основным преимуществом данного стиля разработки является возможность такой организации приложения, при которой каждый из составляющих его сервисов выполняет часть задач, абстрагируясь от других, а взаимодействие между элементами происходит при помощи стандартных протоколов [2]. В результате нами разработано приложение, имеющее следующую структуру (рисунок 1).

Рассмотрим подробнее элементы структуры, представленной на рисунке 1, и схему их взаимодействия. Всю систему можно разделить на две подсистемы, а именно подсистему управления, включающую в себя акторы sorter (a), producer (b), merger (c), stopper (d), и подсистему выполнения, включающую акторы everest (f) и timer (g).

Предположим, что изначально есть массив данных, состоящий из M целых чисел, который необходимо отсортировать. Разделим его на N равных блоков. На каждый блок создается актор типа sorter (a). Sorter (a) взаимодействует с акторами producer (b) и everest (f).

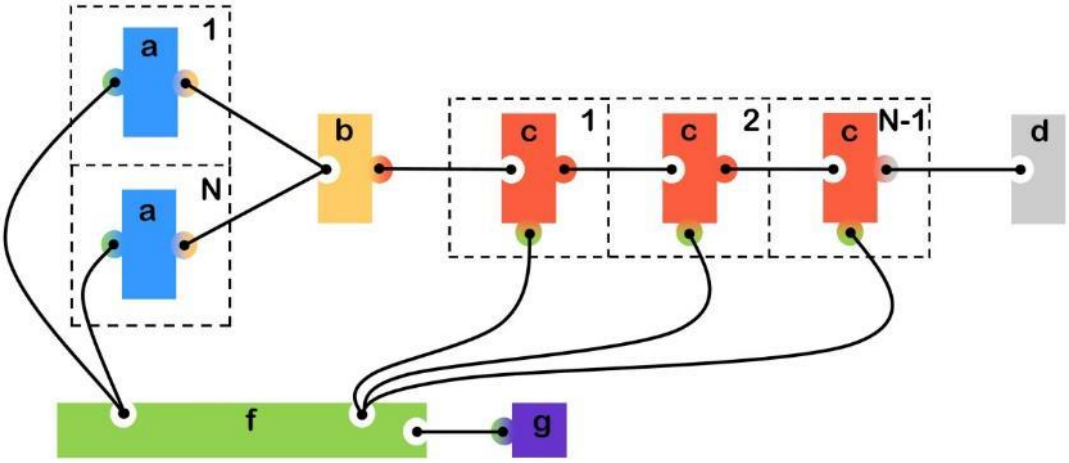


Рис. 9. Структура микросервисного приложения для распределенной обработки данных на примере задачи блочной сортировки

Важным элементом в этой системе является актор everest (f). В дальнейшем он будет взаимодействовать с сервисом Everest [3], который, в свою очередь, будет выполнять полученные задачи, используя имеющиеся аппаратные ресурсы, и возвращать результат. Актор everest (f) является прокси-сервером, основная функция которого состоит в том, чтобы согласовывать форматы запросов, полученных от акторов sorter (a) и merger (c), и ответы, полученные от сервиса Everest. На данный момент он является сервисом, который вызывает выполнение задач при помощи директивы *#pragma omp task* OpenMP 3.0, что позволяет сымитировать работу сервиса на многопроцессорной системе с общей памятью. Помимо акторов sorter (a) и merger (c), everest (f) взаимодействует с актором timer (g). Его основная задача заключается в том, чтобы производить периодический запуск актора everest (f). Если в процессе таких периодических запусков появляются задачи, которые находятся в очереди, то они отправляются на исполнение. Если есть исполненные задачи, то их результат возвращается отправителю.

После получения результата от everest (f), sorter (a) передает актору producer (b) сообщение о выполнении. Тот, в свою очередь, получив достаточное количество уведомлений, начинает передавать их по одному в цепочку акторов merger (c) в порядке возрастания номеров блоков. За счет взаимодействия акторов merger (c) и everest (f) происходит слияние блоков и их сортировка методом вставок, по завершению которых отправляется сообщение актору stopper (d), который останавливает вычисления.

Проведено экспериментальное исследование корректности работы и ускорения вычислений в разработанном микросервисном приложении (таблица 1).

Сортировка выполнялась на многоядерной системе с общей памятью с процессором Intel Core i7-3630QM 2.40 ГГц. Эксперимент по сортировке массива 128000000 чисел продемонстрировал ускорение в 2,76 раз на 4-х ядерной системе с поддержкой 8-ми аппаратных потоков. В дальнейшем планируется

проведение эксперимента блочной сортировки с реальным подключением к сервису Everest, используя сеть персональных рабочих станций в качестве исполнителя блочных операций сортировки и слияния.

Таблица 1 – Результаты экспериментов по блочной сортировке массива.

| Количество блоков | Размер блоков, млн | Последовательный алгоритм сортировки, с | Последовательный алгоритм блочной сортировки, с | Параллельный алгоритм блочной сортировки, с | Ускорение |
|---|---|---|---|---|---|
| 2 | 64 | 9,77 | 10,00 | 6,31 | 1,58 |
| 4 | 32 | 9,69 | 10,59 | 4,43 | 2,39 |
| 8 | 16 | 9,70 | 11,80 | 4,27 | 2,76 |
| 16 | 8 | 9,63 | 13,99 | 6,90 | 2,03 |
| 32 | 4 | 9,46 | 18,59 | 9,88 | 1,88 |
| 64 | 2 | 9,63 | 27,26 | 19,26 | 1,42 |
| 128 | 1 | 9,67 | 43,80 | 36,86 | 1,19 |

Таким образом, разработанное микросервисное приложение позволяет не только выполнять сортировку массива чисел на системе с общей памятью, но может применяться и для организации массивных вычислений на многопроцессорных системах с распределенной памятью. Предложенный подход построения приложений возможно также использовать для выполнения различных операций, требующих значительных аппаратных ресурсов, в компаниях, которые имеют обширную сеть рабочих станций. Приложения, создаваемые в соответствии с предложенной структурой, могут снизить затраты на организацию сетевой инфраструктуры за счёт более эффективного использования имеющихся рабочих станций.

**Литература**

1. Hewitt C. A universal modular ACTOR formalism for artificial intelligence // Proceedings of the 3rd IJCAI. SanFrancisco, CA, USA: Morgan Kaufmann Publishers Inc. – 1973. – P.235–45.

2. Microservices, available at https://martinfowler.com/articles/microservices/

3. Sukhoroslov O. A Web-Based Platform for Publication and Distributed Execution of Computing Applications / Sukhoroslov O., Volkov S., Afanasiev A. // IEEE Xplore. – 2015. – Vol. 14. – P. 175-184.