

ПРОБЛЕМА ДОСТИЖЕНИЯ КОНСЕНСУСА В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Янковский А. Н.

Волорова Н. А. – канд. техн. наук, доцент

Одной из главных целей распределенных вычислений является достижение надежности всей системы в целом при наличии сбоев в некоторых ее компонентах. Это зачастую требует от участников согласия относительно некоторого значения, необходимого во время вычислений, – т.е. достижения консенсуса. В общем случае, некоторые из участников могут быть испорченными (например, взломанные сервера, неисправные сенсоры) и пытаться нарушить работоспособность системы и препятствовать достижению консенсуса среди других участников.

Несмотря на то, что разработка распределенных систем ведется уже не первое десятилетие, в этой области до сих пор остается множество открытых вопросов. Не упрощает дело и тот факт, что с каждым годом требования к надежности и производительности подобных систем растут, равно как и объемы обрабатываемых данных. Однако перед тем, как вести речь о проблемах и вариантах их решения, необходимо сначала определиться с терминологией и дать определения ключевым понятиям.

Консенсус – это задача получения согласия всех участников о некотором конкретном значении на основе их голосов. Все участники должны должны признать одно и то же значение, и оно должно быть предложено одним из участников (т.е. алгоритм достижения консенсуса не может взять значения из ниоткуда). В самом простом случае, значение может быть бинарным (0 или 1), что может использоваться участниками для принятия решение вроде “делать что-либо или нет”. Примеры ситуаций, когда консенсус оказывается полезным:

- Синхронизация реплицированных конечных автоматов и обеспечение гарантии, что все реплики имеют одинаковое (консистентное) представление состояния системы
- Выборы лидера (например, для взаимного исключения)
- Распределенное журналирование с глобальной последовательной согласованностью

В идеальном мире консенсуса достичь легко. Например, обсуждая алгоритмы распределенного взаимного исключения, мы приходим к одной из форм консенсуса, когда все принимают одно и то же решение касательно того, кто должен получить доступ к ресурсу. Наиболее простая реализация – это назначить общего координатора, которые будет отвечать за определение исхода. Двухфазный коммит также является примером системы, когда мы полагаем, что координатор и участники функционируют и могут общаться. Но подвох в этих алгоритмах заключается в том, что все участники должны быть работоспособны. Сбои все усложняют. Под сбоями понимаются отказы участников и сетевые ошибки.

Невозможно гарантированно достичь консенсуса при наличии сбоев у полностью асинхронных участников. Это означает отсутствие каких-либо предположений касательно скоростей участниково или сетевых коммуникаций. Суть проблемы заключается в том, что невозможно проверить, сломался ли участник или же участник в порядке, но сообщения доходят до него неприемлемо медленно. Это было доказано Фишером, Динчем и Паттерсоном. Также, консенсус невозможен в условиях ненадежности коммуникаций, т.к. мы можем не иметь возможности общаться с участником.

Сеть никогда не является 100% надежной и это необходимо учитывать. В формальных доказательствах распределенных систем обычно полагается, что сеть является асинхронной, т.е. сообщения между участниками могут дублироваться, теряться, задерживаться или перемешиваться произвольным образом. Это довольно слабое предположение: некоторые физические сети могут предоставить большие гарантии, но на практике в IP сетях встречаются все эти виды сбоев, поэтому теоретические ограничения асинхронных сетей применяются и к реальным системам.

При возникновении сетевых разделов многие системы переходят в специальный режим с ограниченной функциональностью. CAP теорема гласит, что мы можем обеспечить либо согласованность (технически, линейризуемость на операции чтения-записи), либо доступность (все сервера продолжают обрабатывать запросы), но не одновременно. Что примечательно, немногие базы данных достигают теоретических ограничений CAP, многие просто теряют данные.

Другой вид сбоев в распределенных системах – это ошибки участников (но не fail-stop) в условиях надежной сети. В таких ситуациях сервера продолжают работать, но вместо того, чтобы молчать, они посылают неправильные данные. Эти ошибки называются византийскими в честь известной Проблемы Византийских генералов. В этой проблеме есть n генералов, которые управляют разными отрядами. Коммуникации надежны, однако m генералов – предатели и стараются помешать остальным прийти к согласию, посылая им некорректную информацию. Вопрос заключается в следующем: могут ли верные генералы достичь консенсуса?

Лэмпорт продемонстрировал решение, которое работает в некоторых случаях. Согласно ему, любое решение задачи с m предателями требует как минимум $3m+1$ участников (т.е. $2m+1$ верных генералов). Это означает, что более $\frac{2}{3}$ генералов должны быть верными. Более того, было доказано, что один алгоритм не может справиться с m сбоями менее, чем за $m+1$ раундов обмена сообщениями и $O(m^2)$ самих сообщений.

Если $n < 3m+1$, то задача не имеет решения. Очевидно, что это довольно дорогое решение. Существует вариация Проблемы Византийских генералов, в которой используются подписанные сообщения. Это означает, что сообщения от верных генералов не могут быть подделаны или модифицированы. Для таких случаев существуют алгоритмы, позволяющие достичь консенсуса при условиях $n \geq m + 2$.

Архитектура отказоустойчивых распределенных систем очень сложна. Существует множество различных проблем и ошибок, которые могут возникать в условиях распределенности и система должны уметь их обрабатывать. Корректно обрабатывать ошибки – наиболее сложная задача. На практике большинство ошибок вызваны проблемами с сетью, а византийские модели редко используются в силу своей сложности. Существуют готовые алгоритмы для решения проблемы консенсуса, однако они часто полагаются на различные допущения и поэтому не всегда могут быть использованы на практике без дополнительных модификаций.

Список использованных источников

1. Michael J. Fischer, Nancy A. Lynch and Michael S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process," Journal of the ACM, April 1985, 32(2):374-382. <https://groups.csail.mit.edu/tds/papers/Lynch/jacm85.pdf>
2. D. Dolev. "The Byzantine Generals Strike Again," Journal of Algorithms, 3, 1982, pp. 14-30.
3. L. Lamport. 1983. The Weak Byzantine Generals Problem. J. ACM 30, 3 (July 1983), 668-676. DOI=<http://dx.doi.org/10.1145/2402.322398>
4. Michael J. Fischer, Nancy A. Lynch, and Michael Merritt. 1985. Easy impossibility proofs for distributed consensus problems. In Proceedings of the fourth annual ACM symposium on Principles of distributed computing (PODC '85), Michael Malcolm and Ray Strong (Eds.). ACM, New York, NY, USA, 59-70. DOI=<http://dx.doi.org/10.1145/323596.323602>

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ КОГНИТИВНЫХ СЕРВИСОВ IBM WATSON, ДОСТУПНЫХ В РАМКАХ ОБЛАЧНОЙ ПЛАТФОРМЫ IBM BLUEMIX

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Александр Александров, Виктор Козуб

Иван Пилецкий, канд. физ-мат. наук, доцент

IBM Watson представляет собой когнитивную систему, которая способна понимать, делать выводы и обучаться. Она также позволяет преобразовывать целые отрасли, различные направления науки и техники. Например, предсказывать появление эпидемий или возникновения очагов природных катастроф в различных регионах, вести мониторинг состояния атмосферы больших городов, оптимизировать бизнес-процессы, узнавать, какие товары будут в тренде в ближайшее время. Доступ к части сервисов IBM Watson доступен через использование API облачной платформы IBM Bluemix. О некоторых возможностях IBM Watson и пойдет речь в данной работе.

IBM Watson — одна из первых когнитивных систем в мире. Эта система умеет очень многое, благодаря чему возможности Watson используются во многих сферах — от кулинарии до предсказания аварий в населенных пунктах. В целом-то, большинство возможностей Watson не являются чем-то уникальным, но в комплексе все эти возможности представляют собой весьма мощный инструмент для решения разнообразных вопросов. Например — распознавание естественного языка, динамическое обучение системы, построение и оценка гипотез [1]. Все это позволило IBM Watson научиться давать прямые корректные ответы (с высокой степенью достоверности) на вопросы оператора. При этом когнитивная система умеет использовать для работы большие массивы глобальных неструктурированных данных, Big Data. Получить доступ к возможностям Watson можно используя облачную платформу IBM Bluemix.

В США были проведены психологические исследования, которые подтвердили, что стиль письма человека, порядок слов, эмоциональный окрас напрямую связаны с чертами характера данного человека [2]. Предсказание к предрасположенности к определенным видам деятельности может быть сделано на основании анализа определенных черт характера данного человека. Результаты таких исследований напрямую легли в разработку алгоритмов когнитивных систем.

Одним из сервисов IBM Watson является сервис, позволяющий получить портрет личности человека, основываясь на лингвистическом анализе сообщений из соцсетей, электронной почты или любых других источников текста, который написал данный человек. Этот сервис носит название Personality Insights. Personality Insights базируется на психологии языка вкупе с алгоритмами обработки данных. Выходными данными данного сервиса является представление черт характера исследуемой личности в трёх моделях: модель Большой пятёрки (Big Five), модель потребностей (Needs) и модель ценностей (Values). В рамках данной работы мы исследовали возможности IBM Watson для применения в повседневной жизни. Одну из задач, которые мы попробовали решить, была задача выбора специальности для абитуриента, основываясь на портрете личности данного человека.

Для получения портрета личности, пользователь должен ввести текст о себе, либо предоставить свой