УДК 004.45

Б.А. Кумалаков

Казахский национальный университет имени аль-Фараби, Республика Казахстан, г. Алматы
E-mail: b.kumalakov@gmail.com

Об использовании интеллектуальной гибридной модели MapReduce-MPI для решения проблем ЧС с применением мобильных устройств

В статье представлены результаты вычислительного эксперимента, в ходе которого разработанная ранее MapReduce платформа для распределенных высокопроизводительных вычислений была впервые применена для решения задачи не относящейся к численным. В частности были смоделированы условия чрезвычайной ситуации, спроектирован и формализован алгоритм решения, разработана реализация программного обеспечения с применением многоагентных-технологий и мобильных устройств. Мобильными устройствами в данном случае являются смартфоны и планшетные компьютеры имеющие модуль доступа к беспроводным сетям и обладающие функционалом определения координат собственного положения. Ядром данного приложения является функционал, способный использовать индивидуальное программное обеспечение отдельно взятых вычислительных узлов для решения общих задач, а именно возможность использования геолокационных систем отдельных устройств за счет вызова компонентов мобильных систем. Полученные таким образом данные далее используются для расчета оптимальных точек эвакуации, формирования и интерактивной преподачи маршрута до определенных точек эвакуации каждому устройству, а так же расчета оптимального пути следования транспорта эвакуации служб ЧС. К полученным новшествам так же относятся модификация алгоритма кластеризации к-вершин, а так же эмпирические данные о работе платформы при решении задач не относящихся к классу численных.
Ключевые слова: многоагентные системы, мобильные устройства, задачи ЧС.

B.A. Kumalakov

About using intellectual hybrid MapReduce-MPI model to solve disaster problems using mobile devices

This article presents a case study on applying novel MapReduce platform to solve a disaster management assistance problem. It was initially designed to solve numeric problems in a distributed fashion, making use of parallel computing capabilities of individual nodes. In this case we apply the platform as a distributed computing environment that collects user location data, computes optimal evacuation points, guides mobile device users to them and, finally, computes the optimal pick up route. Throughout this process it makes extensive use of vendor provided geolocation, wireless networking and peer-to-peer communication capabilities. Presented research novelties include advanced clustering k-means algorithm modification and real data on the working extension of the previously designed framework. Conducted studies also prove that agent self-organization phenomena is not only helpful when organizing infrastructure level workload distribution, but is efficiently used and has positive impact on application performance as individual device (computing node) capabilities are used as building block to achieve general system goal. The problem solved proves the platform to be of relevance to multi-functional needs and demonstrates its ability to incorporate independent (device specific) functions of the infrastructure nodes. Moreover, it is proven for the first time that the platform can be treated a multi-functional (general propose) programming environment, given that application algorithm requires distributed and parallel computing efforts.
Key words: multi-agent systems, mobile devices, disaster management.

Б.А. Кумалаков

MapReduce-MPI интеллектуалды гибридті моделін пайдаланып апатты жағдай есептерін шешу үшін қолдану

Бұл мақалада үлестірілген параллельді есептеуге арналған MapReduce бағдаламалау платформасының көмегімен сандық есептер қатарына жатпайтын, апатты жағдайда адамдарды құтқару үшін пайдалынатын бағдарламалық кешенді алдымен жобалап, содан соң құрастырып тестілеу нәтиделері ұсынылған. Атап айтатын болсақ, бағдарламалық қосымша пайдаланушылардың мобильді құрылғыларын пайдаланып координаталарын анықтайды да, оларды эвакуациялау үшін оптималды жерді санап табады. Содан соң, әр мобильді құрылғы өзіне жақын жерге апаратын жол тауып, пайдаланушыға шығарады да, жүйе анықталған жерлерді аралап өту жолын санайды. Зерттеу барысында келесідей жаңалықтарға қол жеткізілді: к-төбелер алгоритімі апатты жағдайда адамдарды құтқару үшін қажетті функционалға бейімделген; санды әдістерді қолданып есептерді шешу үшін құрастырылған платформа мобильді құрылғыларды пайдаланып жеке-дара түйн ресурстарын (ішкі бағдарламаларын) бірбірірмен интеграциялау арқылы ортақ есепті шешу; және осы құрастырылған бағдарламалық кешеннің жұмыс істеу статистикасы жиналды. Соның ішінде түйіндердің жеке ресурстарын интеграциялау көп-агентті жүйелердің тек қана төменгі архитектура деңгейінде ғана емес, бағдарламалық деңгейде де агенттердің өз-өздерін ұйымдастыра алу қабілетін дәлелдейді. Яғни, алғашқы құрастырылған плаформаның қасеттері оған арналып жазылған қосымшаларға да тән болады.

Түйін сөздер: көп-агентті жүйелер, мобильді құрылғылар, апатты жағдай.

Введение

В источниках [1-3] авторы предлагают модели распределения вычислительной нагрузки, позволяющие наиболее полно использовать ресурсы высокопроизводительных кластеров. Особенностью данных походов является использование инструментов виртуализации для интеграции разнородных не гетерогенных ресурсов для организации распределенных вычислений. Однако, несмотря на инновационную новизну, исследования, описанные в [4, 5], указывают на ряд факторов и свойств, снижающих уровень надежности вычислений в рамках предлагаемых моделей. Исходя из этого были разработаны решения [6-8]. В них авторы обосновывают и затем экспериментально доказывают преимущества своих облачных решений для параллельных и в тоже время распределенных платформ. Однако, их основным недостатком является сложный механизм управления вычислениями, который контролирует каждый шаг заданных операций. С точки зрения разработки программного обеспечения это означает то, что разработчики должны детально понимать механизмы управления и закладывать их характеристики в код программ. Потенциально данная проблема влечет наличие ошибок в коде, которые появляются в следствие не знания каких-либо элементов платформы.

В качестве альтернативы в [9] мы предложили новую архитектуру для композитных облачных вычислений направленных на решение больших численных задач. Её ядром является многоагентная система, способная перенастроиться динамический при падении узлов, тем самым обеспечивая высокий уровень отказоустойчивости вычислений. Затем интеллектуальная составляющая платформы была расширена за счёт применения алгоритмов обучения, на базе которых из различных соединений вычислительных машин формируется композитное облако [10]. При этом в ходе выполнения самих вычислений и формирования композитных облаков полностью соблюдаются принципы автономности агентов и сохраняются свойства сложных систем.

Рисунок 1– Вычислительные зоны в композитной облачной среде

В отличие от известных аналогов в нашей работе распределение нагрузки основывается на использовании феномена самоорганизации агентов, что в свою очередь ведет к более высокой степени отказоустойчивости всей системы. Концептуально организация вычислений изображена на рисунке 1. Рисунок изображает кластерную среду, основанную на неоднородных ресурсах (в том числе мобильные, персональные и серверные устройства). Внутри данной среды агенты самоорганизовываются в вычислительные зоны. Вычислительной зоной назовем множество агентов, задействованных в решении задачи, запущенной в определенный момент времени. Как только агент перестает выполнять часть задачи он выбывает из вычислительной зоны и наоборот. Таким образом ресурсы композитного облака динамически высвобождаются по мере выполнения отдельных шагов задач и могут примыкать к тем вычислительным зонам, которые нуждаются в ресурсах не ожидая полного окончания предыдущей коллективной работы.

Важно отметить, что кластер представляет собой физическую архитектуру, в то время как композитное облако является результатом взаимодействия агентов и потому должно пониматься как логическая организация системы распределения нагрузки на вычислительные узлы. Иными словами, вычислительный кластер (как множество машин) может представлять собой как множество вычислительных узлов, так и один мощный узел в зависимости от онтологии, принятой тем или иным сообществом агентов. Таким образом обеспечивается коллективное использование всех ресурсов по мере необходимости и той форме, которая необходима в момент вычислений.

Наконец в [11] вычислительные узлы композитной облачной платформы получили возможность задействовать многоядерный потенциал каждого вычислительного узла в полной мере посредствам интеграции технологии MPI в модель вычислений. Таким образом была сформирована вычислительная платформа, основанная на принципах функционирования сложных систем, способная:

1. динамический само-реконфигурироваться во время выполнения задач при возникновении сбоев на вычислительных узлах;

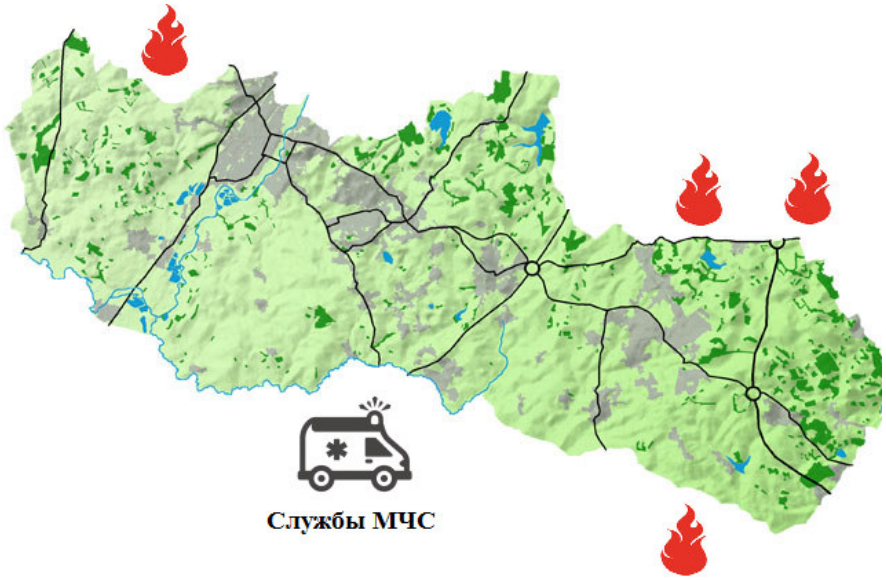

Рисунок 2– Визуальное представление участка леса с обозначением очагов пожара

2. динамически наращивать и высвобождать ресурсы отдельных процессов;
3. формировать композитные облачные структуры в случае острой нехватки ресурсов в пределах одного объединения вычислительных машин;
4. наиболее полно использовать ресурсы каждого вычислительного узла в составе архитектуры.

Однако, в ходе проведенных исследований функциональность платформы проверялась лишь на примерах простейших численных задач. Потому не представляете возможным утверждать, что выявленные качества платформы справедливы при решении прикладных задач, в том числе не численного характера.

В данной статье представлена попытка применить разработанное технологическое решение для решения задачи оперативного планирования эвакуации людей из леса, в котором бушует пожар.

Постановка задачи

Определим задачу следующим образом: в следствие засухи на участке леса развились несколько очагов пожара (Рисунок 2).

Группа служб реагирования министерства по чрезвычайным ситуациям прибывает на место и перед ней стоит ряд задач, среди которых и эвакуация лиц, находящихся в районе пожара. Предположим, что люди гуляют группами или в одиночку, и у каждой группы (или лица если он один) есть при себе мобильное устройство. Однозначно то, что в реальной ситуации некоторые туристы оставляют мобильные устройства дома или в них мог сесть заряд, однако в рамках данной задачи мы рассматриваем тот случай, когда мобильные устройства пользователей используются для непосредственных вычислений. Все лица, не попавшие в множество описанных туристов будут разыскиваться силами МЧС без использования предлагаемой технологии.

По факту прибытия на место, службы МЧС развертывают Wi-Fi сеть при помощи специализированного оборудования и вещают предупреждение с инструкциями по подключению мобильных устройств к ней. Когда мобильные устройства подключаются их координаты вычисляются с помощью системы глобального позиционирования (GPS) и передаются службам МЧС. Далее компьютерная система должна рассчитать оптимальную точку сбора туристов, и вычислить наиболее короткий путь сбора для автобуса МЧС.

Алгоритм решения задачи

Алгоритм 1 формализует последовательность действий по программной поддержке работы служб ЧС в определенных ранее условиях.

initialization;
read local map;
read mobile device coordinates;
define grouping centers;
while *grouping condition is not met* **do**
| group mobile devices by geographical location;
end
compute local gathering pathes;
compute general evacuation path;

Алгоритм 1– Формализованный алгоритм программного продукта поддержки работы бригад эвакуации

Из алгоритма видно, что точка сбора туристов рассчитывается путем кластеризации. В данном случае кластеризация проводится таким образом, что после нахождения центра кластера (эталонного объекта) происходит сдвиг. Данный сдвиг выполняется в направлении наиближайшего возможного места эвакуации. Таким образом, алгоритм сперва группирует туристов по географической отделенности друг от друга, а затем назначает точкой сбора ближайший пункт подходящий для эвакуации. Множество таких пунктов определяется в виде логических выражении экспертом МЧС. Алгоритм кластеризации формализован и представлен в виде алгоритма 2.

Для определения наикратчайшего пути эвакуации использован метод ветвей и границ. При этом в качестве вершин графа использованы пересечения дорог и точки эвакуации, определённые экспертом, а в качестве ребер использованы дороги соединяющие названные объекты.

Вычислительный эксперимент

Для организации вычислительного эксперимента были написаны и протестированы с дальнейшим сравнением результатов последовательная программа и приложение для интеллектуальной платформы. Для реализации алгоритма кластеризации в обоих приложениях за базовую модель была выбрана библиотека «javaml-0.1.5», включающая стандартную реализацию алгоритма к-вершин. С целью ее дальнейшей модификации для нужд приложения автором была дописана логика расчета и присвоения значений

initialization;
read local map and evacuation points R;
read coordinates of mobile devices S;
define class centers Z;
while *MaxIteration is not reached and Z change* **do**
  for every $s \in S$ compute $\rho^*(s_i, Z_j) = min\{\rho(s_i, Z_1), \rho(s_i, Z_2), ...\rho(s_i, Z_n)\}$;
  update Z_k coordinates as $X_k = (x_{z_k} + x_{s_m})/2, Y_k = (y_{z_k} + y_{s_m})/2, Z_k = (z_{z_k} + z_{s_m})/2$
end
compute computer $\rho^*(Z_j, R_t)$ and assign $Z_j = R_t$;

Алгоритм 2– Алгоритм кластеризации для задачи поддержки работы бригад эвакуации
$\rho(Z_j, R_t)$ – функция, возвращающая Евклидово расстояние между заданными объектами

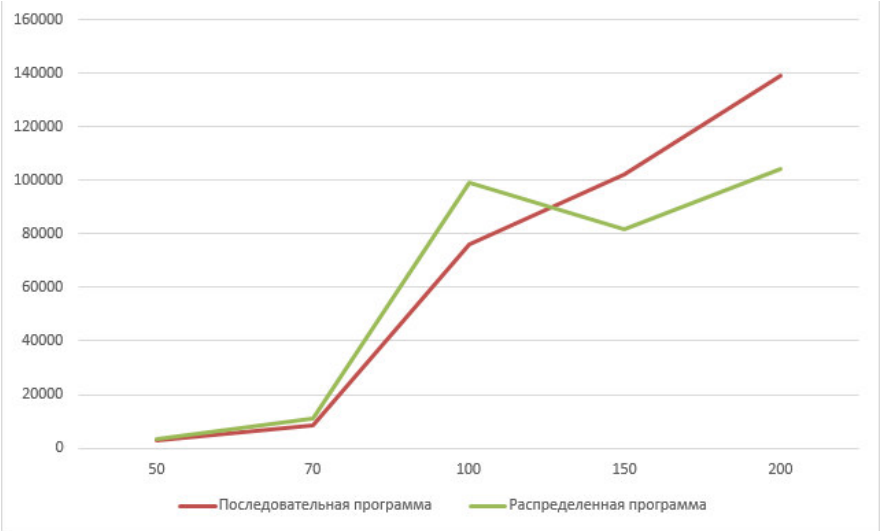

Рисунок 3– Результаты тестирования приложений

точек эвакуации на языке Java. Сама карта местности и потенциальные точки эвакуации, также, как и их координаты сгенерированы автоматический и сохранены в текстовый файл. Последовательное приложение тестировалось на одном компьютере со следующими характеристиками: процессор Intel core i7, ОЗУ 8 Гб. Для моделирования передачи данных по сети промежуточные результаты вычислений сохраняются в текстовый файл и снова считываются при кластеризации и затем расчете. Это позволяет учесть возможные задержки при реальной работе приложения. В параллельной версии на интеллектуальной платформе результаты промежуточных вычислений передаются напрямую в память устройств так, как задействованы два мобильных телефона и персональный компьютер. На рисунке 3 изображены результаты тестирования приложений. Как и ожидалось распределенная версия программы показывает менее высокую производительность на входных данных меньше 150 объектов. Однако, при росте объема данных возрастает выгодность использования разработанной платформы.

Примечательным в данном результате является то, что в отличие от предыдущих случаев был рассмотрен пример применения данной платформы не для решения боль-

ших численных задач, а для решения задач при помощи использования встроенных функций мобильных телефонов. Таким образом продемонстрированы дополнительные возможности, свойственные мобильным устройствам.

References

[1] *Becker J., Dagum L.* Particle simulation on heterogeneous distributed supercomputers // Concurrency-Practice and experience. – 1993. – №5(4). – С. 367–377.

[2] *Fougere D., Malyshkin V.* NumGrid middleware: MPI support for computational grids // Lecture Notes in Computer Science (PACT 2005). – 2005. – V.3606. – P. 313–320.

[3] *Diaz J., Munoz-Caro C., Nino A.A.* Survey of Parallel Programming Models and Tools in the Multi and Many-Core Era // IEEE Transactions on parallel and distributed systems. – 2012. – V.23(8). – P. 1369–1386.

[4] *Cappello F., Djilali S., Fedak G.* Computing on large-scale distributed systems: XtremWeb architecture, programming models, security, tests and convergence with grid // Future generation computer systems. – 2005. – V. 21(3). – P. 417–437.

[5] *Wang J., Liu Z.* Parallel Data Mining Optimal Algorithm of Virtual Cluster // International Conference on Fuzzy Systems and Knowledge. –2008. – V.5. – P. 358–362.

[6] *Pandey S., Buyya R.* Scheduling Workflow Applications Based on Multi-source Parallel Data Retrieval in Distributed Computing Networks // Computer journal. – 2012. – V. 55(11). – P. 1288–1308.

[7] *Liu H., Orban D.* GridBatch: Cloud Computing for Large-Scale Data-Intensive Batch Applications // CCGRID. – 2008. – V. 1. – P.295–305.

[8] *Valilai O., Houshmand M.* A collaborative and integrated platform to support distributed manufacturing system using a service-oriented approach based on cloud computing paradigm // Robotics and computer-integrated manufacturing. –2013. –V.29(1). – P.110–127.

[9] *Ahmed-Zaki D., Dobrowolski G., Kumalakov B.* Peer-to-Peer MapReduce Platform // Proceedings of the 5th International Conference on Agents and Arti?cial Intelligence. (ICAART 2013). – 2013. – V. 2. – P. 565–570.

[10] *Akhmed-Zaki D., Kumalakov B.* Composite Peer-to-Peer MapReduce System // Proceedings of the International Conference on New Trends in Information and Communication Technologies (ICTT 2013). – 2013. – P.44—49.

[11] *Akhmed-Zaki D.Zh., Kumalakov B.A.* Solving complex iterative tasks using intellectual load distribution and MPI // Вестник Национальной инженерной академии Республики Казахстан. – 2014. – №2(52). –P. 25–31.