

УДК 004.7

Проблемы оптимизации и самозащиты репликаций при размещении заданий в облачных вычислениях

А.И. Азиз, Ю.В. Бондаренко*

Воронежский государственный технический университет, ул. 20 лет Октября,
84, Воронеж, 394006, Россия

*E-mail: csit@bk.ru

Аннотация. Репликация данных, которая приближает файлы к потребителям данных – хорошо известный метод, сокращающий время доступа и потребление полосы пропускания. Моделирование с помощью CloudSim доказывает, что предлагаемый алгоритм репликации превосходит основные существующие стратегии репликации с точки зрения коэффициента совпадений, количества репликаций, дисперсии нагрузки, задержки, среднего времени обслуживания, доступности и энергопотребления.

Ключевые слова: репликация данных сети, полоса пропускания, облачный сервис, защита

Problems of optimization and self-defense of replication when placing tasks in cloud computing

A.I. Aziz, Yu.V. Bondarenko*

Voronezh State Technical University, 84, 20 let Oktyabrya str., Voronezh, 394006,
Russia

*E-mail: csit@bk.ru

Abstract. Data replication, which brings files closer to data consumers– is a well-known method that reduces access time and bandwidth consumption. Modeling using CloudSim proves that the proposed replication algorithm surpasses the main existing replication strategies in terms of the coincidence coefficient, number of replications, load variance, latency, average service time, availability and power consumption.

Keywords: network data replication, bandwidth, cloud service, protection

1. Введение

Облачные вычисления привлекают все большее внимание в управлении данными. Репликация данных, которая приближает файлы к потребителям данных – хорошо известный метод, сокращающий время доступа и потребление полосы пропускания. Существуют две проблемы, касающиеся процесса размещения реплик. Во-первых, как снизить затраты на доступ и репликацию, которые являются двумя противоречащими друг другу целями. Для этого необходим многоцелевой оптимизированный алгоритм размещения, основанный на метаэвристическом методе и нечеткой системе, который находит оптимальные местоположения для реплик путем уравнивания компромиссов между шестью задачами оптимизации (т.е. доступность системы, время обслуживания, нагрузка, потребление энергии, задержка и центральность). Второй вопрос – как определить оптимальное количество реплик, поскольку хранение большого количества реплик в облаке – дорогое удовольствие.

2. Цель исследования

Целью исследования является исследование проблем оптимизации и защиты репликаций при размещении заданий в облачных вычислениях.

2.1. Облачные вычисления как современная технология

В настоящее время облачные вычисления стали самой популярной и надежной вычислительной технологией и привлекают все больше и больше внимания во всем мире [1]. Облако разделяет гетерогенный ресурс на основе технологии визуализации и, таким образом, предоставляет гибкие услуги и услуги по запросу для разных пользователей [2]. Многие инженерные и научные приложения, такие как физика элементарных частиц и генетика, работают с огромными объемами данных. Такие крупномасштабные данные обычно хранятся в облачных центрах обработки данных [3]. В связи с этим эффективное управление данными является важной проблемой в облаке и сопряжено с множеством проблем. Репликация данных – хорошее решение этих проблем путем создания реплик на географически распределенных сайтах. Если необходимые файлы реплицируются в некоторых дата-центрах, где выполняется задача, то задача использует нужные файлы без задержки связи. Следовательно, снижаются затраты на доступ к данным. Более того, потеря небольшого количества данных может привести к серьезному экономическому кризису [4].

2.2. Самозащищаемая репликация данных в облачной среде

В настоящее время организации хранят и обрабатывают крупномасштабные данные, поскольку общество становится все более оснащенным. Следовательно, управление полученными данными является ключом к конкурентному преимуществу. Распределенные среды, такие как grid-вычисления, облачные вычисления и мобильные вычисления, хорошо поддерживают известные методы оптимизации для повышения качества предоставляемых услуг. Размер и количество облачных центров обработки данных быстро растет, а масштабируемые виртуализированные ресурсы представлены как услуга по сети. Данные играют главную роль в научных дисциплинах облачных вычислений, где большие коллекции данных являются важными ресурсами. Нам необходимо эффективное управление данными, чтобы предоставлять данные или услуги пользователю по запросу [5]. Репликация данных – это эффективный метод повышения доступности данных, снижения потребления полосы пропускания, и минимизировать время доступа пользователей за счет хранения реплик в нескольких местах. Однако с увеличением количества реплик стоимость управления данными будет увеличиваться [6]. Слишком большое количество реплик может увеличить доступность данных, но некоторые ненужные реплики приводят к бесполезной трате ресурсов. Следовательно, реплики и запросы должны быть разумно распределены для достижения высокой производительности облака и оптимального использования ресурсов. Размещение реплик и управление такими широко распределенными данными становится более сложным по сравнению с мелкомасштабной средой.

Например, если реплики распределены случайным образом, возможно, что одни центры обработки данных очень заняты, а другие простаивают. Следовательно, возникает ряд проблем, таких как дисбаланс нагрузки в облачных центрах обработки данных, плохой параллелизм и низкая производительность. Этот важный недостаток вызывает серьезную озабоченность: как разместить новые реплики и эффективно распределить рабочие нагрузки между облачными центрами обработки данных? Хотя проблема репликации данных уже исследовалась ранее [7], очень немногие из доступных алгоритмов дают целостное представление о различных затратах и преимуществах репликации данных в облачной среде.

Стратегии репликации можно разделить на два типа в зависимости от того, меняются ли количество реплик и расположение реплик во время выполнения: статические и динамические. Стратегия статической репликации определяет положение

реплик во время разработки, а набор реплик и их расположение не изменяются во время выполнения. Слабость метода статической репликации очевидна, когда шаблоны доступа пользователей в облаке сильно меняются, преимущества, которые приносят предопределенные реплики, резко уменьшаются. Стратегия динамической репликации автоматически сохраняет и удаляет реплики в зависимости от изменений поведения пользователя. Из-за динамического характера облачной среды и пользователей для облака подходит метод динамической репликации.

3. Методы и материалы исследования

Необходимо отметить, что частые крупномасштабная передача данных, являющаяся следствием динамической репликации, может привести к чрезмерной нагрузке на ресурсы системы. Следовательно, стратегия репликации должна избегать ненужных реплик в системе и хранить реплики в лучших местах. Алгоритм репликации данных требует ответа на шесть основных вопросов:

- Когда реплицировать?
- Какие файлы копировать?
- Где воспроизвести? (Размещение реплики).
- Какая реплика для доступа к данным для выполнения задания? (Выбор реплики).
- Какие реплики заменить? (Замена реплики).
- Сколько нужно создать подходящих новых реплик? (Оптимальное количество реплик).

Важная проблема – сведение к минимуму количества реплик в системе. Большое количество реплик сокращает расстояние между инициатором запроса данных и ближайшим поставщиком реплик. Очевидно, что создание и обслуживание большого количества реплик в облаке стоит дорого, поэтому количество реплик следует оптимизировать. Следовательно, минимизация стоимости доступа и стоимости репликации – это компромисс. Таким образом, при разработке алгоритма динамической репликации данных необходимо уравнивать конфликтующие цели без снижения производительности. Репликация данных определяется как многокритериальная задача оптимизации, и исследователи применяют вдохновленные природой алгоритмы (например, пчелиные колонии, муравьиные колонии и генетический алгоритм), чтобы найти почти оптимальные решения, поскольку природные методы дают лучший результат, чем традиционный метод. Алгоритм самозащиты – это недавно

разработанный эвристический алгоритм, вдохновленный механизмами растений в природе.

4. Полученные результаты

Для решения поставленных задач планируется к реализации:

1. Метод адаптации динамических параметров в алгоритме самозащиты на основе нечеткой логики.
2. Моделирование для проблем репликации данных шести противоречивых целей, таких как доступность системы, среднее время обслуживания, средняя задержка, дисперсия нагрузки, энергопотребление и центральность.
3. Самостоятельный алгоритм защиты (FSDA) для достижения баланса между шестью целями.
4. Система нечеткого вывода используется для оценки пригодности и более точного описания решения.
5. Хранение новой реплики в наиболее удобном месте, что сокращает время доступа, обеспечивая при этом эффективное использование сети и ресурсов.
6. Оптимизация количества реплик, так как поддержание нескольких копий в облаке очень дорого.

5. Выводы

В статье рассматривается необходимость многоцелевого оптимизационного алгоритма размещения, основанного на метаэвристическом методе и нечеткой системе, который находит оптимальные местоположения для реплик путем уравнивания компромиссов между шестью задачами оптимизации (т.е. доступность системы, время обслуживания, нагрузка, потребление энергии, задержка и центральность).

Также формулируется задача определения оптимального количества реплик, поскольку хранение большого количества реплик в облаке – дорогое удовольствие.

Список литературы

1. Sun, P. Security and privacy protection in cloud computing: discussions and challenges / P. Sun // J. Netw. Comput. Appl. – 2020. – 160(2). – P. 102642.
2. Abdullahi, M. An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment / M.

- Abdullahi, N. M. Asri, S.I. Dishing, S.M. Abdulhamid, B.I. Ahmad // J. Netw. Comput. Appl. – 2019. – 133. – Pp. 60-74.
3. Kumara, M. A comprehensive survey for scheduling techniques in cloud computing / M. Kumara, S.C. Sharma, A. Goel, S.P. Singh // J. Netw. Comput. Appl. – 2019. – 143. – Pp. 1-33.
 4. Sasikumar, K. An efficient replica management based disaster recover using elephant herding optimization algorithm / K. Sasikumar, B. Vijayakumar // International Conference on Computer Networks and Inventive Communication Technologies. – 2020. – Pp. 392-399.
 5. Xia, Q. Efficient data placement and replication for QoS-aware approximate query evaluation of big data analytics / Q. Xia, Z. Xu, W. Liang, S. Yu, S. Guo // IEEE Trans. Parallel Distr. Syst. – 2019. – 30(12). –Pp. 2677-2691.
 6. Mokadem, R. A data replication strategy with tenant performance and provider economic profit guarantees in cloud data centers / R. Mokadem, A. Hameurlain // J. Syst. Software. – 2020. – 159(4). – P. 110447.
 7. Mansouri, N. Using data mining techniques to improve replica management in cloud environment / N. Mansouri, M.M. Javidi, B. Mohammad Hasani Zade //Soft Comput. – 2019. – № 24. – С. 7335-7360.