

ПАРАДИГМЫ РАСПРЕДЕЛЕННЫХ СИСТЕМ

Э.Н.ИСРАФИЛОВА

В работе рассматривается одна из задач, в основе которой лежит проблема синхронизации процессов, а именно, задача обеспечения сериализуемости параллельно выполняемых транзакций в распределенных системах. Предлагается метод, предназначенный для предотвращения образования цикла в D-графе на этапе предварительного анализа распределенного плана подтранзакций. Предложенный алгоритм поиска цикла основан в дискретизации процесса инициации транзакций и построении в каждом узле дерева зависимостей между подтранзакциями, которое в процессе роста определяет наличие или отсутствие цикла в D-графе.

Введение. Проблемы информатики связаны с очень широким кругом вопросов. Одной из таких проблем в процессе информатизации нашего общества является создание компьютерных сетей и построение на их основе распределенных систем обработки информации или просто распределенных систем (РС). Основными требованиями при создании РС являются оперативность доступа к ресурсам и обмена информацией, надежность функционирования всех компонентов и системы в целом, обеспечение достоверности информационных ресурсов, эффективность использования технических средств и т.п. Хотя распределенная обработка стала самостоятельным направлением в развитии информатики и вычислительной техники, но концептуально и содержательно ее предмет и задачи очень часто связываются с конкретными приложениями, такими, как распределенные базы данных, электронная почта, дистанционная работа с файлами и т.п.

Конечной целью создания распределенных систем обработки информации является интеграция информационных и вычислительных ресурсов, а также средств коммуникации и оргтехники и т.п. целого региона и даже мира для своевременного представления возможностей их потребления не зависимо от географического расположения пользователей.

Синхронизация прикладных процессов. Одной из проблем, возникающих в теории и практике РС, функционирующих в сетевой среде, является синхронизация процессов, протекающих в узлах РС. В [1] приведена идея создания универсального ядра синхронизационных механизмов, так называемого синхронизационного автомата (state mashine for

synchronization). Эта идея носит настолько общий характер, что возведена в ранг парадигмы РС. Упомянем, что существуют еще две парадигмы – интерактивное соглашение (interactive consistency) и глобальное состояние (global state). Все три парадигмы тесно связаны друг с другом. На самом деле, с одной стороны, без синхронизационных механизмов невозможно выполнение множества задач в узлах РС, в частности задачи, определения глобального состояния РС. С другой стороны, при параллельном выполнении прикладных процессов, а также при определении глобального состояния РС необходимо, чтобы все узлы достигли консенсуса, т.е. взаимного согласия по общему решению.

Однако в настоящей работе рассматривается только одна парадигма РС - синхронизационный механизм. К настоящему времени теоретические проблемы, и вопросы реализации синхронизационных процессов в РС исследуются многими учеными и являются актуальными. Синхронизация необходима процессам для организации совместного использования ресурсов. Синхронизационные механизмы – неотъемлемая часть задач управления РС. Без таких механизмов невозможно выполнение множества подзадач в узлах РС. В однопроцессорных системах решение задач синхронизации осуществлялось с использованием общих методов, таких как семафоры и мониторы. Эти методы не совсем подходят для РС, так как все они базируются на использовании разделяемой оперативной памяти. Однако в силу специфичности РС распространить эти методы непосредственно на эти системы невозможно. Дело в том, что процессы в различных узлах РС передают сообщения друг другу только через коммуникационную среду, связывающую узлы между собой. При этом возможны существенные задержки доставки сообщений к узлу-адресату. Информация, посланная в сообщении, может устареть, т.е. не отражать состояние процесса, пославшего это сообщение. Учитывая, что в РС таких процессов достаточно много, и что в любой момент времени возможны отказы процессов в узлах РС, необходимы новые механизмы синхронизации. Синхронизация понимается как скоординированное корректное управление процессами, протекающими в РС как последовательно, так и параллельно, и конкурирующими между собой за использование некоторого ресурса. Задача взаимного исключения (критической секции) при выделении ресурса, задача двухфазной фиксации транзакций, задача обеспечения сериализуемости распределенного плана выполнения транзакций – вот далеко не полный перечень частных задач, в основе которых лежит проблема синхронизации процессов в РС.

В работе рассматривается одна из задач, в основе которой лежит проблема синхронизации процессов, а именно, задача обеспечения сериализуемости параллельно выполняемых транзакций в РС. Эта проблема обсуждается в литературе в течение последних лет. И сейчас эта проблема продолжает интересовать исследователей РС как в теоретическом, так и в практическом планах.

Примем следующую модель РС, реализующую транзактную обработку. В множестве узлов $S = \{S_i / i = \overline{1, n}\}$ в каждый момент времени функционируют два подмножества:

- ТМ – подмножество узлов, иницирующих работы (транзакции);
- ДМ – подмножество узлов, выполняющих (обрабатывающих) транзакции.

Ввиду случайности и независимости процесса инициации транзакций узлами ТМ в узлах ДМ могут образовываться очереди подтранзакций, ждущих обработки. Совокупность всех очередей назовем распределенным планом выполнения подтранзакций. Координация параллельной обработки транзакций в рамках всей РС проводится системой управления транзакциями (СУТ). Основным требованием, которое предъявляется к СУТ, является обеспечение сериализуемости транзакций.

Проблема сериализуемости возникает вследствие возникновения цикла в, так называемом, графе зависимостей (D -графе). D -граф является ориентированным мультиграфом $D = \{U, V\}$, здесь U -множество вершин, соответствующих подмножеству активных транзакций $T_i (i = \overline{1, n_{TM}})$, иницированных в узлах-инициаторах транзакций $\{TM_i\} \subseteq \{S\}$, V -множество дуг, соответствующих отношению предшествования, существующего между подтранзакциями T_i^k , стоящими в очередях к ресурсу в узлах обработки $\{DM^k\} \subseteq \{S\}$. Возникновение цикла в D -графе может являться причиной нарушения целостности РБД и возникновения тупиковых ситуаций.

Алгоритм. Ниже предлагается метод, предназначенный для предотвращения образования указанного цикла на этапе предварительного анализа распределенного плана подтранзакций. Этот метод основан на идее включения в СУТ механизма прямой проверки некоторого множества подтранзакций на отсутствие цикла в соответствующем этому множеству D -графе. Назовем это множество испытываемым. Если такой цикл отсутствует и возникновение его в будущем также невозможно, проверенное множество считается ратифицированным и подтранзакции, принадлежащие этому множеству, могут выполняться в том порядке, в котором они стоят в очередях. При наличии цикла одна из транзакций, инцидентная ему, откатывается и через некоторое время стартует вновь. Алгоритмы поиска цикла, точнее принятие решения о том, существует цикл или нет, могут быть различными. Предложенный алгоритм основан в дискретизации процесса инициации транзакций и построении в каждом ТМ дерева зависимостей между подтранзакциями, которое в процессе роста определяет наличие или отсутствие цикла в D -графе.

Процесс инициации дискретизируется для того, чтобы циклы в D -графе не образовывались за конечное время. Дискретизация заключается в следующем. Временная ось всех ТМ делится на равные отрезки ΔT , называемые интервалами дискретности. Начальные отрезки каждого интервала дискретности Δt называются интервалами инициации. Величина Δt подбирается эмпирически и должна удовлетворять неравенству

$$\Delta T \geq 2T_{dist}^{\max} + \Delta t + \delta^{\max} \quad (1)$$

где T_{dist}^{\max} - максимальное время доставки сообщения из одного узла в другой; δ^{\max} - максимальное время обработки управляющего сообщения. Транзакции иницируются только в течение интервалов Δt .

Алгоритм заключается в обмене последовательностями сообщений между узлами, иницировавшими транзакцию, и соответствующими узлами, в которые посылаются подтранзакции, а также узлами, которые конкурируют с данным узлом. Два узла называются конкурирующими, если имеются очереди, в которых находятся подтранзакции, посланные каждым из этих узлов. Обмен сообщениями длится в общем случае в течение нескольких интервалов дискретности (минимум одного). Алгоритм для множества узлов ТМ, иницировавших транзакции в одном и том же интервале инициации, заканчивается, если все транзакции, инициированные в этом интервале, либо ратифицированы, либо ликвидированы в процессе отката.

Множество интервалов дискретности, в течение которых продолжается алгоритм, называется циклом управления. Параллельно со сдвигом не менее чем на один интервал дискретности могут выполняться два и более алгоритмов, при этом множества ТМ для таких параллельных алгоритмов не пересекаются.

Первый интервал дискретности. Рассмотрим некоторый интервал дискретности и обозначим его ΔT_1 , т.е. интервал первый в цикле управления. Множество узлов ТМ иницирует транзакции в течение соответствующего интервала инициации Δt_1 . Инициация транзакции узлом ТМ производится командами типа start. Узлы DM получившие команду start, продлевают очереди к ресурсу, сформировавшиеся у них до этого времени, записями соответствующими вновь поступившим подтранзакциям. Кроме того, узлы DM посылают в узлы ТМ, ответные сообщения acks, содержащие информацию о том, за какой подтранзакцией установлена в очереди подтранзакция, присланная командой start. Перед концом интервала ΔT_1 соответствующая транзакция помечается как принадлежащая испытываемому множеству.

Алгоритм, выполняемый в интервале ΔT_1 , заканчивается тем, что все узлы ТМ получают соответствующие сообщения acks. То, что эти ко-

манды дойдут в течение интервала ΔT_1 до соответствующих адресатов, гарантировано ограничением (1).

Получив все команды asks, узлы ТМ строят у себя начальное дерево высотой $L \leq 1$. Корневая вершина соответствует самому ТМ, а остальные вершины узлам ТМ, подтранзакции которых оказались в очередях непосредственно перед подтранзакциями, присланными узлом ТМ. Если некоторая подтранзакция от данного ТМ установлена либо первой, либо за уже ратифицированной подтранзакцией, либо принадлежит множеству испытываемых в предыдущих циклах управления, соответствующая ветвь не строится, а корневая вершина считается завершенной и надчеркивается.

Последующие интервалы дискретности. Каждый узел ТМ, инициировавший транзакцию, запрашивает у других узлов ТМ, соответствующих висящим вершинам дерева, информацию о деревьях, построенных на предыдущем интервале дискретности. Получив эту информацию, узел ТМ развивает свое дерево, склеивая корневую вершину дерева, полученного из узла ТМ, с соответствующей ей незавершенной висячей вершиной дерева узла ТМ. Когда дерево узла ТМ развилось, проводится его анализ на завершенность корневой вершины и наличие вершин, одноименных с корневой, и соответствующее преобразование.

Корневая вершина считается завершенной, если все дуги, исходящие из нее, соединяют ее с завершенной вершиной. В противном случае корневая вершина считается незавершенной и дуги, соединяющие ее с завершенными вершинами, не строятся.

Параллельно дерево проверяется на наличие вершин, одноименных с корневой. Если такая вершина обнаружена, это означает, что в D -графе имеется цикл.

Алгоритм для некоторого узла ТМ заканчивается, если:

- Корневая вершина дерева оказалась завершенной, транзакция получает право быть ратифицированной, об этом уведомляются соответствующие узлы DM командами ratificate;
- Обнаружен цикл в D -графе, транзакция, имеющая наименьшее значение индекса i_{\min} в множестве транзакций и инцидентная циклу, откатывается, об этом уведомляются соответствующие узлы DM командами abort. Дерево в узле $TM_{i_{\min}}$ уничтожается. В узлах $TM_j (j \neq i_{\min})$ деревья преобразуются указанным выше способом.

Предложенный алгоритм обобщает следующее утверждение.

Утверждение. Алгоритм результативен за k интервалов дискретности, k находится из неравенства

$$k > \left\lceil \frac{\ln n_{TM}}{\ln 2} \right\rceil + 1$$

где n_{TM} - число узлов типа ТМ; $[]$ - ближайшее большее целое.

ЛИТЕРАТУРА

1. Schneider F.B. Paradigms for distributed programs // Lecture Notes in Comput. Sci. – 1985. – N190. – p.431-480.

PAYLANMIŞ SİSTEMLƏRİN PARADİQMALARI

E.N.İSRAFİLOVA

ANNOTASIYA

İşdə paylanmış sistemlərdə proseslərin sinxronlaşdırılması problemi məsələsi kimi paralel yerinə yetirilən tranzaksiyaların seriallığının təmin edilməsi məsələsinə baxılır. Tranzaksiyaların paylanmış planının ilkin analizi mərhələsində D-qrafda dövrün əmələ gəlməsinin qarşısının alınması üçün üsul təklif edilir. Dövrün axtarışı üçün təklif edilmiş alqoritm tranzaksiyaların inisiyasiya prosesinin diskretləşməsi və alttranzaksiyalar arasında artma prosesində D-qrafda dövrün olub olmamasını təyin edən asılılıq ağacının qurulmasına əsaslanır.

PARADIGMS OF DISTRIBUTED SYSTEMS

E.N.ISRAFILOVA

ABSTRACT

In work one of tasks is considered, in which basis the problem of synchronization of processes, namely, task of maintenance serializability in parallel executed out transaction in distributed systems. The method intended for prevention of formation of a cycle in D-graph at a stage of the preliminary analysis of the distributed schedule subtransactions is proposed. The proposed of search of a cycle algorithm is based in discretization of transaction initiation process and construction in each unit of a tree of dependences between subtransactions, which during growth defines presence or absence of a cycle in D- graph.