УДК 004.031.43

В. В. Лосев

АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ФУНКЦИОНИРОВАНИЯ GRID-СИСТЕМЫ КАК АРХИТЕКТУРНОЙ ПЛАТФОРМЫ РАСПРЕДЕЛЕННЫХ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

Рассмотрены основные алгоритмы функционирования GRID-системы, использующейся в качестве архитектурной платформы для развертывания в распределенных автоматизированных системах управления специализированного программного продукта управления, ориентированного на OLTP (On-line Transaction Processing), с целью распределения транзакционной нагрузки между несколькими узлами.

Ключевые слова: автоматизированная система управления, алгоритм, интерфейс, транзакционная обработка данных.

Реализация режима реального времени, рассматриваемая в контексте выполнения логических операций в автоматизированных системах управления (АСУ), становится приоритетной задачей для многих отраслей промышленности. Рассматривая распределенные АСУ, следует отметить, что современные стандарты и протоколы передачи данных позволяют приблизить скорости распределенных вычислений к локальным. Таким образом, нужно сфокусировать внимание на внутрисистемных вычислениях, реализуемых на уровне операционных систем, прикладного программного обеспечения (ПО) и систем управления базами данных (СУБД).

Проектирование и внедрение таких сложных систем, какими являются АСУ, связано с реализацией (материализацией) в тесной взаимосвязи различных видов обеспечения. В соответствии с ГОСТ 24.104–85 для АСУ выделяется ряд основных видов обеспечения: техническое, математическое, программное, информационное, лингвистическое, организационное [1].

Данные виды обеспечения АСУ представляют исследовательский интерес, прежде всего, с точки зрения анализа их функционирования в рамках системы с учетом выполнения логических операций в режиме реального времени.

Архитектурная платформа распределенных АСУ представляет территориально-рассредоточенные вычислительные мощности (узлы), объединенные каналами связи с классическим клиент-серверным взаимодействием. Не является исключением технология SMP (Symmetric Multiprocessing) [2], в которой несколько процессоров имеют равноправный доступ к совместно используемой основной памяти.

Системы класса SCADA (Supervisory Control And Data Acquisition) являются основным программным обеспечением HMI (Human Machine Interface) АСУ технологического процесса (ТП) [3]. Ввод-вывод данных осуществляется в результате взаимодействий с устройством связи с объектом (УСО), посредством драйверов, а также с БД посредством интерфейсов ODBC (Open DataBase Connectivity).

Поддержка реального времени обеспечена как операционными, так и SCADA системами, что является необходимым условием протекания сложных процессов для АСУ с непрерывным характером производства.

В виду гетерогенной сущности взаимодействующих систем вопрос обмена данными остается весьма актуальным, что особенно характерно для обращений к СУБД со стороны приложений.

Современные коммерческие СУБД берут свое начало от «System R» [4], разработанной в 1970-е гг. XX в., наследием которой стали следующие архитектурные черты: структуры хранения данных и индексов, ориентированых на дисковую память; использование многопотоковости для сокрытия временных задержек; механизмы управления параллельным доступом на основе блокировок; восстановление на основе журналов. «System R» была разработана с учетом характеристик аппаратуры, которые по некоторым показателям в тысячи раз ниже сегодняшних, к тому же существовал единственный рынок СУБД – рынок обработки бизнес-данных.

Данные аргументы заставляют пересмотреть классическую, наследуемую архитектуру «System R» современных коммерческих СУБД с целью увеличения производительности относительно времени совершения транзакций.

Предлагается архитектурная платформа распределенных АСУ, основой которой выступает GRID-система (впервые была исследована в проекте «Gamma» [5]), состоящая из *N*-узлов горизонтального разделения в одноранговой (peer-to-peer) архитектуре, взаимодействующих посредством коммуникационной сети. Данная концепция исключает использование общих, разделяемых ресурсов – «shared-nothing». Возможно горизонтальное масштабирование GRID-системы (количеством узлов) по мере расширения мощностей АСУ, с целью увеличения ресурсного потенциала системы и надежности (в виде избыточности узлов).

Современные технологии разделения ресурсов, в частности SMP, ограничены количеством используемых процессоров и величиной основной памяти. Данное ограничение накладывает операционная система. В то время как архитектурный подход «shared-nothing» не препятствует наращиванию единичных мощностей GRID-системы.

Особенность предложенной платформы – возможность развертывания в основной памяти узлов GRID-системы специализированного программного продукта управления данными, ориентированного на OLTP (On-line Transaction Processing) с целью распределения транзакционной нагрузки между несколькими узлами.

Авторами статьи [5], предложен архитектурный подход при проектировании вышеназванного программно-

го средства для достижения производительности, значительно превосходящей производительность существующих распределенных СУБД.

Размещение приложения и хранение данных в основной памяти с последующим совершением транзакций без обмена данными с дисковой памятью позволит исключить временные задержки при дисковых операциях. Таким образом, методы вызова персистентных объектов исключены. Оперирование данными осуществляется короткими, «легковесными» транзакциями с минимальным временем полезной работы. Отсутствие задержек происходит по вине пользователей, например, в АСУ ТП основной персонал представлен операторами и диспетчерами, задачи которых – наблюдение за информацией, полученной от БД на основе запросов чтения (read). Имеет смысл выполнять все команды SQL в рамках последовательных транзакций с использованием однопотоковой модели исполнения. Следовательно, исключается подсистема регулирования ресурсов, используемая в многопотоковых системах для параллельного выполнения запросов. Нет необходимости особым образом структурировать данные, для которых поддерживается параллельный доступ. Восстановление данных на основе журнализации допустимо, например, журнал откатов (undo), как структура данных в основной памяти с удалением при фиксации транзакций. Нет необходимости производить повторное выполнение операций (redo), поскольку требуемого эффекта можно достичь путем восстановления данных по сети методом репликации с удаленного узла.

Однако, отмечая преимущества OLTP-ориентированных приложений при проектировании распределенных автоматизированных систем управления, следует согласиться, что на сегодняшний день существуют проблемные области, связанные с реализацией ряда методов. Это относится, например, к репликации частей БД между узлами без приостановки транзакций; реализации хранимых процедур как альтернативы двухсторонним коммуникациям между SCADA и БД; методу поддержания таблиц БД в основной памяти узлов и совершению транзакций.

Приоритет в рассмотрении алгоритма взаимодействия систем на основе удаленного вызова процедур связан с нахождением возможности пересмотра логики в работе интерфейса с целью сокращения временных затрат.

Интеграция промышленных автоматизированных систем класса SCADA на основе распределенной OLTP-СУБД различима по способам построения интегрирующей среды. В наиболее простом варианте, называемом «точка-точка», взаимодействие систем осуществляется на основе полного графа, т. е. для каждой пары взаимодействующих систем создается специфическая для них интерфейсная связь в виде конверторов данных с языка одной подсистемы на язык другой. Поскольку число таких дуплексных связей может доходить до $N(N-1)/2$, где N – количество узлов в GRID-системе, то вариант «точка-точка» оказывается приемлемым только для малых GRID-систем. Число связей уменьшается до $N+1$ в варианте интеграции на основе общего для подсистем языка, поддерживаемого промежуточной метасредой [6]. Примером такого языка может служить SQL в технологии ODBC, широко представленной в различных системах, взаимодействующих с внешними данными, для которых характерна двухзвенная схема распределенных вычислений (рис. 1).

Рассматривая один из уровней распределенной АСУ, в частности АСУ ТП, следует отметить, что пример данных, выступающих в качестве обмена между SCADA и OLTP-СУБД, может представлять n-мерный массив, сопоставимый, например, с рядом значений параметров ТП. Архивные тренды в SCADA-системах позволяют опрашивать устройства полевых сетей с частотой 0,5 с, тем самым генерируя объем данных с целью архивации в БД. Таким образом, требуемый обмен данными высокой частоты обуславливает специфические требования к интерфейсам.

Примером удовлетворения запроса посредством интерфейса ODBC (рис. 2) является следующая последовательность действий:

– генерация запроса на совершение определенных действий с БД;

– посылка запроса на интерфейс;

– конвертация запроса в набор SQL-команд (формирование транзакции);

– выполнение SQL-команд в рамках транзакции (манипулирование базой данных);

– возврат приложению запрашиваемых данных.

В данном примере мы наблюдаем многостадийность процесса доступа к СУБД и, как следствие, временные задержки на выполнение каждой стадии.

Использование опции ODBC SQLPASSTHROUGH позволит передать SQL-предложение (запрос) напрямую источнику данных. В данном случае драйвер передает SQL-предложение OLTP-СУБД, исключая предварительную, локальную модификацию. Такой подход улучшает характеристики интерфейса в аспекте производительности, но DYNASET (результат выполнения запроса) может быть открыт только на чтение, например:

db.ExecuteSQL(“BEGIN procedurename(param1, param2, param3); END;”, SQLPASSTHROUGH)

Исходя из требования приблизиться к режиму реального времени, становится возможным сократить времен-

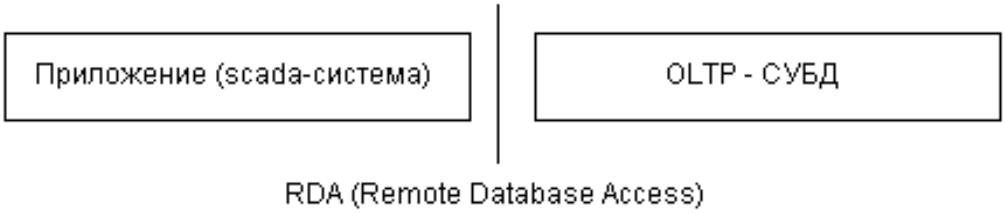


Рис. 1. Двухзвенная схема распределенных вычислений

ные задержки путем реализации модели взаимодействия SCADA с OLTP-СУБД на основе вызова (локальных) хранимых процедур, посредством запуска на выполнение предопределенных транзакций, что заставляет пересмотреть алгоритмическую сущность интерфейса взаимодействия (ODBC).

Анализируя стек работы OLTP-СУБД ODBC, можно отметить, что стадия сетевого транспорта, присутствующая при обращении к удаленной БД (при клиент-серверном взаимодействии), исключается ввиду присутствия части OLTP-СУБД на каждом узле GRID-системы. Также известно, что технология доступа посредством интерфейса ODBC не рассчитана на работу с большим числом клиентов. В случае реализации OLTP-СУБД в основной памяти узлов GRID-системы проблема разрешается с учетом особенности доступа к системе в однопользовательском (монопольном) режиме.

Алгоритм доступа на основе локальных процедур предполагает наличие в стеке работ на стороне OLTP-СУБД дополнительных модулей: менеджер транзакций и реестр транзакций, функция которых – формирование предопределенных транзакций и их хранение.

Для реализации взаимодействия систем на основе вызова локальных процедур посредством интерфейса ODBC драйверу OLTP-СУБД необходимо соответствие требованиям базового уровня API (Application Programming Interface) ODBC (CORE API), а также Уровню 1 (Level 1):

– поддержка хранимых процедур, включая опрос словаря в отношении хранимых процедур;

– доступ к содержимому наборов, созданных в результате выполнения пакетов и хранимых процедур.

Таким образом, применяя логику взаимодействия систем на основе вызова хранимых процедур, а не удаленных, становится возможным снижение нагрузки на ODBC-интерфейс за счет исключения проверок синтаксиса команд, аргументов функций и корректности изменения состояния БД, что позволит увеличить пропускную способность данного интерфейса. При этом вычислительная нагрузка перекладывается с интерфейса на OLTP-СУБД. Рассматривая OLTP-СУБД как распределенную систему, поддерживаемую в основной памяти узлов GRID-системы, следует отметить отсутствие операций с дисковой памятью, что позволит увеличить скорость реакции OLTP-СУБД на запросы от внешних систем.

Библиографический список

1. ГОСТ 24.104–85. Единая системы стандартов автоматизированных систем управления. Автоматизированные системы управления. Общие требования.

2. Бройдо, В. Л. Вычислительные системы, сети и телекоммуникации / В. Л. Бройдо. СПб. : Питер, 2002.

3. Деменков, Н. П. SCADA-системы как инструмент проектирования АСУ ТП / Н. П. Деменков. М. : Изд-во МГТУ им. Н. Э. Баумана, 2004.

4. Kim, W. Relational Database Systems / W. Kim // ACM Comput. Surv. 1979. № 3.

5. The End of an Architectural Era (It's Time for a Complete Rewrite) / M. Stonebraker, S. Madden, D. Abadi et al. Proceedings of VLDB. 2007. Vienna. Austria.

6. Норенков, И. П. Основы автоматизированного проектирования / И. П. Норенков. М. : Изд-во МГТУ им. Баумана, 2006.

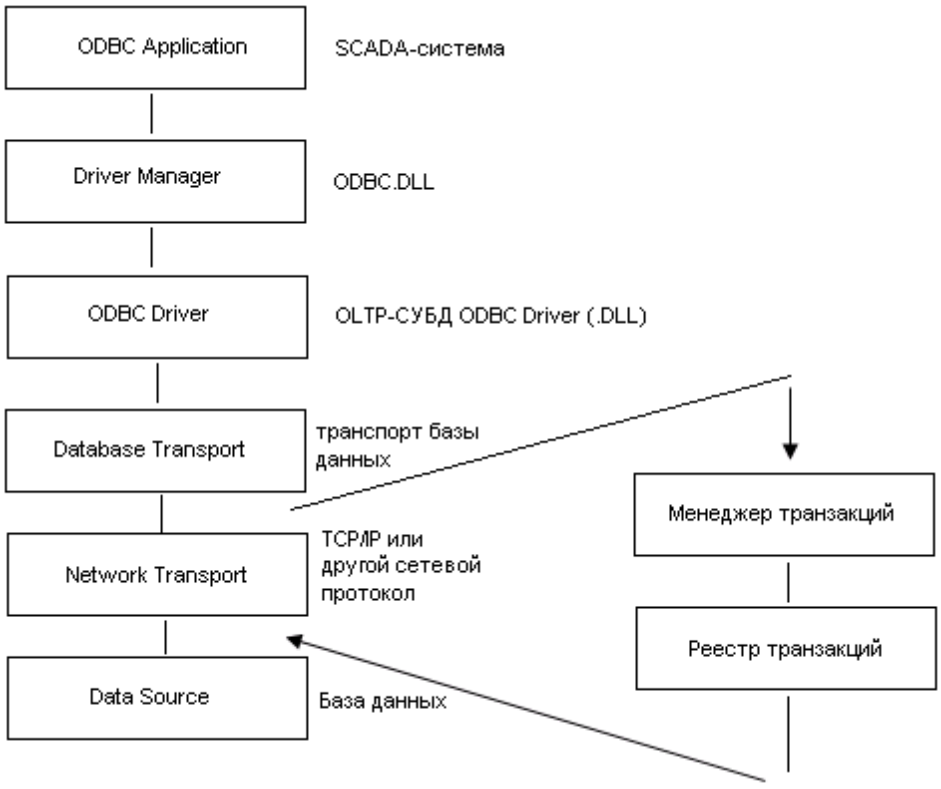


Рис. 2. Стек работы OLTP-СУБД ODBC

V. V. Losev

ALGORITHMIC SUPPORT OF GRID-SYSTEM FUNCTIONING AS THE ARCHITECTURAL PLATFORM OF DISTRIBUTED AUTOMATED CONTROL SYSTEMS

The paper considers the main algorithms of GRID-system functioning used as the architectural platform for the deployment in distributed automated control systems of specialized OLTP-oriented software for data management to distribute the transactional load among multiple nodes.

Keywords: automated control system, algorithm, interface, transaction data processing.

УДК 550.834(075)

В. А. Детков

ИМПУЛЬСНЫЕ НЕВЗРЫВНЫЕ ИСТОЧНИКИ СЕЙСМОРАЗВЕДКИ С ЭЛЕКТРОМАГНИТНЫМ ПРИВОДОМ

Дается анализ методов и конструкций невзрывных источников сейсмических волн, используемых при сейсморазведочных работах на нефть и газ. Описывается принцип действия невзрывных источников, делается акцент на преимуществе импульсных невзрывных источников, реализованных в отечественной системе «Енисей».

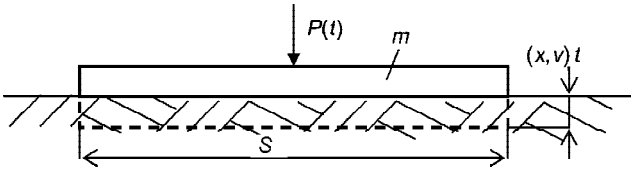
Ключевые слова: невзрывные, импульсные, сейсмоисточники, электромагнитные.

Метод создания сейсмических волн с помощью взрыва заряда взрывчатого вещества в скважине или на поверхности грунта, традиционно применявшийся при проведении сейсморазведочных работ в 1960–70 гг., в основном исчерпал свои возможности развития и применения в связи с большой стоимостью буровзрывных работ и ограниченным применением при его использовании более совершенных приемов обработки сейсмической информации. Взрывной способ наносит значительный вред природе, не может использоваться в местах поселений, вблизи мостов, линий электропередач, железных дорог.

Для проведения сейсморазведочных работ стали разрабатываться альтернативные взрывному невзрывные способы создания сейсмических волн. Такого вида устройства, создающие упругие волны в грунте, получили название невзрывных сейсмоисточников.

Создание сейсмических волн в грунте невзрывным сейсмоисточником происходит в результате деформации x грунта излучающим элементом, обычно выполненным в виде жесткой массивной плиты m, расположенной на поверхности грунта, при воздействии на плиту усилием $P(t)$, создаваемым приводом сейсмоисточника (см. рисунок). В зависимости от характера изменения прикладываемой к плите силы и создаваемых при этом деформаций грунта невзрывные поверхностные сейсмоисточники принято классифицировать на три типа: вибрационные, импульсные и кодоимпульсные (виброимпульсные) [1].

Сейсмические вибраторы создают под плитой квазигармонические деформации грунта с изменяемой частотой колебаний в течение длительного интервала времени (до 10 с и более). Наиболее широкое применение находят мощные электрогидравлические вибраторы. Основными их недостатками являются большая стоимость и эксплуатационные расходы и ограниченные возможности применения ввиду значительного веса вибратора. Эти недостатки обусловлены в основном двумя взаимосвязанными факторами: принципом действия, в основу которого положена необходимость создания квазигармонических силовых воздействий на излучающую плиту, и применением гидравлического силового привода, КПД преобразования потребляемой энергии которого в механическую энергию движения излучающей плиты и упругие деформации грунта не превышает нескольких процентов.



Принцип работы невзрывного сейсмоисточника

При создании импульсных невзрывных сейсмоисточников преследовалось достижение иной цели – решение задач импульсной взрывной сейсморазведки более эффективными и дешевыми техническими средствами и расширение возможных областей применения сейсморазведки [1; 2].

Сейсмоисточники кодоимпульсного типа занимают некоторое промежуточное положение между вибрационными и импульсными. Они создают на излучающую