

ОПТИМИЗАЦИЯ РАСПРЕДЕЛЕНИЯ ЗАДАЧ МЕЖДУ ХРАНИЛИЩАМИ РЕЗЕРВНОГО КОПИРОВАНИЯ НА ОСНОВЕ МУЛЬТИАГЕНТНОГО ПОДХОДА

Васильев А.Д. аспирант кафедры системного программирования
СБбГУ,

st047300@student.spbu.ru

Аннотация

Рассматривается применение мультиагентной системы на основе алгоритма локального голосования для распределения задач резервного копирования. Представлены результаты моделирования и их сравнение с реализацией распределения задач на основе расписания.

Введение

Тема резервного копирования несомненно затрагивает каждого из пользователей современных технологий. Физически носители данных могут выходить из строя, а в последние годы возросло число атак различных вирусов, например таких как шифровальщики. Описанные выше случаи могут произойти как в огромным компаниям, чьи сервисы должны быть доступны 24 часа в сутки, так и со студентом, чей ноутбук вышел из строя перед защитой диплома. Во всех этих случаях может помочь резервное копирование данных.

Тенденцией последних десятилетий является рост объемов данных хранимых, обрабатываемых и передаваемых по сети. Этот процесс можно наблюдать в различных областях, таких как экономика, научные исследования, метеорология и т.д. Помимо увеличения нагрузки на сеть, это приводит к увеличению нагрузки на отдельные машины. В результате возникает множество проблем в области оптимизации нагрузки.

Процесс резервного копирования создает дополнительную нагрузку на сеть и вычислительные мощности как для машины являющейся объектом резервного копирования так и для хранилища резервных копий. Обычно хранилище резервного копирования представляет из себя Систему Хранения Данных — СХД. В качестве СХД может выступать Direct-attached storage, Network-attached storage или Storage area

network. Вычислительная мощность СХД может существенно различаться. Чтобы справиться с этими проблемами администратору резервного копирования требуется решить две задачи.

Первая из них это составление расписания для окна резервного копирования — времени от начала до завершения процедуры резервного копирования. В прошлом оно обычно сооветствовало ночному времени суток, когда нагрузка на инфраструктуру минимальна. Однако, многие современные вычислительные мощности должны быть доступны 24 часа в сутки круглый год. Это приносит большие сложности для составителя расписания резервного копирования.

Ещё одной проблемой могла стать перегрузка некоторых хранилищ резервных копий, которая возникает в результате не верного распределения нагрузки на хранилища данных. В больших компаниях есть тысячи станций пользователей и сотни серверов, которые нужно распределить по хранилищам резервных копий. Дополнительным фактором сложности является то, что количество машин, которые требуется защищать может изменяться, а СХД могут выходить из строя.

В условиях изменяющейся инфраструктуры создание оптимальной конфигурации резервного копирования является сложной задачей поставленной перед администратором. При этом учет особенностей СХД, например производительности, также остается его задачей. В случае ошибок планирования окно бекапа может растянуться на день, а машины будут застревать в очереди на резервное копирование, в следствии чего критически важные машины, например сервера баз данных, могут не получать необходимую им ежедневную защиту.

Решить эту проблему можно несколькими путями. Например, добавлением новых хранилищ т.е. горизонтальным масштабированием. Это дорогое решение, требующее траты на оборудование и его сопровождение или оплату хранения данных в облаке. Кроме этого требуется изменение конфигурации резервного копирования.

Альтернативой может выступить подход, в котором задачи сами оптимально распределяются между хранилищами, основываясь на информации о размере машин и производительности хранилища. Такая оптимизация позволила бы снизить нагрузку на серверы и машины, а значит уменьшить затраты на горизонтальное масштабирование. Это становится особенно актуально в смешанных сетях, где часть ресурсов расположено локально, часть в Центре Обработки Данных (ЦОД), а часть располагается в облаке.

Стандартным, при необходимости резервного копирования, является использование централизованного сервера, который управляет по-

токами заданий и передачи данных (см. Рис. 1). Такой сервер мог бы отвечать за автоматическое оптимальное распределение нагрузки. Однако, в этом подходе есть ряд недостатков, таких как слабая отказоустойчивость, плохая масштабируемость и низкая производительность.

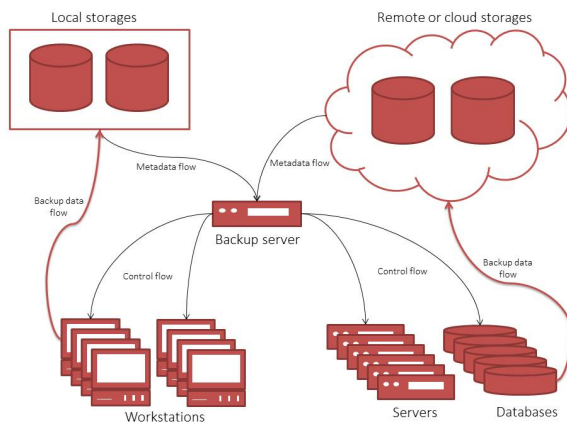


Рис. 1: Схема организации резервного копирования с центральным сервером

Цель данной статьи — продемонстрировать механизм, основанный на мультиагентном походе и предполагающий размещение интеллектуального агента на каждом из СХД (см. Рис. 2). Вместе эти агенты образуют сеть хранилищ между которыми могут перераспределяться задачи. Задачи представляют собой описание размеров защищаемой машины и поступают в хранилище задач. Агенты забирают задачи из общего хранилища задач по мере их поступления. Применение мультиагентного подхода значительно упрощает добавление новых СХД в сеть и позволяет избежать бутылочного горлышка, которое создается центральным сервером резервного копирования.

Схожий подход к распределению нагрузки в беспроводных сетях был продемонстрирован в [3]. В работах [1], [2] было показано, что оптимальное распределение заданий может формулироваться как задача достижения консенсуса в сети. Для достижения поставленной цели необходимо адаптировать алгоритм локального голосования описанный в работе [1], реализовать прототипа и провести моделирование с целью оценки оптимальности алгоритма и отказоустойчивости.

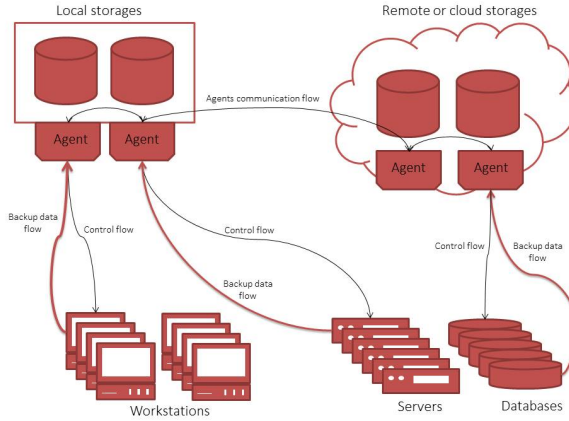


Рис. 2: Схема организации резервного копирования на основе мультиагентного подхода

Постановка задачи

Пусть $N = 1, \dots, n$ множество интеллектуальных агентов. Задачи поступают на разные узлы и сохраняются в очередь, свою для каждого агента. В момент времени t связь между узлами определяется графом G_t . Кроме этого для i — агента определены q_t^i — длина очереди в момент времени t и s_t^i — производительность узла в момент времени t $i = 1, \dots, n$ и $i \in N$.

С течением времени состояние системы изменяется. В момент времени t для каждого агента в системе поступают новые задания z_t^i . Тогда динамику изменения состояния системы можно описать как

$$q_t^i + 1 = q_t^i - s_t^i + z_t^i + u_t^i \quad (1)$$

, где u_t^i — управление системой, которое перераспределяет задания между агентами. Задача состоит в поддержке равномерной нагрузки на все узлы сети.

В такой постановке задачи наименьшее время работы системы будет получено при равномерной загрузке узлов, при котором $q_t^i/s_t^i = q_t^j/s_t^j \forall i, j \in N$. Выполнение заданий осуществляется только работоспособными узлами для которых $s_t^i > 0$. При этом перераспределение задач может происходить между всеми узлами сети. Это позволяет не учитывать особенности топологии связей между агентами.

Протокол управления

Топология сети в каждый момент времени t определяется с помощью оргарфов $G_t = (N, T_4)_{t \geq 0}$, где $N = i, \dots, n$ - вершин, где каждая вершина соответствует агенту. В свою очередь E_t - меняющееся во времени множество ребер соответствующее множеству связей между агентами. Элемент множества E_t определяется парой (i, j) и обозначает получение информации агентом i от агента j в момент времени t .

Пусть $u_t^i \in R$ - управляющее воздействие, которое узел i получает в момент времени t . Каждый узел обладает информацией о своем состоянии $y^{ii} = x_t^i$ и зашумленной информацией о состоянии соседей $y^{ij} = x_{t-d_t^{ij}}^j + w_t^{ij}$, $J \in N_t^i$, где w_t^{ij} - помехи в момент времени t , а $0 \leq d_t^{ij} \leq D$ - задержка в момент времени t , которая ограничена максимальной задержкой D . Воспользуемся «протоколом локального голосования», который определяет управление как:

$$u_t^i = a_t \sum_{j \in N_t^i} b_t^{ij} (y_t^{ij} - y_t^{ii}) \quad (2)$$

т.е. определим управление как взвешенную сумму разницы его состояния и зашумленной информации о его соседях. $a_t > 0$ - шаг протокола управления.

Будем считать, что агенты i и j находятся в состоянии согласования, когда $x_t^i = x_t^j$. Для достижения консенсуса требуется согласованность между всеми узлами.

Таким образом динамика замкнутой системы 1 имеет следующий вид:

$$x_{t+1}^i = x_{t-1}^i + a_t \sum_{j \in N_t^i} b_t^{ij} * (y_t^j / s_t^j - q_t^i / s_t^i) \quad (3)$$

Моделирование

В рамках моделирования были реализованы прототипы агентов и центрального хранилища задач резервного копирования. Было проведено три эксперимента, в каждом из них задачи распределялись между 30 СХД. 10 из них обрабатывают данные со скоростью 100 Мб/сек, 10 со скоростью 50 Мб/сек и 10 со скоростью 25 Мб/сек

В первом эксперименте 600 задач случайно распределялись между агентами, постепенно приходя в систему. Далее производилось сравнение мультиагентного алгоритма — Multi-Agent Algorithm (MAA)

и случая, когда задачи распределялись случайно — Random Task Distribution (RTD). Эксперимент производился по 100 раз, на графике представлено среднее значение времени окна резервного копирования Рис. 3.

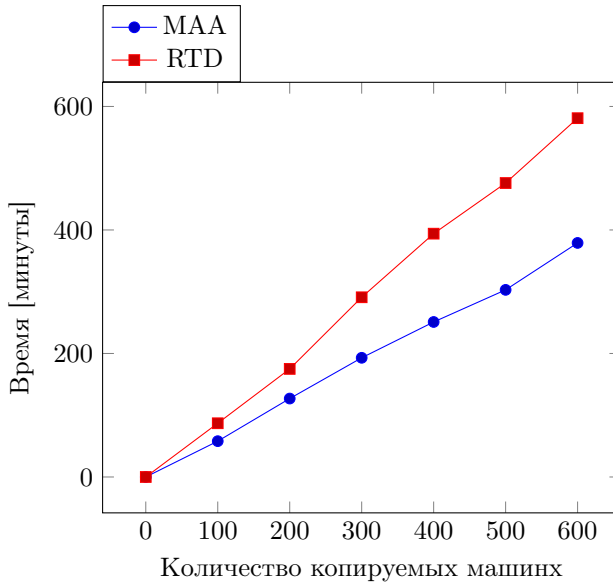


Рис. 3: Сравнение случайного и оптимизированного распределения задач

Во втором эксперименте мультиагентный алгоритм сравнивался с расписанием бекапов — Scheduled Task Distribution (STD) для 600 машин из 10 конфигураций различающихся по количеству хранимых данных. Это расписание считало оптимальным. Расчеты времени окна резервного копирования показаны на Рис. 4. В третьем эксперименте расширяется эксперимент два и происходят случайные отключения хранилищ. В случае такого отключения задачи распределенные по расписанию могли быть переданы на случайный СХД. Эксперимент производился по 100 раз, на графике представлено среднее значение времени окна резервного копирования показаны на Рис. 4.

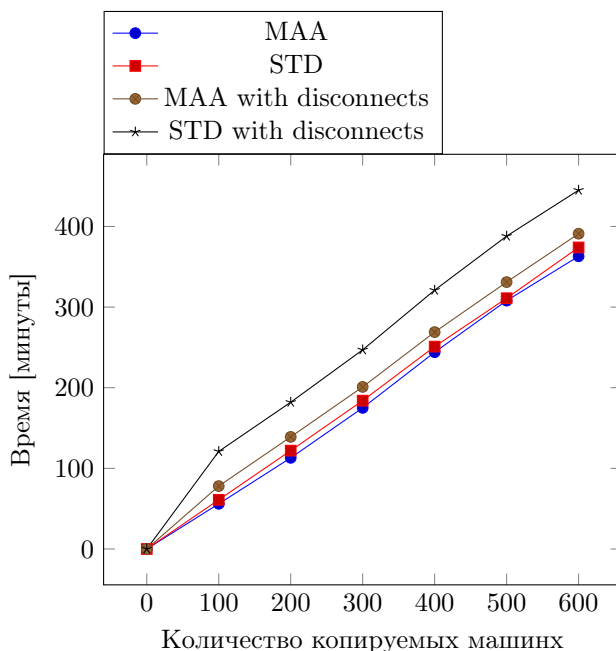


Рис. 4: Сравнение расписания задач резервного копирования и оптимизированного распределения задач в условиях разрывов соединений и без

Заключение

В статье было описано новое применение алгоритма локального голосования для распределения задач между хранилищами резервного копирования и приведены результаты моделирования, выполненного на основе описанного алгоритма. Продемонстрированы преимущества такого подхода по сравнению с традиционным.

Существуют несколько параметров, учет которых необходим, чтобы сделать технологию приближенной к реальным условиям:

1. Учет стоимости хранилищ, на которых установлены агенты. Это важно т.к. некоторые агенты могут работать на локальных СХД, которые принадлежат пользователям, а некоторые могут быть расположены в облаке. Стоимость таких ресурсов может принципиально различаться и быть очень важна;

2. Учет различий в производительности разных участков сети. В задаче оптимизации резервного копирования этот параметр крайне важен, т.к. зачастую именно передача данных занимает большую часть времени, а скорость соединения может быть разной для ЦОД и локальной серверной;
3. Учет размеров хранилищ. В этой статье хранилища считаются безразмерными, однако в реальной жизни это совершенно не так;
4. Учет приоритетов задач. Задания резервного копирования могут иметь разный приоритет. Так например для пользователей почты всегда гораздо приоритетней резервное копирование серверов с базами данных, чем рабочих станция обычных сотрудников.

Список литературы

- [1] Amelina Natalia, Granichin Oleg, Kornivets Aleksandra. Local voting protocol in decentralized load balancing problem with switched topology, noise, and delays // 52nd IEEE Conference on Decision and Control / IEEE. — 2013. — P. 4613–4618.
- [2] Multi-agent stochastic systems with switched topology and noise / Konstantin Amelin, Natalia Amelina, Oleg Granichin, Olga Granichina // 2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing / IEEE. — 2012. — P. 438–443.
- [3] Toward optimal distributed node scheduling in a multihop wireless network through local voting / Dimitrios J Vergados, Natalia Amelina, Yuming Jiang et al. // IEEE Transactions on Wireless Communications. — 2018. — Vol. 17, no. 1. — P. 400–414.