N E W S
OF THE NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
PHYSICO-MATHEMATICAL SERIES
ISSN 1991-346X
Volume 2, Number 312 (2017), 167 – 172

UDC 004.421.2.519.178

A.S. Shomanov[1], D.Zh. Akhmed-Zaki[1], E.N. Amirgaliyev[1], M.E. Mansurova[1]

[1]al-Farabi Kazakh National University, Almaty, Kazakhstan.
e-mail: adai.shomanov@gmail.com

About the problem of key distribution in Mapreduce model

Abstract. An important result in the field of processing of large amounts of data is the invention of the Mapreduce model. This model is based on the separation of data processing between parallel running processes. Data processing is performed using two types of functions: map and reduce. Map defines the transformation of input text by parallel processes on the basis of which a set of key / value pairs is generated. Reduce is operate on specific key and a list of all values associated with this key by performing a set of operations on this list of values, result of which is a pair with a key value and a certain aggregate value obtained as a result of these operations. The working environment of Mapreduce is a cluster consisting of a set of compute nodes. The nodes in the cluster must be connected by the communication network. Each node is scheduled to launch map and reduce processes.

Thus, an important task is to plan the distribution of keys among reduce processes in order to minimize data transfer operations over the network. Importance is justified by the fact that data transfer operations over the network greatly increase the processing time in case of incorrectly planned key distribution process. This task is NP-complete. The NP-completeness of the problem limits its exact solution, even for small parameters of the input data. For its effective solution in the current article, an approach based on the genetic algorithm is proposed. This approach can not guarantee an exact solution, but gives rather good approximated results. Another advantage of this approach is the ability to solve problems with large parameters of input data.

Keywords: Mapreduce, optimization, parallel processing, Generalized assignment problem.

УДК 004.421.2.519.178

А.С. Шоманов[1], Д.Ж. Ахмед-Заки[1], Е.Н. Амиргалиев[1], М.Е. Мансурова[1]

[1]Казахский Национальный Университет им. аль-Фараби

О ЗАДАЧЕ ОПТИМИЗАЦИИ РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ В MAPREDUCE МОДЕЛИ

Аннотация. Важным результатом в области обработки больших объемов данных является изобретение Mapreduce модели. В основе данной модели лежит разделение обработки данных между параллельными процессами. Обработка данных производится с использованием двух типов функций: map и reduce. Map задает преобразование входного текста параллельными процессами на основе которого генерируется набор пар ключ/значение. Reduce на основе определенного ключа и списка всех значений, связанных с этим ключом, производит набор операций над списком значений, результатом которого является пара со значением ключа и определенного агрегированного значения, полученного как результат этих операций. Средой работы Mapreduce является кластер, состоящий из набора вычислительных узлов. Узлы в кластере должны быть соединены коммуникационной сетью. На каждом узле производится запуска процессов map и reduce.

Таким образом, важной задачей является планирование распределения ключей по процессам reduce с целью минимизации операций передачи данных по сети. Важность обоснована тем, что

операции передачи данных по сети в значительной степени увеличивают время обработки данных в случае неправильно спланированного процесса распределения ключей. Данная задача является NP-полной. NP-полнота задачи ограничивает ее точное решение даже для малых параметров входных данных. Для ее эффективного решения в текущей статье предлагается подход на основе генетического алгоритма. Данный подход не может гарантировать точного решения, но дает достаточно хорошие аппроксимированные результаты. Другим преимуществом данного подхода является возможность решения задач с большими параметрами входных данных.

Ключевые слова: Mapreduce, оптимизация, параллельная обработка данных, Обобщенная задача о назначениях.

В основе технологии Mapreduce [1-5] лежит идея параллельной обработки данных с применением двух основных функций: map и reduce. Функция map позволяет на основе входного набора данных произвести преобразование, в котором формируются в результате пары ключ/значение. Далее происходит группировка полученных пар ключ/значение, так что каждая преобразование reduce оперирует только на множестве значений, имеющих один и тот же ключ. Для параллелизации процесса обработки на основе Mapreduce, данные разбиваются между параллельно работающими map процессами на этапе map, а затем на этапе reduce создаются параллельные процессы для выполнения функции reduce по каждому ключу. Таким образом, при обработке больших массивов данных достигается высокий уровень масштабирования. Основной цикл работы Mapreduce состоит из 4 этапов :

- **Init**. Задается описание функций map и reduce, входные и выходные директории и другие параметры.
- **Map.** Каждый процесс map сканирует данные, переданные ему в качестве входного параметра. В ходе обработки данных функцией map формируется список пар ключ/значение согласно функции map, заданной пользователем.
- **Shuffle.** Происходит распределение пар ключ/значение по reduce процессам таким образом, что каждый reduce процесс обрабатывает только один, предназначенный для него, уникальный ключ.
- **Reduce.** Каждый reduce процесс выполняет операции на наборе пар ключ/значение согласно функции reduce, заданной пользователем.

Одной из основных проблем, связанных с обработкой больших объемов данных на основе технологии Mapreduce является оптимизация задачи по назначению ключей различным параллельным reduce процессам. Задача оптимизации включает в себя балансировку нагрузки между этими процессами, сокращения объема передаваемых данных, а также уменьшения количества операций передачи и обмена данными по сетевым каналам.

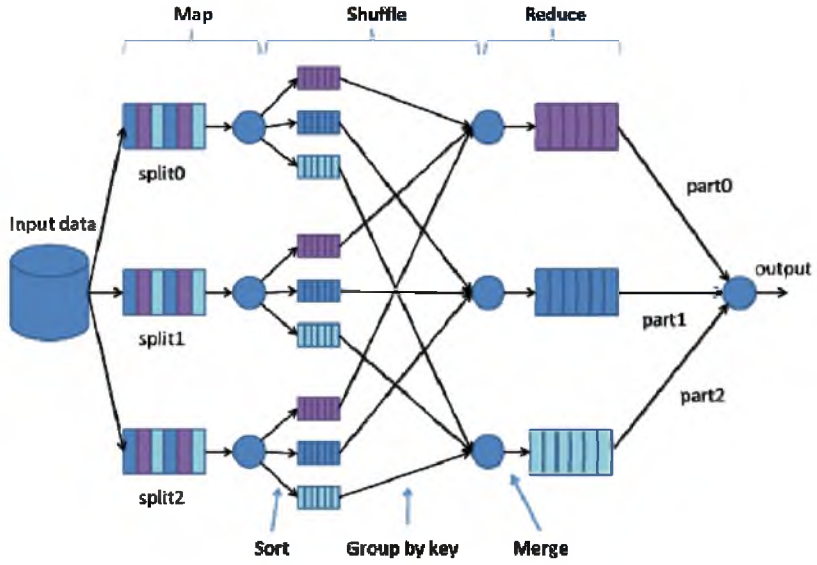

Рисунок 1 - Принцип работы Mapreduce

Из рисунка 1 видно, что данные с одинаковыми ключами на этапе распределения ключей (Shuffle) передаются одному процессу, ответственному за обработку определенного ключа. Так как объем данных может быть огромным, то соответственно, важно правильно определить, какому процессу нужно назначить определенный ключ для минимизации операций передачи данных по сети.

Для задачи распределения вычислительной нагрузки между параллельно работающими процессами и оптимального балансирования объема данных нами был применен эвристический подход на основе генетического алгоритма. Генетические алгоритмы [6-9] используются для решения многих задач в области комбинаторной и многокритериальной оптимизации. Основную часть задач в данных областях составляют NP-полные задачи, для которых не существует полиномиального алгоритма для их решения. Таким образом, многие важные задачи из данных областей остаются без эффективных подходов, которые способны были бы за разумное время решить данную задачу даже для самых малых параметров входных данных. Таким образом, единственными методами способными дать более или менее подходящее решение для больших размерностей данных задач являются различные классы эвристических алгоритмов.

Формулировка задачи оптимизации состоит из следующего множества условий:

$$min \sum_{i=0}^{threads-1} \sum_{j=1}^{keys} x_{ij} \times cost_{ij} \quad (1)$$

$$x_{ij} \in \{0,1\} \quad (1)$$

$$min\left(\max_{i,j=0..threads-1} |load_i - load_j|\right) \quad (2)$$

$$load_i = \sum_{t=0}^{threads-1} \sum_{j=1}^{keys} x_{ij} \times size_{tj} \quad (3)$$

Задача оптимизации, приведенная выше, является модификацией “Обобщенной задачи о назначениях” [10-13]. Данная задача относится к классу NP-полных задач. Различные генетические алгоритмы для решения данной задачи были описаны ранее [14,15].

Для того чтобы найти стоимость назначения ключа j потоку i нам требуется создать матрицу, в которой каждому элементу данной матрицы $cost_{ij}$ присваивается значение стоимости назначения ключа j потоку i. Данная стоимость задается на основе количества элементов определенного ключа, которые требуется передать определенному потоку. В формуле (3) дается определение функционала балансировки нагрузки. Функционал балансировки нагрузки рассчитывается как минимальное значение максимальной разницы общей нагрузки $load$ между любыми парами различных потоков. Значение общей нагрузки $load$ для каждого потока рассчитывается согласно формуле (4). Формула (2) описывает область определения для переменной x_{ij}. Значение x_{ij} задается равным 1, если поток i был назначен для обработки ключа под номером j, в противном случае, данное значение равно 0 и, соответственно, поток i не был назначен для обработки ключа j. Для модифицированной “Обобщенной задачи о назначениях” применимо к процедуре распределения ключей в Mapreduce модели, требуется оптимизировать значение обоих функционалов для получения приемлемого решения.

Для того чтобы применить к вышеописанной задаче подход с применением генетического алгоритма нужно определить как можно представить задачу на языке генетического алгоритма. Решение (особь) для задачи можно представить в виде вектора значений, где -ому элементу данного вектора присваивается номер потока назначенного для обработки i-ого ключа. Популяция

представляет собой множество особей, выбранных согласно методам отбора генетического алгоритма. Размер популяции задается, как отдельный параметр алгоритма и может быть изменен в зависимости от специфических свойств решаемой задачи. Хромосома является упорядоченной последовательностью генов. Ген представляет собой атомарный элемент хромосомы. Начальная популяция выбирается случайной генерацией значений генов для хромосомы каждой особи. Функция приспособленности (фитнес-функция) определяет меру приспособленности определенной особи. Для нашей задачи оптимизации фитнес-функция рассчитывается как взвешенная сумма значений функционалов (1) и (3) для каждой особи в популяции. Задача генетического алгоритма, таким образом, выражается в нахождении особи с лучшим значением фитнес-функции. Генетический алгоритм для задачи распределения ключей в Mapreduce модели работает согласно следующей процедуре:

```
Load balancing procedure
Initialize algorithm
Generate initial random population p of chromosones
While i <= MAX_ITERATIONS
Update fitness values of each element of p
For i=1..P_NUM
Choose two parent elements p1 and p2 from current population p by applying tournament selection
Perform crossover on p1 and p2 to generate child chromosone c
Perform mutation on child chromosone c
Add child chromosone c to the list of new population elements np
End Of For
Assign current population p to newly obtained population list np
End of While
Choose list member with best fitness value and assign it to array sol
```

Рисунок 2 - Генетический алгоритм распределения ключей в Mapreduce модели

Основной цикл алгоритма состоит из P_NUM шагов. На каждой итерации алгоритма сначала обновляются значения фитнес-функции для каждой особи в текущей популяции. Затем, выбираются две особи популяции p_1 и p_2 на основе турнирного отбора, цель которого заключается в том, чтобы выбрать из случайно выбранного множества особей из текущей популяции наиболее приспособленного (с наилучшим значением фитнес-функции). После этого, производится операция кроссинговер (crossover), и из двух особей, посредством данной операции, генерируется новая особь потомок, которая наследует свойства обоих родительских особей. После этого новая особь потомок добавляется в новую популяцию. Затем, производится операция мутации над новой особью потомком, после которой основной цикл алгоритма завершается и происходит переход на новую итерацию.

ЛИТЕРАТУРЫ

[1] Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM, 51*(1), 107-113. doi:10.1145/1327452.1327492

[2] Dean, J., & Ghemawat, S. (2010). Map reduce: A flexible data processing tool. *Communications of the ACM, 53*(1), 72-77. doi:10.1145/1629175.1629198

[3] Lee, K. -., Lee, Y. -., Choi, H., Chung, Y. D., & Moon, B. (2011). Parallel data processing with MapReduce: A survey. *SIGMOD Record, 40*(4), 11-20. doi:10.1145/2094114.2094118

[4] Jiang, D., Ooi, B. C., Shi, L., & Wu, S. (2010). The performance of mapreduce: An indepth study. *Proceedings of the VLDB Endowment, 3*(1), 472-483.doi: 10.14778/1920841.1920903

[5] Dean, J. (2006). Experiences with MapReduce, an abstraction for large-scale computation. Paper presented at the *Parallel Architectures and Compilation Techniques - Conference Proceedings, PACT, , 2006* 1. doi:10.1145/1152154.1152155

[6] Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: A survey. *Computer, 27*(6), 17-26. doi:10.1109/2.294849

[7] Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety, 91*(9), 992-1007. doi:10.1016/j.ress.2005.11.018

[8] Storn, R., & Price, K. (1997). Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341-359.doi:10.1023/A:1008202821328

[9] Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization, 26*(6), 369-395. doi:10.1007/s00158-003-0368-6

[10] Gavish, B., & Pirkul, H. (1991). Algorithms for the Multi-Resource Generalized Assignment Problem. *Management Science, 37*(6), 695-713.doi: 10.1287/mnsc.37.6.695

[11] Shmoys, D. B., & Tardos, E. (1993). Approximation algorithm for the generalized assignment problem. *Mathematical Programming, Series B, 62*(3-8), 461-474.doi:10.1007/BF01585178

[12] Ross, G. T., & Soland, R. M. (1975). A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming, 8*(1), 91-103. doi:10.1007/BF01580430

[13] Cattrysse, D. G., & Van Wassenhove, L. N. (1992). A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research, 60*(3), 260-272. doi:10.1016/0377-2217(92)90077-M

[14] Chu, P.C. & Beasley, J.E. 1997, "A genetic algorithm for the generalised assignment problem", *Computers and Operations Research,* vol. 24, no. 1, pp. 17-23.doi: 10.1016/S0305-0548(96)00032-9

[15] Liu, Y. Y., & Wang, S. (2015). A scalable parallel genetic algorithm for the generalized assignment problem. *Parallel Computing, 46*, 98-119. doi:10.1016/j.parco.2014.04.008

REFERENCES

[1] Dean, J., & Ghemawat, S. **(2008**). MapReduce: Simplified data processing on large clusters. *Communications of the ACM, 51*(1), 107-113. doi:10.1145/1327452.1327492

[2] Dean, J., & Ghemawat, S. (**2010)**. Map reduce: A flexible data processing tool. *Communications of the ACM, 53*(1), 72-77. doi:10.1145/1629175.1629198

[3] Lee, K. -., Lee, Y. -., Choi, H., Chung, Y. D., & Moon, B. (**2011)**. Parallel data processing with MapReduce: A survey. *SIGMOD Record, 40*(4), 11-20. doi:10.1145/2094114.2094118

[4] Jiang, D., Ooi, B. C., Shi, L., & Wu, S. (**2010).** The performance of mapreduce: An indepth study. *Proceedings of the VLDB Endowment, 3*(1), 472-483.doi: 10.14778/1920841.1920903

[5] Dean, J. (**2006)**. Experiences with MapReduce, an abstraction for large-scale computation. Paper presented at the *Parallel Architectures and Compilation Techniques - Conference Proceedings, PACT, 2006* 1. doi:10.1145/1152154.1152155

[6] Srinivas, M., & Patnaik, L. M. (**1994**). Genetic algorithms: A survey. *Computer, 27*(6), 17-26. doi:10.1109/2.294849

[7] Konak, A., Coit, D. W., & Smith, A. E. (**2006)**. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety, 91*(9), 992-1007. doi:10.1016/j.ress.2005.11.018

[8] Storn, R., & Price, K. (**1997**). Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11*(4), 341-359.doi:10.1023/A:1008202821328

[9] Marler, R. T., & Arora, J. S. (**2004**). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization, 26*(6), 369-395. doi:10.1007/s00158-003-0368-6

[10] Gavish, B., & Pirkul, H. (**1991)**. Algorithms for the Multi-Resource Generalized Assignment Problem. *Management Science, 37*(6), 695-713.doi: 10.1287/mnsc.37.6.695

[11] Shmoys, D. B., & Tardos, E. (**1993)**. Approximation algorithm for the generalized assignment problem. *Mathematical Programming, Series B, 62*(3-8), 461-474.doi:10.1007/BF01585178

[12] Ross, G. T., & Soland, R. M. (**1975**). A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming, 8*(1), 91-103. doi:10.1007/BF01580430

[13] Cattrysse, D. G., & Van Wassenhove, L. N. (**1992)**. A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research, 60*(3), 260-272. doi:10.1016/0377-2217(92)90077-M

[14] Chu, P.C. & Beasley, J.E. **1997**, "A genetic algorithm for the generalised assignment problem", *Computers and Operations Research,* vol. 24, no. 1, pp. 17-23.doi: 10.1016/S0305-0548(96)00032-9

[15] Liu, Y. Y., & Wang, S. (**2015**). A scalable parallel genetic algorithm for the generalized assignment problem. *Parallel Computing, 46*, 98-119. doi:10.1016/j.parco.2014.04.008

А.С. Шоманов[1], Д.Ж. Ахмед-Заки[1], Е.Н. Амиргалиев[1], М.Е. Мансурова [1]

[1]әл-Фараби атындағы Қазақ Ұлттық Университеті, Алматы, Қазақстан Республикасы

КІЛТТЕРДІ MAPREDUCE ҮЛГІСІНДЕ ТАРАТУ ЕСЕБІ ТУРАЛЫ

Үлкен деректерді өңдеу саласында Mapreduce үлгіні ойлап шығаруы маңызды нәтиже деп есептелінеді. Осы үлгінін негізінде деректерді өңдеу тапсырмасын параллельді процесстер арасында бөлуі жатыр. Деректерді өңдеу тапсырмасы екі функция арқылы орындалады: map және reduce. Map кіріс мәтінді параллельді процесстермен түрлендіру тапсырмасын орындайды, нәтижесінде кілт/мағынасы жиынтығы құрастырылады. Белгілі кілт және осы кілтпен байланысты мағыналар жиынтығы негізінде reduce мағыналар жиынтығымен әртүрлі операцияларды орындайды, нәтижесі ретінде кілт және операциялардан алынған жинақ мағынасы боп табылады. Mapreduce жұмыс істеу ортасы бірнеше есептеу орталықтардан құрылған кластерден тұрады. Кластердін орталықтары коммуникациялық желілермен байланыстырылған болу керек. Әр есептеу орталығында map және reduce процесстердін іске косыуы орындалады.

Сонымен, желілер бойынша деректерді жіберу азайту мақсатымен кілтерді reduce процесстер ішінде тарату есебі мыңызды боп есептелінеді. Кілттерді тарату тапсырма дурыс емес жоспарылған кезде деректерді желілер арқылы жіберу мағыналы дәрежесінде деректерді өңдеу уақытын көбейтеді. Осы есеп NP-толық есептеріне жатады. NP-толықтығы түгіл ең кіші деректердін кіріс параметрлердін жағдайында осы есептін нақты шешімін табуын шектейді. Осы есепті тиімді түрде шешу үшін осы мақалада ғенетикалық алғоритм негізінде тәсілдеме ұсынылады. Осы тәсілдеме нақты шешімін жеткізе алмай мүмкін, бірақ өте жақсы жуықтама нәтижесін көрсете алады. Осы тәсілдеменін үлкен параметрлермен кіріс деректерден құрылған есептерді шешуі басқа артықшылық боп есептелінеді.

Тірек сөздер: Mapreduce, үйлесімділеу, деректерді параллельді өңдеу, Жалпылама тағайындау есебі.