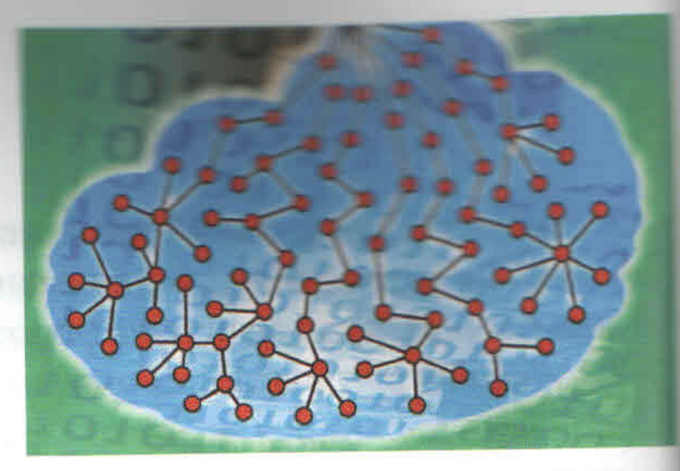

Интернет в сетях больших графов

В Интернете вообще и в социальных сетях в частности потенциально имеется много полезных сведений, для выявления которых невозможно обойтись без технологий обработки больших графов.

Ключевые слова: социальные сети, граф
Keywords: MapReduce, social networks, graph, multithreading

Александр Фролов,
Александр Семенов

Анализ больших графов с количеством вершин от сотен миллионов до нескольких триллионов и количеством ребер, на несколько порядков превышающим число вершин, представляет собой крайне сложную научно-техническую задачу, для решения которой требуются особые технологии. Графовые задачи отличаются тем, что обрабатываемые данные, как правило, имеют нерегулярную структуру и порядок вычислений над ними определяется самими данными — топологией графа, а операции доступа к данным преобладают над вычислениями. Кроме того, большинство графов реального мира — социальные сети, Интернет, веб-графы, структуры биоинформатики и т. п. — подчиняются степенному закону распределения степеней вершин, что приводит к сильной асимметрии функции распределения в направлении к вершинам с малыми степенями и выделению небольшого количества вершин с большим числом соседей. Например, в семантической базе данных DBPedia, построенной на основе данных, извлеченных из Wikipedia, количество вершин, имеющих менее пяти соседей, составляет 90%, в то время как имеется несколько вершин с количеством соседей, превышающим 100 тыс. Такое распределение может привести к несбалансированной нагрузке на узлы вычислительной системы, при этом итоговое время выполнения задачи будет определяться временем работы наиболее загруженных узлов.

Для графовых приложений характерны низкая пространственно-временная локализация обращений к памяти вычислительных узлов, интенсивный поток коротких сообщений по коммуникационной сети с паттерном, близким к «все ко всем», и несбалансированность вычислений. На решение этих и других проблем параллельной обработки графов направлены сегодня основные усилия исследователей, использующих различные программные модели параллельной обработки больших графов, например в задаче поиска кратчайших путей из одной вершины до всех достижимых вершин (задача Single Source Shortest Paths, SSSP). SSSP используется во многих реальных приложениях — например, при построении маршрутов в Интернете, в системах автоматизированного проектирования, робототехнике, логистике, при анализе социальных сетей, построении маршрутов в транспортных сетях и т. п.

Можно выделить пять основных подходов к параллельной обработке графов: параллельные СУБД, MapReduce, модели BSP, модели Data-Flow и многопоточные модели вычислений с общей памятью с глобальным адресным пространством (MT/PGAS). Возможны также гибридные подходы. Следует отметить, что, несмотря на бурное развитие таких технологий, как поколоночное хранение, обработка в памяти и др., для анализа больших графов параллельные СУБД используются крайне редко из-за их низкой эффективности и слабой масштабируемости.

MAPREDUCE

Появление реализаций модели MapReduce, в частности Hadoop, позволило обрабатывать большие графы с использованием технологий анализа Больших Данных. В рамках парадигмы MapReduce прикладной программист представляет задачу в виде двух стадий: Map (распределение задач по дочерним узлам) и Reduce (сборка и вычисление результата решения задачи) — а затем реализует их в виде функций. За остальные действия (передачу сообщений, объединение сообщений с одинаковым ключом, чтение и запись данных из распределенной файловой системы) отвечает программная реализация MapReduce. Количество процессов, выполняющих Map и Reduce, задается пользователем при запуске программы, причем их соотношение может быть произвольным, хотя на практике рекомендуют 2:1.

Использование модели MapReduce для обработки больших графов достаточно удобно в силу простоты программной модели и естественного отображения на нее абстракции графа. С другой стороны, применение MapReduce для обработки графов целесообразно только в случаях, когда не требуется высокая производительность. Тем не менее существенным преимуществом MapReduce является прозрачное обеспечение отказоустойчивости системы за счет сохранения состояния графа между итерациями выполнения алгоритма.

Удобство, надежность и простота использования модели MapReduce для анализа графов, а также наличие открытой реализации (Hadoop) позволили этой модели стать одним из наиболее привлекательных средств для анализа больших графов. Сегодня создано множество расширений MapReduce как общего назначения (например, Dryad, Stratosphere, Haloop, Yarn и др.), так и ориентированных непосредственно на графовую обработку (Pegasus, Surfer, GBASE, GraphX и Spark). Также стоит отметить появление реализаций MapReduce с использованием MPI (например, библиотека MR-MPI), предназначенных для выполнения на параллельных системах.

Основной особенностью модели Bulk Synchronous Parallel (BSP) и ее известной адаптации к графовым задачам — модели Pregel — является представление вычислений в виде последовательности итераций («супершагов»), на каждой из которых параллельно выполняются вычисления. Обмен сообщениями происходит между итерациями — сообщения, отправленные в течение i-й итерации, будут доставлены адресатам на итерации i+1. Таким образом, гарантируется отсутствие дедлоков во время выполнения вычислений, однако после завершения каждой итерации требуется выполнение глобальной барьерной синхронизации. Как и в случае MapReduce, модель BSP позволяет легко организовать фиксацию контрольной точки за счет сохранения пользовательских данных после выполнения барьерной синхронизации.

На основе BSP в Google была разработана программная модель Pregel, ориентированная на параллельную обработку больших графов. Кроме BSP в основе Pregel лежит подход, в котором вычислительный процесс представляется изнутри вершины графа, то есть программа на Pregel — это описание программы функционирования конечного автомата с двумя состояниями («активен», «пассивен»), набором входных данных в виде сообщений, поступающих по входящим дугам из соседних вершин, и генерируемыми выходными данными в виде сообщений, передаваемых по исходящим дугам соседним вершинам. При получении входящего сообщения вершина активируется. Выполнение программы заканчивается, когда в графе не останется ни одной активной вершины.

Появилось множество исследований по реализации и развитию Pregel для различных платформ. Одной из наиболее известных является библиотека Apache Giraph, используемая в Facebook, LinkedIn и Yahoo!. Среди других реализаций модели Pregel можно упомянуть Apache Hama, SignalCollect, PregelX. Особо стоит отметить библиотеку GraphLab, разработчики которой расширили Pregel поддержкой асинхронного режима выполнения задач, что по сути является отходом от принципов BSP.

Подход Data-Flow к параллельной обработке больших графов объединяет в себе программные модели, в основе которых лежит принцип управления потоком данных. К таким средствам можно отнести язык программирования Charm++, систему параллельного выполнения программ SWARM, а также библиотеку AM++ (Active Pebbles).

Charm++ — параллельное расширение языка C++, реализующее модель вычислений с мелкозернистым, асинхронным параллелизмом и управлением потоком сообщений. Программа на Charm++ состоит из множества объектов, распределенных по узлам вычислительной системы (в терминах Charm++ такие объекты называются «chare»). Для объектов определены интерфейсные функции (entry-методы), которые можно вызывать вне зависимости от того, где находится объект (на локальном или удаленном вычислительном узле). Пользователь может создавать на одном узле десятки тысяч chare-объектов, что позволяет получить достаточно параллельных процессов для запуска на процессорных ядрах. Одновременно у одного chare-объекта может выполняться только один вызов entry-метода, что гарантирует атомарность изменения данных, находящихся внутри объекта.

Принципиальным отличием Charm++ от Pregel является асинхронность выполнения программы. В случае Pregel программа выполняется синхронно и обновление вершин происходит итеративно с выполнением барьерной синхронизации между итерациями, а в Charm++ — асинхронно, что потенциально должно обеспечить более высокую производительность. Charm++ — это университетский проект, и в коммерческих компаниях он пока не применяется.

Еще одним направлением развития технологий обработки больших графов является применение распределенной общей памяти и массово-мультитредовой вычислительной модели. Идеологически данный подход унаследовал многое от принципов работы суперкомпьютера Tera MTA [1], разработанного в начале 90-х годов. В первую очередь — использование массово-тредового параллелизма для достижения толерантности к задержкам обращений к памяти и передачи сообщений по сети, а также общей памяти с глобальным адресным пространством.

Представителями данного подхода являются модели Stinger, GraphCT и Grappa (SoftXMT). Основными предпосылками появления подобных решений стали следующие архитектурные особенности современных параллельных систем. Во-первых, микропроцессоры способны выполнять сотни тысяч операций в каждом ядре за время выполнения одного обращения к удаленной памяти, что позволяет использовать технику легковесных программных тредов («корутин») с минимальным контекстом, которые запускаются в ядре на фоне выполнения выданных ранее обращений к памяти. Таким образом достигается совмещение вычислений и доступа к памяти (локальной и удаленной), а также необходимое количество операций обращений к памяти для насыщения пропускных способностей памяти и сети.

Во-вторых, современные коммуникационные сети имеют относительно низкий темп передачи коротких сообщений и высокую пропускную способность при передаче длинных, что делает возможным программно агрегировать короткие сообщения, порождаемые обращениями к удаленной памяти, и достичь высокого коэффициента полезной загрузки линков сети и высокого темпа выдачи коротких сообщений, упакованных в длинные сообщения.

Для приложений с интенсивным потоком обращений к распределенным массивам данных, вызывающих интенсивный обмен короткими сообщениями между узлами коммуникационной сети, получение высокой реальной производительности требует толерантности к задержкам передачи сообщений по сети за счет совмещения коммуникаций и вычислений, а также агрегации коротких сообщений.

Всеми описанными свойствами обладает исследовательская система Grappa, представляющая собой фреймворк для решения задач с интенсивной нерегулярной работой с памятью, основанный на массово-мультитредовой вычислительной модели и распределенной общей памяти. Работы над Grappa ведутся в Вашингтонском университете с 2011 года. Разрабатывается Grappa на языке C++11, и приложения, использующие Grappa, также должны быть написаны на нем. Для реализации коммуникаций используется GASNet — библиотека с поддержкой односторонних коммуникаций и активных сообщений.

***

Ценность данных, содержащихся в больших графах, стимулирует проведение исследований в области технологии их обработки — например, в США, Японии и Китае такие работы финансируются через специальные целевые федеральные программы (DARPA UHPC/PERFECT, GraphCREST). Для параллельной обработки больших графов могут применяться программные модели MapReduce, Pregel, Charm++ и Grappa, используемые для работы как в частных облаках, так и на высокопроизводительных кластерных системах. Однако все эти модели предоставляют достаточно «низкий уровень» абстракции с точки зрения конечного пользователя, поэтому сегодня актуальны работы по созданию специализированных языков программирования для решения графовых задач и их реализации поверх моделей.

Можно предположить, что будущее графовых технологий будет определяться, с одной стороны, увеличением объемов обрабатываемых данных и производительностью, а с другой — повышением их результативности и доступности, что позволит ускорить их внедрение в разных прикладных областях. ■

ЛИТЕРАТУРА

1. Александр Семенов, Александр Фролов, Анатолий Никитин, Владимир Кабыкин. Суперкомпьютеры для графовых задач // Открытые системы.СУБД. — 2011. — № 7. — С. 38–43. URL:http://www.osp.ru/os/2011/07/13010498 (дата обращения 18.06.2014).

Александр Фролов (frolov@nicevt.ru), Александр Семенов (semenov@nicevt.ru) — сотрудники компании «НИЦЭВТ» (Москва).