

## ПОСТРОЕНИЕ РАСПРЕДЕЛЕННЫХ ПРИЛОЖЕНИЙ

Исследованы средства для построения распределенных приложений, доступных в среде Lotus Domino. Предлагается подход к построению алгоритмов параллельной обработки документов на основе одного ведущего с нескольких ведомых процессов с использованием кластеров Lotus Domino.

Распределенные вычисления сегодня затрагивают не только область математического моделирования физических и химических процессов, но и входят в область социальной сферы, экономики и бизнес-приложений.

При этом такого рода вычисления могут происходить на разных уровнях вычислительных систем – от симметричной мультипроцессорности до кластеров прикладного уровня. Вычислители в этом спектре систем могут находиться на различных расстояниях – от сантиметров до десятков и даже сотен километров.

Цель данной работы – выявить основные методы разработки алгоритма параллельных кластерных вычислений на уровне приложений с использованием соответствующих средств в клиент-серверном продукте для групповой работы Lotus Domino.

### ТЕХНОЛОГИИ РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ

1. Встроенная архитектура «клиент-сервер». Эта технология является основой для взаимодействия клиента и сервера, она существовала с момента появления многопользовательской версии. Распределение вычислений между клиентом и сервером здесь управляется настройками построения индексов: при сохранении клиентом данных в базе сервер выполняет действия по отбору документов и помещению документа в индекс «на свое место», в зависимости от критериев отбора вида и параметров сортировки значений колонок. Кроме того, в системе есть средства запуска процедур (агентов) на сервере непосредственно с клиента. Это средство обладает одной особенностью: при запуске процедуры на сервере клиентская станция не отключается, а «ожидает» завершения вычислений агентом, таким образом, время выполнения вычислений уменьшается, но клиент, тем не менее, «занят» вычислениями.

2. Интерфейс с CORBA для взаимодействия с другими приложениями по протоколу ORB. В Lotus Domino есть набор классов, позволяющий выполнять запуск объектов Domino другими приложениями, а также обращаться к объектам других приложений, поддерживающий архитектуру CORBA. Построение единой иерархии классов, отражающей функции подразделения или предприятия в целом, возможно вне зависимости от того, какие именно приложения разработаны и установлены в организации. Распределение вычислений между приложениями зависит от того, каким образом построена иерархия классов, какие она охватывает приложения и в какой степени. Например, вычисление свойства объекта одного класса может повлечь инициализацию объекта другого класса в другом приложении, находящемся на другой вычислительной установке.

3. Поддержка виртуальной машины Java и сервлетов. С развитием интернет, языков разметки страниц и Java многие производители СУБД включили в свои продукты поддержку протокола HTTP и виртуальной машины Java, что позволяет разрабатывать на Java апплеты, выполняющиеся на сервере, способные обращаться к ис-

точникам данных в других СУБД по запросу клиента из браузера. Распределение вычислений здесь зависит от того, насколько развита информационная инфраструктура в организации, какие СУБД установлены и какие интерфейсы для связи используются.

4. Кластерные технологии. Дают возможность построить отказоустойчивую систему серверов. Lotus Domino имеет одним из свойств кроссплатформенность, что позволяет построить кластер на уровне приложений (баз данных Domino) с использованием нескольких аппаратно-программных платформ. Распределение вычислений происходит в пределах одной СУБД и управляется настройками отказоустойчивости. Отказоустойчивость и масштабируемость вместе являются родителями нового свойства – сбалансированная нагрузка. Поскольку в кластерной системе по определению присутствует несколько вычислителей, то вполне реально говорить о балансировке нагрузки между вычислителями. Эта задача решается введением некоего виртуального индекса – «индекса доступности» (availability index). Чем выше индекс доступности, тем вероятнее отклик у данного сервера. Все члены кластера начинают работу со значения индекса, равного 100. По мере загрузки индекс может уменьшаться или увеличиваться в зависимости от количества проводимых одновременно транзакций и числа подключенных пользователей. Кроме статусов «доступен» или «недоступен» при балансировании нагрузки сервер также может принимать значение «занят», означающее, что очередной запрос от клиента не будет принят в случае, если в кластере есть доступный сервер. Если же доступного сервера нет, то запрос от клиента к занятому серверу все-таки пройдет, чтобы не нарушить принципа отказоустойчивости.

При организации работы или построения системы с распределенными вычислениями необходимо учитывать свойства таких систем.

### ПАРАЛЛЕЛИЗМ

Параллелизм – это свойство, характеризующее объект, способный выполнять несколько действий одновременно. При этом параллельно выполняющиеся подзадачи могут быть частью одной вычислительной задачи, а могут выполняться независимо. В обоих случаях достигается цель распараллеливания выполнения определенной программы – повышение производительности системы в целом.

Если под объектом, обладающим параллелизмом, понимать кластер уровня приложений, то здесь речь идет далеко не о большом количестве однообразных вычислительных подзадач, составляющих, например, задачу моделирования климата. Кластеры уровня приложений зачастую нацелены на решение задач, характеризующихся большими массивами данных, требующими нестандартной обработки. Такими свойствами обладают бизнес-приложения, базы данных для поддержки принятия решений, видеоконференции и др. Кла-

стер, организованный на прикладном уровне, практически стирает грань между одной целостной распределенной задачей и большим набором однородных служб (сервисов), действующих независимо. Тем не менее кластер обладает свойствами, характерными для обоих из приведенных алгоритмов: во-первых, выполняется целостная многофункциональная и многопользовательская задача по обслуживанию запросов пользователей, а во-вторых, множество сеансов различных пользователей (и даже несколько сеансов одного и того же пользователя) протекают параллельно и не зависят друг от друга.

Задача управления параллельными вычислениями в кластерной среде Lotus Domino делится на несколько подзадач.

### КОММУНИКАЦИИ

Коммуникации – это передача межпроцессных сообщений. Каждый процесс, работая независимо, должен иметь связь с другими процессами для передачи результатов обработки и получения заданий на обработку.

В Lotus Domino такие межпроцессные коммуникации возможно осуществлять на основе того, что в кластеризуемых базах данных изменения рассылаются мгновенно на все серверы-члены кластера. Таким образом, записав документ, содержащий результаты работы или указание на завершение части обработки, сервер сообщает о своих действиях другим членам кластера. В этом служебном документе (сообщении) необходимо предусмотреть тип действия (ожидание, завершение, передача управления), а также адресную информацию (имя сервера-адресата или специальный код, указывающий на то, что сообщение широковещательное). Запись документа-сообщения должна производиться в специально отведенную для этого разделяемую служебную область.

### РАЗДЕЛЯЕМАЯ ПАМЯТЬ

В общем случае разделяемая память для кластера уровня приложений Lotus Domino – это набор баз данных (баз документов), над которыми производятся действия по обработке информации (поиск, изменение, удаление документов и т.д.). Какие именно базы данных участвуют в обработке, зависит от конкретной задачи. Эти базы данных являются распределенными по кластеру, что дает возможность каждому серверу обращаться к локальной реплике разделяемой памяти без накладных расходов на коммуникации.

### РАЗДЕЛЯЕМАЯ СЛУЖЕБНАЯ ОБЛАСТЬ

Это специальная база данных, содержащая параметры оперативного окружения процессов, запущенных на серверах-членах кластера. Эта база данных также должна быть кластеризуемой с тем, чтобы каждый процесс (в т.ч. и ведущий) мог получить текущее состояние других процессов во время обработки. Например, используя несколько процессов для поиска конкретного документа, ведущий процесс дает зада-

ния (посредством документов-сообщений) нескольким процессам, запущенным на нескольких серверах, обработать несколько областей одной и той же базы данных. При поиске данных один из процессов обнаруживает искомым документ в разделяемой памяти, немедленно записывая в разделяемую служебную область документ с указанием на то, что поиск успешно закончен, и завершая работу. Остальные процессы в течение поиска периодически сканируют служебную область и, получив сообщение о завершении, прекращают поиск, извещая о готовности к выполнению новых заданий документом-сообщением.

### МАСШТАБИРОВАНИЕ

Используя коммуникации и разделяемую служебную область, новые члены кластера (а точнее, новые процессы) могут регистрироваться в параллельной системе, записывая свой служебный документ с указанием состояния готовности в служебную область. При получении от пользователя нового запроса на обработку ведущий процесс сканирует служебную область на наличие свободных процессов, среди которых окажется новый процесс нового члена кластера. Эта операция происходит без остановки кластерной системы и без остановки параллельного алгоритма, что важно для критичных к этому бизнес-приложений.

### ОТКАЗОУСТОЙЧИВОСТЬ

В случае, если один из процессов или серверов останавливается или выходит из строя, параллельная система должна быть об этом уведомлена. Для этого процесс, получающий задание, помечает свой рабочий документ в служебной рабочей области пометкой «получил задание», фиксируя время получения задания. Ведущий процесс по наступлении определенного таймаута и не получении результата исключает этот процесс из числа готовых к работе, переводя его рабочий документ в статус «недоступен». При возобновлении работы соответствующий процесс находит в рабочей области свой документ и переводит его в статус «готов». В целях сохранения производительности при настройке отказоустойчивости необходимо тщательно настраивать таймаут – время, которое достаточно точно отличает задержку ответа сервера от его недоступности.

### ИНСТРУМЕНТАРИЙ

Для программирования параллельных алгоритмов в Lotus Domino можно использовать агенты, написанные на встроенном объектно-ориентированном языке Lotus Script. Несмотря на то, что этот язык не дает такой производительности, как модули, написанные на универсальных языках более низкого уровня, его использование позволяет проще разворачивать систему на нескольких аппаратно-программных платформах без переписывания модулей и их перекомпиляции. Эти агенты должны начинать работать по триггеру «добавление, удаление, или изменение документов».

### ЛИТЕРАТУРА

1. Коновалов Н., Крюков В. Параллельные программы для вычислительных кластеров и сетей // Открытые системы. 2002. №3. С.12–18.
2. Елагин В. Кластеры против катастроф // Открытые системы. 2002. №6. С.29–36.
3. Дубова Н. Учет и контроль для «коммунальных вычислений» // Открытые системы. 2003. №1. С.34–37.

Статья представлена кафедрой теоретических основ информатики факультета информатики Томского государственного университета, поступила в научную редакцию 30 апреля 2003 г.