

КЛАССИФИКАЦИЯ ПОДХОДОВ В ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ ДЕЦЕНТРАЛИЗОВАННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Введение

В настоящий момент значительную актуальность приобрели вычислительные системы, организованные из большого количества разнородных или самостоятельных рабочих станций, объединенных сетевой средой. В качестве примеров можно привести интернет-приложение, выполняемое кластером разнородных вычислительных станций, виртуализованную среду распределенных вычислений, поддерживаемую децентрализованной Grid-системой, пиринговую систему обмена контентом [1].

В итоге развиваются различные программные среды, выполняющие виртуализацию предоставляемых отдельным узлом служб и объединение узлов в одну логическую систему. В качестве примеров здесь укажем такие среды как JXTA 2, GridGain, OverlayWeaver.

При разработке таких программных сред необходимо оценивать качество реализации определенных компонент и алгоритмов. Так как опытная эксплуатация такой системы связана с большими затратами, то практически оправданным оказывается имитационное моделирование.

В представленной статье выполняется краткий обзор подходов в имитационном моделировании, применяемом при разработке и проектировании распределенных вычислительных систем. Цель представленной статьи — выбор подходов, применимых для децентрализованных систем большого масштаба, и его обоснование. Под системами большого масштаба здесь и далее подразумеваются системы с тысячами узлов.

Классификация средств моделирования вычислительных систем

Выбор соответствующего уровня детализации для задачи имитационного моделирования является важной и сложной задачей. Слишком слабая детализация может привести к некорректности результатов, но усиление детализации требует времени для реализации, отладки и сопровождения, замедляет выполнение моделирования. Все системы моделирования можно считать специализированными, так как создатели вынуждены уменьшать уровень детализации за пределами непосредственной области исследования.

© А.С. Краевой, 2009

Средства моделирования можно разделить по уровню детализации моделирования. Можно выделить классы средств, используемые для логического, статистического и структурного моделирования. Логическое моделирование подразумевает разработку формальной модели взаимодействий компонент системы и последующее использование программных средств для проверки соответствия взаимодействий определенным критериям. Статистическое моделирование позволяет воспроизвести сложные динамические явления в моделируемой сети: коммутацию пакетов, управление очередями потоковой передачи. Структурное моделирование применяется к системам большого масштаба, на котором статистические свойства структуры сети распределенной системы оказывают заметное влияние на характеристики ее работы.

Моделирование систем для проверки логических свойств

Для реализации любой прикладной задачи в распределенной вычислительной системе необходимо задать протокол взаимодействия ее компонент. Такими компонентами могут быть узлы децентрализованной системы контроля версий, узлы-реплики кластера-хранилища данных, процессоры, совместно использующие систему памяти с иерархическим кешированием. В каждом конкретном случае протоколы могут содержать различное количество элементарных операций, внутренних состояний и их изменений.

Обычно исследователи задают формальную модель протокола и используют определенные программные средства для подтверждения его корректности, логической целостности и непротиворечивости. При этом программные средства моделирования используют различные стратегии поиска и верификации в пространствах состояний сложных систем.

С помощью указанного поиска возможно определить достижимость состояний, связанных с такими проблемами как непредсказуемое поведение компонент, игнорирование сообщений, взаимные блокировки, переполнение очередей сообщений. Такие средства рассматривают вычислительную систему как иерархию параллельно работающих дискретных автоматов, обменивающихся асинхронными сообщениями, и, возможно, обладающих общим состоянием. Эти модели можно строго формализовать посредством численных и цветовых сетей Петри, которые для выражения некоторых аспектов расширяются посредством добавления очередей, задержек и приоритетов.

Также существуют специализированные стандарты для формального описания таких систем: SDL/Promela [2]. Эти стандарты можно использовать для описания алгоритмов взаимного исключения доступа, коммуникационных протоколов, управляющих устройств, процедур выбора лидера. Среди программных систем моделирования этого типа можно указать системы автоматического доказательства теорем, например, PVS, и проверки моделей автоматов SPIN [3].

Рассмотрим применения таких систем автоматической проверки протоколов для систем, использующих параметризованные сообщения или состояния. При учете таких параметров происходит быстрый рост мощности множества состояний моделируемой системы, либо вовсе такое множество оказывается бесконечным. Такие случаи все же доступны для моделирования — с помощью выделения остова состояний и сообщений, которое обеспечивается устранением зависимостей от параметров [4].

Значительный рост количества компонент системы приводит к экспоненциальному росту мощности пространства состояний даже для упрощенных моделей взаимодействий, что было показано в работе [4]. Потому такие средства редко оказываются практически применимыми для децентрализованных систем большого масштаба.

Статистическое моделирование потоков данных

Имитация потоков данных может рассматривать следующие аспекты: стресс-тестирование для специфичных приложений, взаимное влияние алгоритмов управления насыщением для потоков ТСП/IP, экспериментальные среды передачи данных. Задачи моделирования коммутации пакетов в настоящий момент разработаны весьма детально, некоторые из них кратко описаны далее.

Так как реализация стека сетевых протоколов часто встроена в саму операционную систему (ОС), сравнительно редки задачи, в которых можно не учитывать специфику используемой ОС. В основном такие задачи рассматривают рабочую станцию на аппаратном уровне, вне контекста сетевой среды. В качестве примера приведем задачу оценки эффективности протокола поддержки согласованности совместно используемой памяти на уровне контроллеров памяти [5].

Проблема зависимости от ОС решается путем виртуализации – имитации аппаратной конфигурации системы, с исполнением кода ОС и конкретных приложений. Например, в сценариях системы М5 присутствует фаза инициализации, в которой выполняется настройка сетевых соединений и распределение данных в иерархиях оперативной памяти рабочих станций [6].

Укажем активно поддерживаемые системы, использующие виртуализацию: SimFlex [7] и GEMS [8]. Отметим некоторые негативные стороны виртуализации как подхода к моделированию:

- настройка среды и выполнение сценария часто дороже сбора статистики во время работы реальной системы;
- модели плохо масштабируются: при линейном расширении моделируемой системы, происходит степенное увеличение требуемого для моделирования машинного времени;
- виртуализация нового аппаратного компонента системы требует дополнительной разработки и тестирования среды виртуализации.

Следовательно, виртуализация используется редко, в основном при сложности воспроизведения сценария, дороговизне или недоступности реального оборудования.

В связи со сложностью масштабирования виртуализованных моделей некоторую часть системы заменяют статистически аналогичным генератором. Например, при виртуализации новой модели маршрутизатора возможна замена региональных сетей случайными генераторами потоков данных. Например, в работе [9] в этом контексте рассмотрены модели самоподобных потоков данных. Хорошее качество аппроксимации гарантируется в стационарном состоянии модели, в то время как реалистичное моделирование значительных структурных изменений системы (сбоев компонент, изменений маршрутов) редко оказывается возможным [10].

Чаще всего такие модели используются для оценки стратегий маршрутизации, коммутации и управления очередями в вычислительных сетях, что мы можем увидеть на примере развития комплекса моделей, систем моделирования, экспериментальных данных и средств визуализации ns-2 [11]. Модель системы при детализации на уровне передачи пакетов редко содержит более сотен узлов, но и в таком случае, в связи с вычислительной сложностью моделирования, исследователем рассматривается только опорная сеть маршрутизаторов.

Для децентрализованных систем большого масштаба статистическое моделирование применимо в ограниченной мере: дискретные события связываются с передачей сообщений прикладного уровня между узлами системы, и не проводится моделирование нижележащих уровней стека сетевых протоколов. Необходимость такого обобщения детализации для систем большого масштаба была показана в обзоре сред моделирования в работе [12].

Структурное моделирование вычислительных систем

Структурное моделирование и оптимизация вычислительных систем в основном подразумевает задачи получения и оценки сетевых структур, отображающих определенный уровень организации реальной системы.

Для получения сетевой структуры, выражающей некоторые характеристики вычислительной системы, можно использовать непосредственные данные: топологию Интернет на уровне автономных систем, статистику маршрутизации в MPLS-сети, листинги делегирования задач в кластерных вычислениях. После оценки статистических характеристик реальных сетей можно переходить к вероятностным методам генерации сетевых структур – моделям Барабаши-Альберта, Ердоса-Реньи и прочим [13].

Задачи формальной оценки структуры сети могут включать математическое моделирование и численную оценку определенного состояния вычислительной системы с заданной структурой. Например, можно реализовать алгоритм, вычисляющий процент близких к насыщению каналов связи в сети с заданной структурой каналов и трафика [14], либо

алгоритм, вычисляющий количество непересекающихся по соединениям маршрутов между узлами сети.

Так как децентрализованным системам присуща значительная степень абстрагирования ресурсов и служб конкретных узлов, то анализ структуры таких систем в целом возможно проводить без знания точных характеристик каждого узла. Покажем один из способов получения сетевой структуры децентрализованной системы большого масштаба.

Рассмотрим множество физических узлов вычислительной сети N_i , некоторые пары которых соединены физическими каналами связи $L_{i,j}$. При этом в качестве физической сети можно рассмотреть и сеть автономных систем определенного региона, или же множество узлов кластерной системы. Некоторые из физических узлов выполняют приложение, поддерживающее службы децентрализованной системы, то есть содержат логический узел децентрализованной системы из множества логических узлов V_i . Заметим, что иногда на одном физическом узле может размещаться несколько логических узлов.

С каждым логическим узлом связан идентификатор из множества K — строка, либо двоичная последовательность однозначным образом определяющая логический узел. Для объединения логических узлов в целостную структуру используются маршрутные таблицы, которые для каждого логического узла содержат идентификаторы и адреса логических узлов системы, связанных с текущим узлом.

Рассмотрим маршрутную таблицу каждого узла и представим каждую запись этой таблицы в виде дуги ориентированного графа. В этом орграфе роль вершин выполняют логические узлы V_i, V_j , а ребра $E_{i,j}$ соответствуют записям (ссылкам) в маршрутной таблице. По сути, такой граф является сетью связей и взаимодействий логических узлов.

Такую сетевую структуру называют оверлейной сетью, что буквально означает, что эта логическая сеть $\{V, E\}$ “наложена” на реальную аппаратную структуру $\{N, L\}$. В качестве примеров можно привести сеть маршрутизаторов, поддерживающих протокол BGP, сеть узлов пиринговой системы, сеть узлов кластера с динамической структурой.

Базисом децентрализованной системы является протокол поддержки оверлейной сети, обрабатывающий добавления, удаления узлов, сбои компонент [15,16]. На основе этого протокола реализуется функция поиска узла V_i в логической структуре по заданному ключу $k \in K$. Поиск в децентрализованной системе в общем случае сводится к рекурсивному приближению поискового запроса к искомому ключу в оверлейной сети.

Прикладные задачи в децентрализованной системе опираются на описанную функцию поиска, которая обеспечивает выполнение следующих задач: маршрутизацию поисковых запросов и удаленных вызовов, поддержку децентрализованной хеш-таблицы, доставку групповых сообщений. Следовательно, характеристики работы системы тесно связаны со свойствами структуры оверлейной сети.

Приведем несколько примеров такой связи: максимальная длина поискового маршрута очевидным образом связана с диаметром оверлейной

сети, размер маршрутной таблицы связан со степенью связности узла. Например, исследователи показали, что в эффективной системе эти величины должны быть ограничены сверху логарифмическими выражениями от количества узлов [15,16].

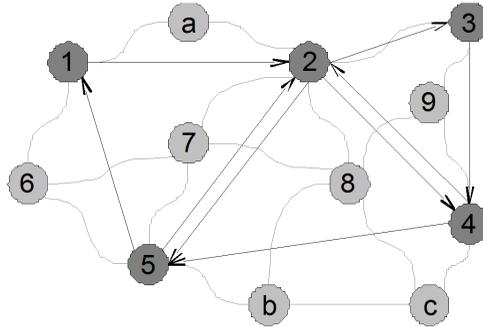


Рис. 1 – Упрощенный пример схемы организации оверлейной сети.

Например, узлы 1–5 (рис. 1) поддерживают распределенное приложение. Прямолинейные связи соответствуют логическим связям. Диаметр сети и максимальное количество ретрансляций поискового запроса равны 3 (запрос от узла 3 к узлу 1). Самая большая маршрутная таблица находится на узле 2, степень его связности (по исходящим связям) равна 3.

Следовательно, при увеличении масштаба моделируемой системы практическую важность приобретают статистические характеристики структуры оверлейной сети: распределение длин маршрутов, коэффициенты расширения, кластеризации, наличие компонент сильной связности, характер распределения степеней связности.

Таким образом, оценка структуры оверлейной сети методами анализа сложных сетей приобретает большое практическое значение, равно как и использование таких оценок в прикладных задачах. Приведем некоторые из таких методов, и соответствующих численных метрик, кратко описанных автором в работах [17,18].

Распределение степеней связности

Для каждой вершины степень связности задается выражением:

$$k_s = \sum_{(t \in V)} e_{st}, \tag{1}$$

где s, t – узлы логической сети, e_{st} – элементы матрицы связности (т.е. $e_{st} = 1 \Leftrightarrow s \sim t$). Для сетевых структур, основанных на последовательном развитии и преимущественном связывании, распределение степеней связности задается вероятностным распределением:

$$P(k) \sim k^{-\gamma}, \quad (2)$$

при этом для многих классов реальных сетей были получены оценки степени $-1 \geq \gamma \geq -2$. При увеличении γ возрастает вероятность образования ядра сильной связности — что увеличивает эффективность поиска и широковещательной передачи информации в сети, но делает ее более уязвимой для управляемой атаки на узлы.

Эффективность ретрансляций

Можно провести аналогию с метрикой глобальной эффективности, известной для географических моделей [18]:

$$\hat{E} = \frac{1}{|E|} \sum_{s,t:(s,t) \in E} \frac{1}{d_{s,t}}, \quad (3)$$

где $d_{s,t}$ равно количеству передач запроса в физической сети между узлами оверлейной сети s, t . Уменьшение этой метрики соответствует низкой эффективности структуры оверлея с точки зрения ретрансляций сообщения (или пакета) в физической сети.

Кластеризация

Кластеризация в самом простом виде представляет собой соотношение количества 3-циклов к количеству связей в сети:

$$C = \frac{3 \sum_{i < j < k} e_{ij} e_{ik} a_{jk}}{\sum_{i < j < k} (e_{ij} e_{ik} + e_{ji} e_{jk} + e_{ki} e_{kj})} = 3 \frac{N_{\Delta}}{N_3}. \quad (4)$$

Сети с высоким уровнем кластеризации часто содержат несколько слабо связанных между собой компонент со сравнительно большей внутренней связностью. Также высокие значения присущи звездообразным структурам с ядром высокой связности и изолированной периферией.

Спектр матрицы Лапласа

Матрица Лапласа имеет вид:

$$L = D - E, \quad (5)$$

где D — диагональная матрица степеней связности: $d_{ii} = \sum_j e_{ij}$. Спектр матрицы Лапласа (упорядоченная по возрастанию последовательность собственных чисел) позволяет оценить связность сети, степень близости структуры сети к двудольному графу, наличие циклов определенной длины.

В особенности интересна связь второго собственного числа с минимальным сечением сети. Под сечением понимается набор ребер, делящий сеть на изолированные компоненты. Точно вычислить весь значения спектра сложной сети чаще всего невозможно, однако существуют методы, позволяющие оценить с приемлемой точностью первые его элементы.

Практический пример

Приведем краткий практический пример: некая организация поддерживает некоторое число вычислительных сетей региональных отделений, расположенных в различных автономных системах сети интернет. Этой организации необходимо обеспечить прозрачную синхронизацию данных между рабочими станциями региональных отделений. При росте количества запросов, логическая структура сложной сети потоковых передач оказывает все более значительное влияние на эффективность работы системы. Например, если алгоритмы управления структурой оверлейной сети не учитывают степень ее кластеризации (4), то вероятно возникновение компонент сильной связности, которое приведет к значительному падению производительности системы на узлах, связующих кластера.

Выводы

Как было показано в работах [12,14], оценка показателей работы проектируемых распределенных систем часто проводится с помощью сред имитационного моделирования. Большой масштаб децентрализованных вычислительных систем накладывает определенные ограничения на подходы к моделированию. Как было показано в работе, моделирование децентрализованной системы как множества конечных автоматов или же моделирование дискретных событий на уровне коммутации пакетов для таких масштабов может играть только вспомогательную роль.

Децентрализованная природа и большой масштаб системы позволяет обобщить характеристики отдельных компонент и перейти к рассмотрению статистических характеристик логической структуры сети. Таким образом, применение методов оценки сложных сетей является актуальной и новой задачей при разработке децентрализованных вычислительных систем.

Актуальным направлением дальнейших исследований автор считает разработку алгоритмов типичных служб, учитывающих статистические характеристики структуры оверлейной сети для улучшения показателя работы децентрализованной системы.

Литература

1. Тимошенко Ю. Краевий А. Огляд і класифікація програмно-апаратних засобів побудови розподілених обчислювальних систем // Наукові вісті НТУУ „КПІ”. — 2006. — 2. — 13 с.
2. Vlaovic B. Vree A. Brezovcnik Z. et al. Automated generation of Promela model from SDL specification // Comput. Stand. Interfaces Journal. — Elsevier Science Publishers B. V., Amsterdam. — 2007. — Vol. 29, No. 4. — P. 449-461. — ISSN 0920-5489.
3. Holzmann G. J. The Model Checker SPIN // IEEE Trans. Softw. Eng. — IEEE Press, Pictaway. — 1997. — Vol. 23, No. 5. — P. 279-295. — ISSN: 0098-5589

4. Havelund K. Shankar N. Experiments in theorem proving and model checking for protocol verification // FME '96: Proc. of the third intern. symp. of formal methods, Europe on industrial benefit and advances in formal methods. – Springer-Verlag, London. – 1996. – pp. 662-681.
5. Archibald J. Baer J.-L. Cache coherence protocols: evaluation using a multiprocessor simulation model // ACM Trans. Comput. Syst. – ACM Press, New York. – 1986. – Vol. 4, No. 4. – P. 273-298. – ISSN 0734-2071.
6. Binkert N. L. Dreslinski R. G. et al. The M5 Simulator: Modeling Networked Systems // IEEE Micro. – IEEE Computer Society Press, Los Alamitos. – 2006. – Vol. 26, No. 4. – P. 56-60. – ISSN 0272-1732.
7. Hardavellas N. Somogyi S. et al. SimFlex: a fast, accurate, flexible full-system simulation framework for performance evaluation of server architecture // SIGMETRICS Perform. Eval. Rev. – ACM Press, New York. – 2004. – Vol. 31, No. 4. – P. 31-34. – ISSN 0163-5999.
8. Martin M. M. K. Sorin D. J. Beckmann B. M. et al. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset // SIGARCH Comput. Archit. News. – ACM Press, New York. – 2005. – Vol. 33, No. 4. – P. 92-99. – ISSN 0163-5964.
9. Paxson V. Fast approximate synthesis of fractional Gaussian noise for generating self-similar network traffic // SIGCOMM Comput. Commun. Rev. – Vol. 27, No. 5. – 1997. – pp. 5-18. – ISSN: 0146-4833.
10. Paxson V. Floyd S. Why we don't know how to simulate the Internet // WSC '97: Proceedings of the 29th conference on Winter simulation, Atlanta. – IEEE Computer Society, Washington. – 1997. – P. 1037-1044 – ISBN 0-7803-4278-X.
11. Lucio G. F. Paredes-Farrera M. Jammeh E. et al. Opnet modeler and ns-2: Comparing the accuracy of network simulators for packet-level analysis using a network testbed // In Proc. of 3rd WSEAS Int. Conf. on Simulation, Modeling and Optimization, Rethymno, Greece. – World Scientific and Engineering Academy and Society (WSEAS), Stevens Point. – October 2003 – No. 3, Vol. 2. – P. 700-707. – ISSN: 1109-2750.
12. Naicken S. Livingston B. Basu A. et al. The state of peer-to-peer simulators and simulations // SIGCOMM Comput. Commun. Rev. – ACM Press, New York. – Vol. 37, No 2. – April 2007. – P. 95-98.
13. Albert R.S. Statistical mechanics of complex networks // Ph.D. thesis, dir. Barabasi A.L. – University of Notre Dame, Notre Dame. – 2001. – ISBN 0-493-08502-5.
14. Rahman M.A. Pakstas A. Wang FZ. Network Modelling and Simulation Tools // PGNET 2007 The 8th Annual Postgraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting (ed. Merabti M.). – John Moores University, Liverpool. – 2007.

15. Stoica I. Morris R. Liben-Nowell D. et al. Chord: a scalable peer-to-peer lookup protocol for internet applications. // IEEE/ACM Transactions on Networking. – Feb. 2003. – IEEE Press, Piscataway: – Vol. 11, No 1. – P: 17 – 32. – ISSN 1063-6692.
16. Aberer K. Cudr'e-Mauroux Ph. Datta A. et al. P-Grid: a self-organizing structured P2P system // SIGMOD Rec. – ACM Press, New York – 2003. – Vol. 32, No. 3. – P. 29-33. – ISSN 0163-5808.
17. Краевой А. Способи оцінки моделей обчислювальних мереж // Управління розвитком: Тезиси міжнарод. научн.-практ. конф. “Стратегії ІТ-технологій в освіті, економіці та екології” (15-16 листопада 2007 р.). – 10. – Харків. – Харківський нац. екон. ун-т. – стр. 22-23.
18. Краевой А. Підходи в оцінці стійкості мережових структур // Тезиси 26-ої научн.-техн. конф. “Моделювання” 12-13 лют. 2007, стр. 40-42

Отримано 20.03.2009 р.