

Распределенный метод хранения больших данных в облаках с использованием системы остаточных классов

Е. А. Кучукова¹, С. Аль-Гальда², Н. Г. Гудиева³, Л. В. Андрухив³, А. Е. Шаньгина⁴, Е. Р. Абдулина³

Северо-Кавказский федеральный университет
Ставрополь

¹e.cat.kuchukova@gmail.com, ²yzorsv@gmail.com, ³mgbabenko@ncfu.ru, ⁴anastasya66@mail.ru

Аннотация. В статье предлагается подход к повышению надежности хранения и обработки данных в облаках за счет избыточной системы остаточных классов (ИСОК). Показано, что использование свойств ИСОК позволяет повысить производительность вычислительных систем в условиях неопределенности времени обработки данных, произвести контроль результатов вычисления, повысить надежность и обеспечить безопасность данных. Использование корректирующих способностей ИСОК позволяет обнаружить, локализовать и исправить возникающие ошибки, уменьшить время ожидания результатов вычислений

Ключевые слова: облачное хранилище; избыточная система остаточных классов; Hadoop MapReduce

I. ВВЕДЕНИЕ

Облачные вычисления играют важную роль в обработке и хранении больших данных. Однако стоит заметить, что хранение и обработка больших данных требует решения ряда проблем, связанных высокой вероятностью потери части или всех данных из-за технических сбоев. Для повышения надежности системы хранения данных используют репликацию и коды исправления ошибок [1, 2].

Одним из механизмов для построения надежных систем хранения и обработки данных является Google File System (GFS). GFS - масштабируемая распределенная файловая система для больших распределенных приложений с интенсивной обработкой данных. Архитектура GFS состоит из кластера, состоящего из одного главного и множества серверных блоков. Принцип хранения в GFS основан на разделении файлов на 64-битные блоки фиксированного размера [8]. Одной из реализаций GFS является Hadoop Distributed File System (HDFS), которая представляет собой систему хранения данных с открытым исходным кодом [3–5].

Основными недостатками HDFS являются увеличение времени получения результата из-за неопределенности времени выполнения задачи на различных устройствах и увеличение нагрузки на вычислительные сети. Для устранения этих недостатков в работе [6] предложена новая структура под названием MapReduce Agent Mobility,

которая обеспечивает отказоустойчивость, уменьшение времени загрузки и требований к памяти.

Для повышения эффективности механизма репликации и уменьшения накладных расходов в работе [7] авторы предложили метод проактивной проверки реплик. Использование данного подхода позволяет уменьшить нагрузку на вычислительные сети. Однако проблема большой избыточности данных остается открытой.

Альтернативным механизмом для обеспечения надежности хранения больших данных и уменьшения их избыточности является использование кодов исправления ошибок на базе системы остаточных классов (СОК). Исследуем вопрос применимости СОК для построения надежной системы хранения и обработки больших данных.

II. БОЛЬШИЕ ДАННЫЕ И ИХ ХАРАКТЕРИСТИКА

Хранение и обработка больших данных требует особых подходов к построению вычислительной системы [2]. Облачные системы позволяют решить проблему хранения и обработки больших данных, генерируемых различными источниками, и являются эффективной альтернативой традиционным системам хранения [1, 2]. Как показано на рис. 1, объем хранимой информации возрастает по экспоненциальному закону.

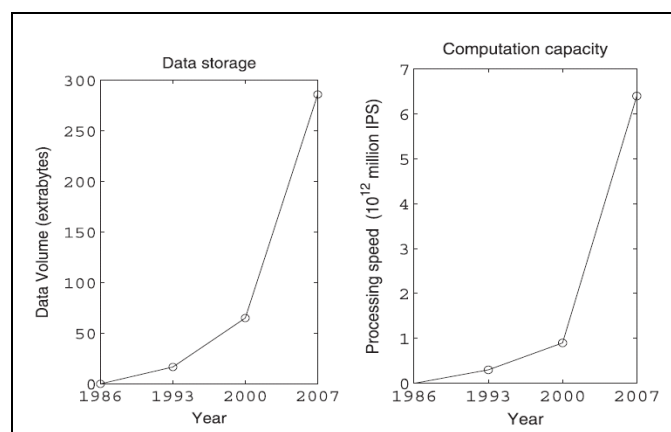


Рис. 1. Зависимость объема хранимых данных от времени

Учитывая динамику роста объема хранимых данных, можно сделать вывод о том, что избыточность данных является ключевой проблемой при хранении информации. Из-за больших объемов данных возникает проблема с масштабируемостью системы [14]. Помимо этого одной из основных проблем при хранении данных является обеспечения ее сохранности.

Популярным методом повышения надежности облачных хранилищ является репликация, которая требует большой избыточности данных [8, 9]. Для обеспечения надежности хранения данных в дата-центрах используют репликацию данных, состоящую из «мастер файла» и трех его реплик [3], что ведет к увеличению объема данных в четыре раза.

Согласно отчету о безопасности «Cloud Security Alliance» при использовании облачных технологий велика вероятность потери данных [10]. Для повышения надежности хранимых данных в работе [1] предложен подход, основанный на идее разбиения файла на куски и загрузки в несколько облачных провайдеров. Преимущество данного подхода заключается в том, что пользователи могут получить файл даже в случае отказа в доступе к данным, хранимым в одном или нескольких облачных хранилищах.

III. ИЗБЫТОЧНАЯ СИСТЕМА ОСТАТОЧНЫХ КЛАССОВ (ИСОК) И ЕЕ СВОЙСТВА

Введем следующие обозначения: $\{p_1, p_2, \dots, p_n\}$ – модули ИСОК, попарно взаимно простые числа; k – количество рабочих оснований, r – количество контрольных оснований; $[0, P-1]$ – рабочий диапазон ИСОК, где $P = \prod_{i=1}^k p_i$; $[0, \bar{P}-1]$ – избыточный диапазон ИСОК, где $\bar{P} = \prod_{i=1}^n p_i$; $\{a_1, a_2, \dots, a_n\}$ – представление числа в ИСОК, где $a_i = |A|_{p_i}$.

Для перевода числа из ИСОК в позиционную систему счисления используют Китайскую Теорему об Остатках следующим образом

$$A = \left| \sum_{i=1}^n P_i \left| P_i^{-1} \right|_{p_i} a_i \right|_P,$$

где $P_i = P/p_i$ и $\left| P_i^{-1} \right|_{p_i}$ – мультипликативное, обратно для P_i по модулю p_i .

ИСОК является нетрадиционным методом, который имеет возможность обнаруживать переполнения и исправлять ошибки, которые могут возникнуть во время передачи данных. Принцип работы алгоритма обнаружение, локализации и исправления ошибок показано на рис. 2.

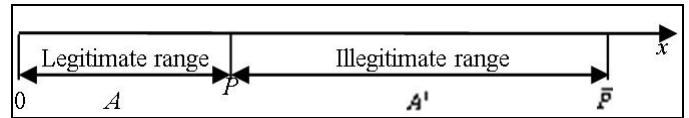


Рис. 2. Структура диапазона RRNS

IV. МОДИФИКАЦИЯ HDFS

Для уменьшения избыточности данных и уменьшения нагрузки на вычислительные сети предлагается использовать ИСОК. Предложенная модель распределяет данные в HDFS путем преобразования их представления в ИСОК. Использование арифметических свойств ИСОК позволяет распараллеливать основные операции сложения, вычитания и умножения и производить контроль результатов вычислений. На рис. 3 представлена модифицированная модель обработки данных.

HDFS предназначена для хранения очень большого количества наборов данных в пределах большого кластера или NameNode, поэтому данные предлагается хранить в остаточном представлении, что позволит уменьшить размер арифметических вычислений, например, относительно простых модулей с динамическим диапазоном. Рассмотрим любые два целых числа, например, 921 и 463, в пределах динамического диапазона. Тогда их двоичное представление будет иметь вид 001110011001 и 000111001111 соответственно и может быть преобразовано в представление остатков относительно выбранного набора модулей. Операции сложения, вычитания и умножения в системе остаточных классов выполняются параллельно.

Данные будут передаваться из HDFS в MapReduce. Перед началом работы в MapReduce будут принимать логическое представление, которое называется расщеплением, что поможет уменьшить количество mapрег в отличие от деления на блоки данных, что означает разбиение больших данных размера на более мелкие части с целью упрощения обработки. Это называется Mapper будет взаимодействовать с InputSplits, как показано на рис. 4. Job Tracker контролирует задачи MapReduce, которые выполняются на различных подчиненных узлах основного кластера (т.е., NameNode). Все задачи map(карте) и reduce stages(этапах сокращения) выполняются треклистами Task, которые являются основными компонентами map и reduce(карты и сокращения).

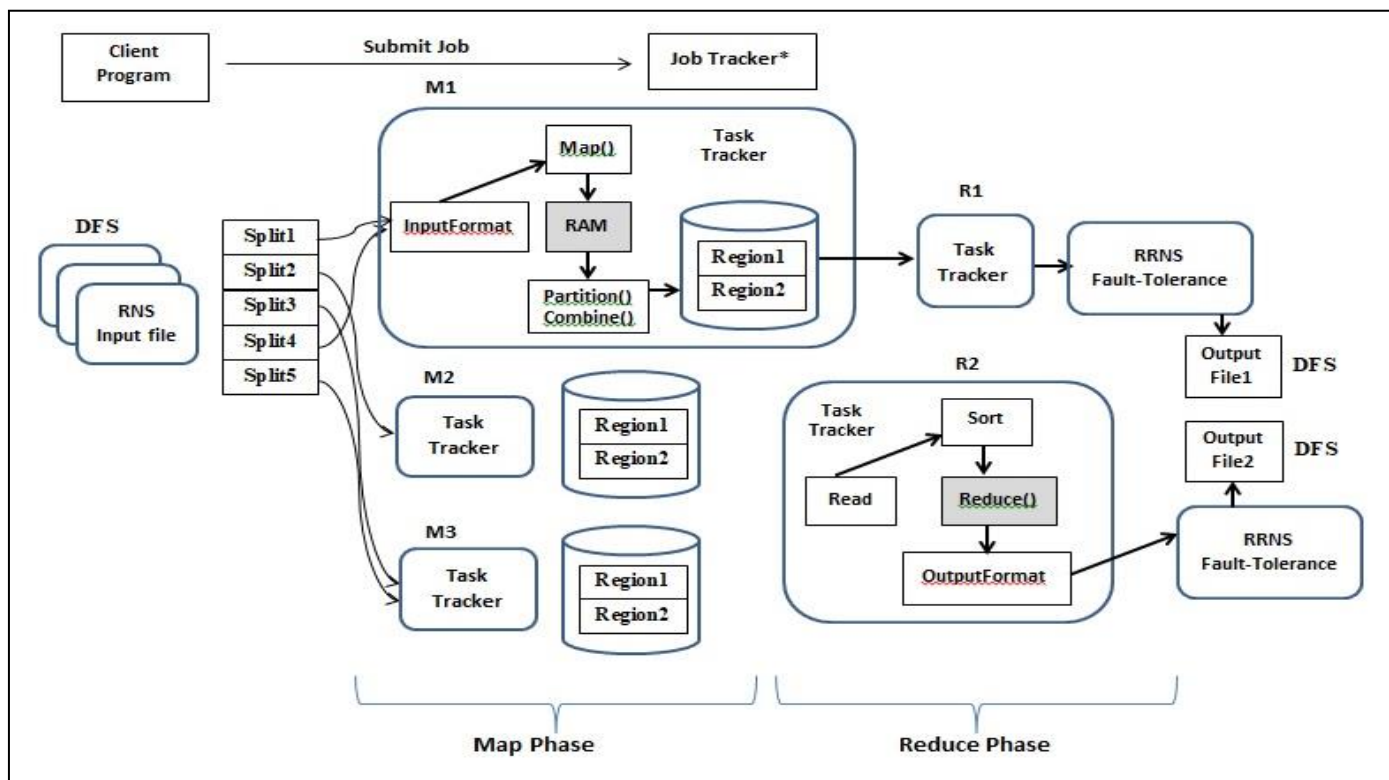


Рис. 3. Предлагаемая модель

Добавленный блок управления и контроля результата в ИСОК, позволяет уменьшить время вычисления, уменьшить нагрузку на сети передачи данных, повысить надежность системы, но при этом требует дополнительных вычислительных затрат связанных с переводом чисел из позиционной системы счисления в ИСОК и обратно [11].

V. ВЫВОД

Предлагаемая модель нацелена на повышение уровня надежности, что можно достичь путем добавления дополнительных алгоритмов ИСОК, обеспечивающих отказоустойчивость и коррекцию ошибок системы после преобразования представления данных из позиционной системы счисления в ИСОК. Использование ИСОК позволяет уменьшить время получения результата, уменьшить нагрузку на вычислительные сети и уменьшить избыточность за счет арифметических свойств ИСОК.

СПИСОК ЛИТЕРАТУРЫ

- [1] Chervyakov N., Babenko M., Tchernykh A., Kucherov N., Miranda-López V, Cortés-Mendoza, J.M., AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security, Future Generation Computer Systems, 2017. DOI: 10.1016/j.future.2017.09.061.
- [2] Tchernykh A., Schwiegelsohn U., Talbi E.G., Babenko M., Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability, Journal of Computational Science, 2016,. DOI: 10.1016/j.jocs.2016.11.011.
- [3] Sanjay G., Howard G., Shun-Tak L., The Google File System, New York, USA. ACM, 2003.
- [4] Li W., Yang Y., Chen J., Yuan D., A cost-effective mechanism for Cloud data reliability management based on proactive replica checking, Proc. of the 12th IEEE/ACM International Symposium on Cluster Cloud and Grid Computing, 2012.
- [5] Karun K.A., Chitharanjan K.A. Review on Hadoop – HDFS Infrastructure Extensions, Proc. of the IEEE Conference on Information and Communication Technologies, 2013, pp. 132-137.
- [6] Youssef M.E., Gamal A., Ayman E. Mobile agent based new framework for improving Big Data analysis, Proc. of the International Conference on Cloud Computing and Big Data, 2013, pp. 381-383.
- [7] Li W., Yang Y., Chen J., Yuan D. A cost-effective mechanism for Cloud data reliability management based on proactive replica checking, Proc. of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2012.
- [8] Sagirolu S., Sinanc S. Big Data: A Review, IEEE, 2013, pp. 42-47.
- [9] Li W. et al. Reliability Assurance of Big Data in the Cloud Cost-Effective Replication-Based Storage, Centre for Computing and Engineering, Software Systems, School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia, 2015.
- [10] Miranda-López V., Tchernykh A., Cortés-Mendoza J. M., Babenko M., Radchenko G., Nesmachnow S., Du Z. Experimental Analysis of Secret Sharing Schemes for Cloud Storage Based on RNS, Communications in Computer and Information Science, 796, 2017, pp. 370-383, DOI: 10.1007/978-3-319-73353-1_26.
- [11] Omondi A., Premkumar B. Residue Number Systems theory and Implementation, London: Imperial College Press, 2007, p. 312.