

УДК 681.3

**КОМПЛЕКСНЫЙ ПОДХОД К СОЗДАНИЮ
РАСПРЕДЕЛЕННЫХ СИСТЕМ****С.Б.САМЕДОВ***Бакинский Государственный Университет**samir.samedov@gmail.com*

В данной работе предложен комплексный подход к созданию распределенных систем. Рассмотрены современные технологии и методы, используемые для поддержания фундаментальных правил распределенных систем. Проведены экспериментальные сравнительные анализы влияния на производительность распределенной системы при использовании различных технологий.

Ключевые слова: распределенная система, база данных, транзакция, производительность.

С развитием информационных систем, применение распределенных систем (РС) растет с каждым днем. РС это множество территориально удаленных друг от друга узлов, объединенных системой передачи данных и взаимодействующих посредством обмена сообщениями [1]. Также надо отметить, на сегодняшний день особым видом распределенных систем, представляющих большой интерес, являются распределенные базы данных (РБД). Для пользователей при работе с РБД это выглядит, не что иное, как работа с обычной единой базой данных.

На сегодняшний день имеется большинство компаний предлагающих на рынок свои продукты РС, но лидером среди них является продукты компаний Oracle, IBM, Ingres.

Имеются фундаментальные правила, которым должна удовлетворять РС:

- Локальная независимость
- Отсутствие зависимости от центрального узла
- Непрерывное функционирование
- Независимость от расположения
- Независимость от фрагментации
- Независимость от репликации
- Обработка распределенных запросов
- Управление распределенными транзакциями
- Аппаратная независимость

- Независимость от операционной системы
- Независимость от сети
- Независимость от типа СУБД

Создание правильной конструированной производительной распределенной системы является одной из актуальных тем на сегодняшний день. В различных публикациях рассматриваются отдельно взятые технологии и методы, применяемые при построении РС. Главной целью данной работы является исследование комплексного подхода при построении РС.

Фундаментальные правила

Рассмотрим более подробно каждое из этих правил и какие технологии имеются на сегодняшний день удовлетворяющие этим правилам.

- ***Локальная независимость***

Данное свойство указывает на то, что узлы объединены в одну РС, но не смотря на это, каждый узел локально представляет собой независимую единицу, функционирующую независимо от других узлов системы.

К примеру, технология Oracle Real Application Cluster (RAC) компании Oracle [2]. РС сохраняет работоспособность в случае сбоя некоторых узлов кластера, система автоматически перераспределяет нагрузку на оставшиеся рабочие узлы.

Архитектура RAC состоит из:

а. Публичная сеть для связи между клиентами и серверами кластера. Именно с использованием этой сети производится подключение клиентских сессий к базе данных, их балансировка между узлами и аварийное переключение в случае сбоя.

б. Приватная или внутренняя сеть, обычно называемая межсоединением, для передачи сообщений между узлами. Дублирование сетевых интерфейсов увеличивает надежность кластерной конфигурации.

в. Независимые компьютеры / узлы кластера в роли узлов могут выступать компьютеры с различной конфигураций. Мы можем получить производительную и отказоустойчивую систему с обычными РС не тратя деньги на дорогостоящие mainframe-ы.

г. Разделяемое устройство хранения – в данных устройствах хранится база данных.

Таким образом, можно сделать вывод о том, что поддерживая локальную независимость в РС, обеспечивается отказоустойчивость и возможность динамической масштабируемости системы.

- ***Отсутствие зависимости от центрального узла***

В зависимости от связей между узлами РС делятся на централизованные и децентрализованные. Централизованные системы это системы, где связь между узлами осуществляется через один центральный узел. В децентрализованных системах связи между узлами осуществляются на прямую.



Рис. 1. Централизованная система (слева) и децентрализованная система (справа).

Использование централизованной системы не желательно, так как при этом центральный узел становится узким местом системы. Нужно продумать все аспекты как мощность центрального узла, сетевую пропускную способность, так как весь обмен происходит через центральный узел. И конечно же, система становится уязвимой к отказу, так как при выходе из строя центрального узла, выйдет из строя вся РС.

В случае же РБД все узлы также равноправны и независимы, а расположенные на них базы являются равноправными поставщиками данных в общее пространство данных. База данных на каждом из узлов самодостаточна - она включает полный собственный словарь данных и полностью защищена от несанкционированного доступа.

Таким образом, можно сделать вывод о том, что централизованная система имеет слабую отказоустойчивость.

- **Непрерывное функционирование**

Одно из больших преимуществ распределенных систем как раз и состоит в том, что они должны предоставлять непрерывное функционирование. К примеру, технология High Availability Cluster Multiprocessing (HACMP) [3, 4] компании IBM:

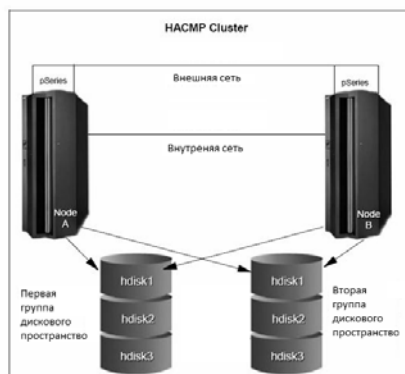


Рис. 2. Технология High Availability Cluster Multiprocessing.

Система построена на принципе того, что при выходе из строя первого узла, автоматически его роль подхватывает второй узел, таким образом, обеспечивая непрерывное функционирование.

На сегодняшний день для большинства систем ставиться требова-

ние 24/7 (24 дня, семь дней в неделю), то есть непрерывное функционирование. Но система требует определенных обновлений и административные работы, в этом случае при использовании НАСМР возможно прерывание профилактических работ на втором узле (пассивном узле), по завершению работ перевести систему на работу во втором узле и начать работу в первом узле, не влияя на работу всей системы.

Таким образом, можно сделать вывод о том, что непрерывное функционирование это одно из самых главных требований, которое должны поддерживать РС работающие в режиме на 24/7.

- **Независимость от расположения**

Основная идея заключается в том, что работа пользователей с СУБД или РБД с одиночным узлом должна быть прозрачной, им не надо знать, где физически расположены данные. Благодаря независимости от расположения упрощаются пользовательские программы. В приведенном выше технологии Oracle RAC, как видно, обеспечивается вся необходимая независимость от расположения.

Таким образом, можно сделать вывод о том, что независимость от расположения облегчает работу пользователям с РС.

- **Независимость от фрагментации**

Фрагментация это разделение информации на несколько порций и распределение этих порций на различные дисковые пространство. В этом случае данные могут храниться в том месте, где они чаще всего используются, что позволяет достичь локализации большинства операций и уменьшения сетевого трафика, благодаря этому получается выигрыш в производительности всей системы.

К примеру, в системе имеется объект содержащий в себе информацию о прошедших финансовых транзакциях. Таким образом, если фрагментировать информацию в зависимости от необходимости информации в том или ином узле, то получается выигрыш в производительности так как локализуются операции.

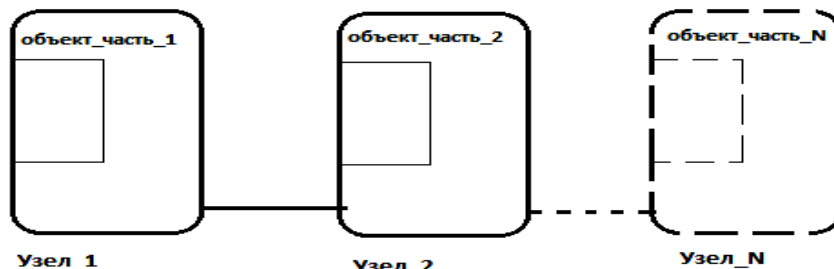


Рис. 3. Фрагментация объекта в РС.

Таким образом, можно сделать вывод о том, что фрагментация полезна для повышения производительности системы.

- **Независимость от репликации**

Репликация это хранение нескольких отдельных копий (реплик), которые хранятся на нескольких отдельных узлах. Благодаря этому подходу можно реплицировать информацию на нескольких узлах. При этом можно преследовать две разные цели.

Первая цель это выигрыш производительности, так как если несколько узлов часто используют одну и ту же информацию среплицировав ее по этим узлам, получается высокая скорость чтения, так как считывание локально с диска намного быстрее сетевого считывания, а также уменьшение сетевого трафика приводит к увеличению производительности системы.

Вторая цель это реплицирование данных для создания резервной копии на случай потери основной копии. Для репликации данных имеется очень удобная технология Oracle Streams [5].

Архитектура Oracle Streams очень гибкая и состоит из трех основных элементов: сбор, хранение, потребление.

Для создания резервной копии также используется технология Oracle Data Guard [6]. В технологии Data Guard основная база данных (primary site) зеркалируется на резервные узлы (standby site).

Таким образом, можно сделать вывод о том, что репликация системы увеличивает производительность системы, а также доступность так как пока объект реплицирован в системе это означает, что по крайней мере у нас есть копия объекта в случае потери самого объекта.

- **Обработка распределенных запросов**

Распределенный запрос, это запрос требующий выполнение запроса не локально на узле, а в узлах РС. Поэтому при выполнении распределенных запросов очень важен механизм позволяющий определение узлов, на которых располагаются необходимые для выполнения запроса объекты.

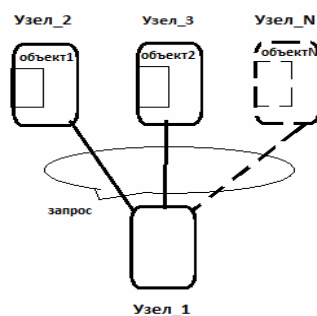


Рис. 4. Распределенный запрос в РС.

Данное свойство актуально для РБД, так как в случае расположения различных систем управления базы данных (СУБД) на различных узлах РБД должно обеспечить правильное выполнение распределенных запросов. На сегодняшний день продукты компаний Informix Software [7], Ingres Corp [8], Oracle полностью удовлетворяют этому условию.

Таким образом, можно сделать вывод о том, что обработка рас-

предельных запросов в РС является одним из главных аспектов.

- ***Управление распределенными транзакциями***

Данное требование также актуально для РБД, так как в независимости от РС или обычной системы транзакции должны удовлетворять свойству АСИД (атомарность, согласованность, изолированность, долговечность).

Также в РС системах актуально управление параллельностью, большинство систем для поддержания параллельности используется механизм блокировки, хотя в последнее время также широко применяется метод поддержки многих версий.

Таким образом, можно сделать вывод о том, что РС должна поддерживать как свойства транзакций АСИД, так и параллельность выполнения транзакций.

- ***Аппаратная независимость***

На сегодняшний день парк серверов в компаниях может состоять из различных компьютеров, к примеру, компьютеры производства компаний IBM, Fujitsu, HP, персональные компьютеры и т.д. Данное свойство полностью удовлетворяется в технологии Oracle RAC.

Благодаря Oracle RAC возможно в РС включить в качестве узлов как дорогостоящий сервер, так и обычный персональный компьютер. К примеру, Министерство обороны США создало суперкомпьютер из 1760 игровых приставок Sony PlayStation3. Вычислительная машина под наименованием Condor способна выполнять 500 триллионов операций с плавающей запятой в секунду и размещена в исследовательской лаборатории ВВС США в Нью-Йорке.

Таким образом, можно сделать вывод о том, что используя в узлах РС недорогие вычислительные машины можно получить серьезный дорогостоящий сервер.

- ***Независимость от операционной системы (ОС)***

Данное правило в какой-то степени аналогично предыдущему правилу, так как имея различные компьютеры различных фирм, мы имеем, соответственно, различные ОС, к примеру, на машинах IBM устанавливается ОС AIX, на машинах HP устанавливается ОС HP-UX и т.д.

Таким образом, можно сделать вывод о том, что поддержка различных ОС в РС также необходима.

- ***Независимость от сети***

При построении РС связь между различными узлами может быть установлена различными сетевыми технологиями. Сетевая технология – это согласованный набор стандартных протоколов и реализующих их программно-аппаратных средств, достаточный для построения локальной вычислительной сети. Сетевые технологии называют базовыми технологиями или сетевыми архитектурами локальных сетей.

В настоящее время сетевые технологии локальных сетей IEEE802.3/Ethernet являются наиболее популярными в мире. Популярность обеспечивается простыми, надежными и недорогими технологиями. В классической локальной сети Ethernet применяется стандартный коаксиальный кабель двух видов (толстый и тонкий).

Таким образом, можно сделать вывод о том, что хотя РС, в целом, не зависит от типа сети организованной в ее узлах, но производительность РС напрямую зависит от выбранной сетевой технологии, влияющей на пропускную способность обмена информацией между ее узлами.

- **Независимость от типа СУБД**

Данное требование актуально для РБД, так как на различных узлах могут быть использованы различные СУБД. РБД можно разделить на однородные и неоднородные. В однородных РБД в качестве СУБД для всех узлов выступает одна и та же СУБД. В неоднородных РБД в различных узлах могут функционировать, соответственно, различные СУБД. На сегодняшний день продукты компаний Informix Software, Oracle полностью удовлетворяют этому правилу.

Таким образом, можно сделать вывод о том, что современные РС должны поддерживать как однородные, так и неоднородные системы.

Экспериментальная часть

При комплексном подходе необходимо применение всех передовых методов, благодаря этого получается повышение производительности всей РС. Из рассмотренных методов главную роль на производительность системы играет репликация данных, поддержка параллельных запросов, фрагментация данных. Рассмотрим более подробно влияние каждого из метода на производительность. Эксперимент будет производиться на тестовом стенде РБД Oracle Database 11g Enterprise Edition Release 11.2.0.1.0.

- **Репликация данных**

При репликации данных получаем повышение производительности на вторичных узлах за счет локализации операций при работе с реплицируемыми данными. Но при включении репликации воздействуем на производительность работы первичного узла. Таким образом, поставлена задача выяснение влияние репликации на производительность первичного узла.

Таблица 1

Проведение операций вставки при включенной и выключенной репликации

Эксперимент №	Кол-во операций	Состояние объекта	Потраченное время на исполнение (в секундах)
1_1	10,000	репликация объекта выключена	24
1_2	10,000	репликация объекта включена	28
2_1	20,000	репликация объекта выключена	47
2_2	20,000	репликация объекта включена	49
3_1	30,000	репликация объекта выключена	76

3_2	30,000	репликация объекта включена	79
4_1	40,000	репликация объекта выключена	103
4_2	40,000	репликация объекта включена	109
5_1	50,000	репликация объекта выключена	133
5_2	50,000	репликация объекта включена	137

На рис. 5 построен график работы основываясь на табл. 1.

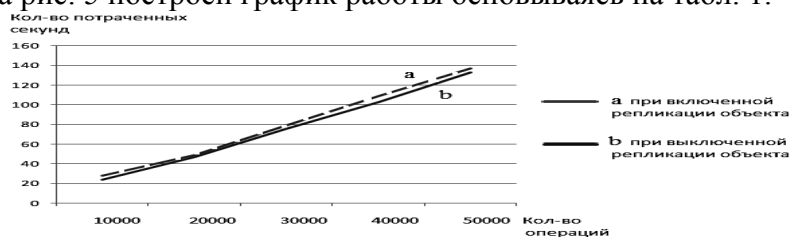


Рис. 5. График прохождения операций вставки при включенной и выключенной репликации.

Получается, что включение репликации оказывает минимальное воздействие на производительность операций в первичном узле.

- **Параллельные запросы**

Используя параллельности можно получить также повышение производительности РС. Таким образом, поставлена задача выяснение влияние параллельности на производительность РС.

Проведем перенос данных с одной таблицы в другую с использованием параллельности и без использование. При использовании параллельности будет использоваться след запрос.

```
INSERT /*+ PARALLEL(test_load_tbl,2) */ INTO test_load_tbl
SELECT /*+ PARALLEL(test_load_tbl_2, 2) */ * FROM test_load_tbl_2;
```

Таблица 2

Перенос данных при включенной и выключенной параллельности

Эксперимент №	Кол-во операций	Состояние объекта	Потраченное время на исполнение (в секундах)
1_1	250,000	параллельность выключена	19
1_2	250,000	параллельность включена	13
2_1	500,000	репликация объекта выключена	37
2_2	500,000	репликация объекта включена	25
3_1	750,000	репликация объекта выключена	59
3_2	750,000	репликация объекта включена	40
4_1	1,0000,0000	репликация объекта выключена	78
4_2	1,0000,0000	репликация объекта включена	53
5_1	1,250,0000	репликация объекта выключена	98
5_2	1,250,0000	репликация объекта включена	69

На рис. 6 построен график работы основываясь на табл. 2.

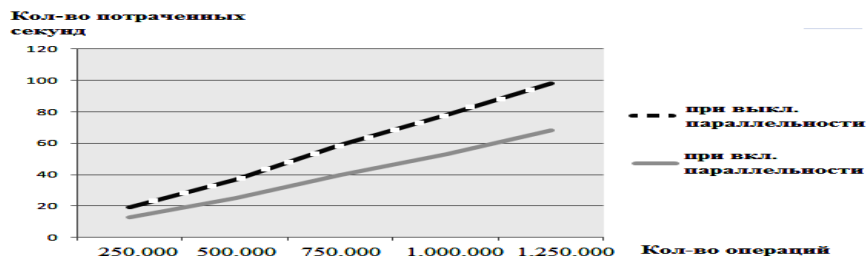


Рис. 6. График переноса данных при включенной и выключенной параллельности.

Получается, что использование параллельности воздействует на повышение производительности в РС.

- **Фрагментация данных**

При фрагментации данных получается повышение производительности РС. Таким образом, поставлена задача выяснение влияние фрагментации на производительность РС.

Таблица 3

Проведение операций вставки на узле Node 1

Эксперимент №	Кол-во операций	Состояние объекта	Потраченное время (в сек.)
1 1	10,000	Объект фрагментирован и располагается на узлах: Node 1 и Node 2	40
1 2	10,000	Объект не фрагмен. и располагается весь на втором узле Node 2	55
2 1	20,000	Объект фрагментирован и располагается на узлах: Node 1 и Node 2	78
2 2	20,000	Объект не фрагмен. и располагается весь на втором узле Node 2	112
3 1	30,000	Объект фрагментирован и располагается на узлах: Node 1 и Node 2	124
3 2	30,000	Объект не фрагмен. и располагается весь на втором узле Node 2	171
4 1	40,000	Объект фрагментирован и располагается на узлах: Node 1 и Node 2	163
4 2	40,000	Объект не фрагмен. и располагается весь на втором узле Node 2	221
5 1	50,000	Объект фрагментирован и располагается на узлах: Node 1 и Node 2	211
5 2	50,000	Объект не фрагмен. и располагается весь на втором узле Node 2	277

На рис. 7 построен график работы основываясь на табл. 3.

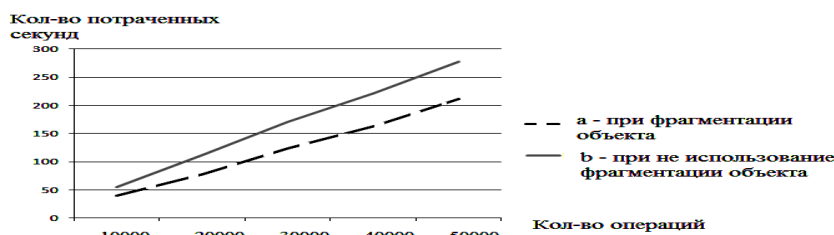


Рис. 7. График прохождения операций вставки на узле Node_1

Получается, что фрагментация объектов воздействует на увеличение производительности в РС.

Заключение

Таким образом, в данной работе предложен комплексный подход к созданию распределенных систем. Изучены современные решения, предлагаемые для поддержания фундаментальных правил распределенных

систем. Применив различные технологии и методы, экспериментально показали повышение производительности распределенных систем.

ЛИТЕРАТУРА

1. Дейт К. Дж. Введение в системы баз данных. М.: Вильямс, 2005, 1328 с.
2. Oracle Literature. Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide. Part No. B14197-10. United States of America, 2010, 402 p.
3. IBM Redbooks. IBM TotalStorage: FASSt Best Practices Guide. United States of America, 2003, 122 p.
4. IBM Redbooks. Linux HPC Cluster Installation. United States of America, 2001, 254 p.
5. Thomas Kyte. Expert Oracle Database Architecture 9i and 10g Programming Techniques and Solutions. United States of America, 2005, 768 p.
6. Larry Carpenter, Joseph Meeks, Charles Kim, Bill Burke. Oracle Data Guard 11g Handbook. United States of America, 2009, 544 p.
7. IBM Redbooks. IBM Informix Flexible Grid Extending Data Availability. United States of America, 2011, 570 p.
8. Ingres Literature. Database Administrator Guide. United States of America, 2010, 557 p.

PAYLANMIŞ SİSTEMLƏRİN YARADILMASINDA KOMPLEKS YANAŞMA ÜSULU

S.B. SƏMƏDOV

XÜLASƏ

Bu məqalədə paylanmış sistemlərin yaradılması üçün kompleks yanaşma üsulu təklif edilmişdir. Müasir texnologiyalar və metodlar nəzərdən keçirilmiş və paylanmış sistemlərin fundamental qaydalarını dəstəkləmək üçün istifadə olunmuşlar. Paylanmış sistemlərin məhsuldarlığına müxtəlif texnologiyaların təsiri müqayisəli araşdırmalar nəticəsində müəyyənləşdirilmişdir.

Açar sözlər: paylanmış sistem, verilənlər bazası, tranzaksiya, məhsuldarlıq.

AN INTEGRATED APPROACH TO THE CREATION OF DISTRIBUTED SYSTEMS

S.B.SAMADOV

SUMMARY

In this paper, we propose an integrated approach to building distributed systems. The modern technologies and methods used to maintain the fundamental rules of distributed systems have been studied. Experimental comparative studies have been conducted on the impact performance of a distributed system with different technologies.

Key words: a distributed system, database, transaction, performance.

Поступила в редакцию: 23.09.2011 г.

Подписано к печати: 19.12.2011 г.