НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК АЗЕРБАЙДЖАНА
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

На правах рукописи

САМИР БАХТИЯР ОГЛЫ САМЕДОВ

ИССЛЕДОВАНИЕ И РАЗРАБОТКА АЛГОРИТМОВ И ПРОГРАММНЫХ СРЕДСТВ ПОВЫШАЮЩИХ ПРОИЗВОДИТЕЛЬНОСТЬ РАСПРЕДЕЛЕННЫХ СИСТЕМ

3338.01– системный анализ, управление и обработка информации

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
доктора философии по техническим наукам

Баку – 2013

Работа выполнена в Бакинском Государственном Университете.

Научный руководитель:
доктор технических наук, профессор,
заслуженный учитель **А.А. АЛИЕВ**

Официальные оппоненты:
доктор технических наук, профессор **М.А. АХМЕДОВ**

доктор философии по техническим наукам **Х.Х. АБДУЛЛАЕВ**

Ведущая организация: Национальная Авиационная Академия, кафедра «Аэрокосмических информационных технологий и системы управления»

Защита диссертации состоится « 19 » июня 2013 года в 16:00 часов на заседании диссертационного совета FD.01.231 при Институте Информационных Технологий Национальной Академии Наук Азербайджана по адресу: AZ1141, г. Баку, ул. Б.Вахабзаде, 9

С диссертацией можно ознакомиться в библиотеке Института Информационных Технологий НАН Азербайджана

Автореферат разослан « ____ » _____ 2013 г.

Учёный секретарь
диссертационного совета FD.01.231,
доктор философии по техническим наукам **Р.Г. Шыхалиев**

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы. Современные информационные технологии очень сложно представить себе без распределенных систем (РС). РС это система, состоящая из различных узлов, взаимодействующих между собой в сети посредством сообщений.

Разумеется, по мере распространения и использования РС, у людей возникает потребность в быстрой, надежной и бесперебойной обработки информации. Одним словом, возникает потребность в производительности, а именно в ее повышении. Повышение производительности – это одна из актуальных и важных задач, стоящая перед каждым, использующим распределенные системы.

Повышение производительности РС возможно при настройке приложений, работающих с РС, настройке самого программного обеспечения РС, настройке операционных систем, использующихся в РС и наращиванием аппаратной части. Правильная настройка РС обеспечивает максимально возможную производительность приложения, работающего в конкретной конфигурации программной и аппаратной среды.

В особенности актуальны РС, где каждый узел представляет собой полноценную локальную систему управления базой данных (СУБД), такие РС называют системами распределенной базы данных (РБД). Существуют следующие фундаментальные правила РБД: локальная независимость, отсутствие зависимости от центрального узла, отказоустойчивое 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.

Таким образом, при повышении производительности РС необходимо акцентироваться на компоненты, из которых состоит и должна поддерживать РС. Ключевые места при повышении производительности РС являются: повышение эффективности транзакционной обработки информации; оптимальный подход при логировании для обеспечения восстановления РС; применения технологий репликации и фрагментации; комплексное использование

всех имеющихся технологий для создания оптимальной распределенной системы.

Цель и задачи работы. Целью диссертационной работы является разработка алгоритмов и программного обеспечения, повышающих производительность распределенных систем и реализация предложенных алгоритмов в системах с распределенной структурой.

Для достижения этой цели в диссертационной работе решены следующие задачи:

- исследование общих принципов построения распределенных систем обработки информации;
- анализ фундаментальных правил распределенных систем и исследование воздействия каждого правила на производительность распределенных систем;
- анализ современных методов и алгоритмов, использующихся при повышении производительности распределенных систем;
- разработка методов, повышающих эффективность обработки информации в распределенных системах;
- разработка алгоритмов репликации данных и программного обеспечения, увеличивающих производительность распределенных систем;
- разработка алгоритмов, увеличивающих производительность распределенных систем при фрагментации данных;
- реализация предложенных алгоритмов и методов в системах распределенной структуры.

Методы исследования базируются на применении теории распределенных систем, теории баз данных, теории распределенных баз данных, теории транзакционной обработки данных. Были построены различные тестовые стенды, на которых проведены эксперименты предложенных алгоритмов.

Основные положения, выносимые на защиту:

- методы, повышающие эффективность обработки информации в распределенных системах;
- алгоритм асинхронной репликации данных и программное обеспечение, увеличивающие производительность неоднородных распределенных систем;

- алгоритм репликации данных с первичных узлов на вторичные узлы, повышающий производительность распределенной системы;
- алгоритмы фрагментации данных, увеличивающих производительность распределенных систем.

Научная новизна диссертационной работы состоит в следующем:

1. Предложен экономичный модифицированный метод логирования при загрузке большого объема данных в базу данных, ускоряющий процесс восстановления базы данных при различных отказах системы.
2. Предложен модифицированный метод восстановления базы данных, увеличивающий производительность распределенной системы обработки информации. Предложено расположить UNDO табличное пространство непосредственно в оперативной памяти.
3. Разработан алгоритм асинхронной репликации данных, увеличивающий производительность в неоднородных распределенных базах данных.
4. Разработан алгоритм репликации данных с первичных узлов на вторичные узлы, повышающий производительность распределенной системы.
5. Разработаны алгоритмы фрагментации данных в распределенных базах данных, увеличивающие эффективность распределенной системы обработки информации.

Практическая ценность и реализация результатов работы.

Полученные результаты представляют собой практическую ценность и могут быть использованы как для улучшения производительности уже имеющихся распределенных систем, так и при построении различных распределенных систем.

Основные результаты диссертационной работы были использованы и реализованы при построении распределенной системы процессингова центра Азерикард, являющимся полностью сертифицированным Международными платежными системами MasterCard, Visa, American Express и Diners Club.

Апробация работы. Основные научно-теоретические и практические результаты диссертационной работы докладывались и обсуждались на : XIV, XV, XVI республиканских научных

конференциях докторантов и молодых исследователей Баку, 2009, 2010, 2011 (в 2009 г. работа заняло первое место среди работ, сданных в разделе информационных технологий); научной конференции посвященной 90-летию Бакинского Государственного Университета «Современные проблемы математики, информатики и экономики», Баку, 2009; 1-ой Международной научно-практическая конференция «Информационные системы и технологии в таможенной сфере», Баку, 2009; ежегодной международной открытой научной конференции «Современные проблемы информатизации в анализе и синтезе программных и телекоммуникационных систем», Воронеж, 2010; 3-ей международной научно-практической конференции «Информационные системы и технологии в таможенной сфере», Баку, 2011; восьмой международной научно-практической конференции «Интернет-образование-наука» (ИОН), Винница, 2012.

Публикации. По результатам диссертационной работы опубликовано 15 печатных работ, из них 7 статей и 8 тезиса.

Структура и объем работы. Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы, приложения, и изложена на 144 страницах. Работа содержит 12 рисунков, 7 таблиц. Список литературы включает 114 наименования.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертационной работы, сформулированы цели и задачи исследований, научная новизна и практическая ценность полученных результатов, приводится краткое содержание работы.

В первой главе рассматриваются компьютерные сети, эталонная модель компьютерных сетей. Рассматриваются требования, предъявляемые к современным сетям, основные характеристики производительности современных компьютерных сетей.

Далее дается общее определение распределенных систем. Рассматриваются продукты большинства компаний, предлагающих на рынок свои продукты РС. Создание правильной конструированной, производительной распределенной системы является одной из актуальных тем на сегодняшний день. Приведены 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.

Рассмотрено более подробно каждое из этих правил, и какие технологии имеются на сегодняшний день, удовлетворяющие этим правилам. Предложено применение различных технологий и методов, приводящих к повышению производительности распределенных систем. Далее в главе проводится обзор и сравнительный анализ по имеющимся работам, для повышения производительности распределенных систем.

Во второй главе рассмотрена отказоустойчивость РС. Отказоустойчивость – это способность системы продолжать стабильное функционирование, при возникновении отказов ее элементов. Цель создания каждой отказоустойчивой системы это сохранение функционирования системы, при максимально больших количествах отказов. Минимально отказоустойчивая система гарантирует стабильность системы, при любом отказе одного из ее элементов. Обсуждены недостатки метода “логирования для восстановления” при загрузке большого объема данных. Рассмотрены недостатки утилиты SQL*Loader при методе “прямой загрузки”. Предложено использование экономичного метода логирования, для ускорения процесса восстановления базы данных при отказах.

Шаг 1. В журнальных файлах фиксируется начало процесса загрузки.

Шаг 2. Перед началом процесса загрузки информации, находятся High Water Mark (HWM) таблиц, участвующих при загрузки.

Шаг 3. Выключается стандартное логирование для данной загрузки.

Шаг 4. Начинается загрузка данных. Данные загружаются только в блоки выше HWM.

Шаг 5. а) В случае неуспешной загрузки:

- вся загруженная информация игнорируется – при этом просто HWM переносится на прежнее место;

- в журнальных файлах фиксируется, что данный процесс оборван и для восстановления не нужен;

б) В случае успешной загрузки:

запускается процесс копирования всех блоков с первоначального HWM до текущего HWM в журнальные файлы.

Шаг 6. Только после успешного логирования всех блоков в журнальных файлах происходит фиксация загрузки (commit).

Далее в главе рассмотрено автоматическое и ручное управление табличным пространством UNDO. Рассмотрены инициализационные параметры, использующиеся при конфигурировании табличного пространства UNDO. Предложен метод увеличения производительности системы при расположении UNDO табличного пространства непосредственно в ОЗУ.

Шаг 1. Резервируется место в ОЗУ для использования его как обычного жесткого диска, при этом для работающих приложений это выглядит, как виртуальный диск V: (используется утилита SoftPerfect RAM Disk Version 3.0.2).

Шаг 2. После того как ОЗУ представлено в виде виртуального диска V:, создается кроме основного UNDO табличного пространства, еще одно UNDO табличное пространство, располагая его на виртуальном диске V:

```
CREATE UNDO TABLESPACE UNDO_TEST_ON_RAM
DATAFILE  'V:\UNDO_TEST_ON_RAM01.dbf'
SIZE 350M AUTOEXTEND OFF
RETENTION NOGUARANTEE
BLOCKSIZE 8K
FLASHBACK ON;
```

Шаг 3. Создается тестовая таблица, в дальнейшем она будет служить для измерения производительности выполнения операций, при различном расположении UNDO табличного пространства на физическом носителе, то на ОЗУ.

Шаг 4. Запускается скрипт, который заполняет тестовую таблицу UNDO_RAM_TEST данными. После выполнения скрипта количество записей в таблице равно 266430.

Шаг 5. Запускается несколько раз скрипт, который три раза изменяет данные и далее запускает команду отката ROLLBACK, замеряется.

Шаг 6. Перенастраивается база данных на использование ранее созданного undo табличного пространства

UNDO_TEST_ON_RAM.

SQL>ALTER SYSTEM SET UNDO_TABLESPACE =

UNDO_TEST_ON_RAM;

Шаг 7. Повторно запускается несколько раз скрипт, который три раза подряд изменяет данные и далее запускает команду отката ROLLBACK, замеряется. Приведены результаты эксперимента, а также график работы системы при расположении UNDO как в оперативной памяти, так и на жестком диске.

В третьей главе проанализированы современные технологии корпорации Oracle для создания репликации в РБД. Oracle содержит семейство программных продуктов: Oracle Transportable Tablespaces, Модули Distributed SQL и Distributed Transactions, гетерогенный доступ посредством механизмов Transparent Gateways, Data Guard и Oracle Streams, которые отвечают эксплуатационным требованиям РС, таким как универсальность, надежность, готовность, удобство обслуживания, защищенность и управляемость.

Разработан алгоритм, увеличивающий производительность в неоднородных РБД при асинхронной репликации данных, этот алгоритм представлен в виде распределенного программного комплекса, проведен эксперимент по влиянию на производительность работы первичного узла при настроенном на нем асинхронной репликации объекта в однородных РБД.

Шаг 1. В каждом узле РБД создаются таблицы для каждого объекта системы. Эти таблицы содержат всю необходимую информацию для данных объектов.

Шаг 2. Для тех объектов, которые необходимо реплицировать на вторичные узлы РС, создаются таблицы изменений на первичном узле. Также в шлюзе устанавливается первоначальное временное значение t* в переменную t1.

Шаг 3. На первичном узле РС инициализируется транзакция и передается на выполнение. До фиксации результатов выполнения в таблицах, соответствующих объектов, срабатывает триггер. Триггер - процедура на событие операций изменение данных на объекте (вставка, изменение, удаление). Таким образом, в триггере объектов прописывается процедура, которая фиксирует изменения в

соответствующих таблицах изменений и после этого фиксируется результат транзакции.

Шаг 4. В шлюзе устанавливается интервал времени T* по истечению, которого шлюз обращается к таблицам изменения данных, забирая оттуда информацию, при условии того, что время данных (колонка “oper_date”) больше по сравнению со временем в переменной t1, а также записывает последнею дату измененных данных t** с колонки “oper_date” в переменную t2.

Шаг 5. Собранная информация с первичного узла посылается на обработку вторичным узлам.

Шаг 6. Каждый вторичный узел, получив информацию от первичного узла для обработки, приступает к ее применению. После успешной обработки вторичным узлом информации, присланный с первичного узла, вторичный узел сообщает об успешности прохождения операции первичному узлу. Первичный узел после получения положительного ответа от всех вторичных узлов проводит операцию фиксации (commit) и записывает значение переменной t2 в переменную t1. Далее идет переход на Шаг 3.

Шаг 7. В случае неполучения первичным узлом положительных ответов от всех вторичных узлов, первичный узел генерирует операцию отката (rollback) и далее идет переход на Шаг 3.

Далее в главе рассмотрен метод снятие снимков, который является одним из методов репликации, в этом случае снимается снимок системы на текущей момент и передается вторичным узлам. Смысл снимков заключается в том, что для многих систем возможно, а даже необходима обработка данных в том виде, в котором они были в определенный момент времени. К данной категории приложений относятся приложение для создания аналитических, статистических, а также разного рода отчетов.

Также рассмотрена технология RMAN, служащая для снятия бэкапов в ORACLE. Рассмотрен алгоритм инкрементального снятия бэкапа, широко использующегося в Oracle, и файл записи изменяющихся блоков данных. На основании данных технологий разработан алгоритм для повышения снятия снимков при использовании файла записи измененных блоков.

Шаг 1. Снимается первоначальный снимок Snapshot_1 с первичного узла и передается на вторичные узлы распределенной

системы. Получается полная копия первичного узла на вторичных узлах.

Шаг 2. Далее при прохождении каких-либо транзакций на первичном узле, вызывающие изменение данных, регистрируются все номера измененных блоков данных и записываются в файл измененных блоков, расположенных на первичном узле.

Шаг 3. При следующем снятии снимка Snapshot_N на первичном узле, вместо полного копирования Snapshot_N на первичный узел и передаче данного снимка на вторичные узлы, сперва идет обращение в файл измененных блоков для выяснения блоков, которые были изменены с последнего снятого снимка Snapshot_(N-1).

Шаг 4. После выяснения блоков, которые были изменены с последнего снятого снимка Snapshot_(N-1), происходит непосредственное копирование данных измененных блоков в Snapshot_N на первичном узле.

Шаг 5. Первичный узел передает блоки данных, которые были изменены с последнего снятого снимка Snapshot_(N-1) на вторичные узлы.

Шаг 6. Как на первичном узле, так и на вторичных узлах происходит уже локальное копирование недостающих неизмененных блоков данных с последнего снятого снимка Snapshot_(N-1).

В четвертой главе рассмотрено более подробно фундаментальное правило - независимость от фрагментации. Фрагментация в РС это разделение информации на несколько порций и распределение этих порций на различные узлы. В этом случае данные могут храниться в том узле, где они чаще всего используются, что позволяет достичь локализации большинства операций и уменьшения сетевого трафика. Благодаря этому получается выигрыш в эффективности всей РБД.

Рассмотрены различные виды проблем при использовании фрагментации и методы их решения:

- Как определить фрагментируемые объекты
- Где размещать информацию о критериях разделения фрагментируемых объектов
- Как происходит восстановление фрагментируемых объектов

- Как будет проходить транзакция, включающая в себя различные куски информации фрагментируемого объекта, расположенного на различных узлах.

Проанализированы типы фрагментаций. Разработаны алгоритмы фрагментации данных в распределенных базах данных, увеличивающие эффективность системы.

Шаг 1. В каждом узле РБД создаются таблицы для каждого фрагментируемого объекта системы. Эти таблицы содержат всю необходимую информацию для данных объектов. В каждом узле имеется свой словарь данных, в нем описаны все объекты узла, своего рода справочник всех объектов. В справочниках узлов предложено создание дополнительной таблицы – таблица условий фрагментируемых объектов. Данная таблица будет содержать информацию о фрагментируемых объектах, условия фрагментации объектов и соответственно номера узлов, в которых расположена информация, удовлетворяющая данным условиям.

Шаг 2. В РБД имеется множество диспетчеров транзакции, которые располагаются на узлах РС. На одном из узлов инициализируется транзакция, начинается процесс анализа транзакции. В зависимости от включаемой информации в запросе методом сравнения с колонкой condition таблицы условий фрагментируемых объектов выясняется подмножество узлов, которые будут участвовать в распределенной транзакции.

Шаг 3. Узел, инициирующий транзакцию посылает узлам, участвовавших в распределенной транзакции, SQL запрос на выполнение. Каждый узел после получения запроса, анализирует полученный запрос. В случае если запрос включает в себе условие, связанное с фрагментированным объектом, то с помощью таблицы условий фрагментируемых объектов берутся соответствующие условия фрагментации с колонки condition таблицы условий фрагментируемых объектов для данного узла. Далее условие в запросе, которое включает в себе условие, связанное с фрагментированным объектом, заменяется этим условием в обрабатываемом запросе.

Благодаря этому, увеличивается скорость обработки запроса, так как не приходится проверять условия, которые заведомо возвращают FALSE. Количество всех строк таблицы emp на узле_N равняется M. Условие column_name in (expression_1, expression_2, …. ,

expression_N), если для данного узла сравнение справедливо при последнем expression_N, то узлу_N придется проделать N*M операций сравнения. Но после замены условия с таблицы условий фрагментации объектов, узлу придется сделать всего лишь M операций сравнения. После идет непосредственное выполнение данного запроса. В случае положительного результата узел посылает сообщение инициирующему узлу результат выполнения и при этом фиксация не проделывается COMMIT. В случае же отрицательного результата узел посылает сообщение инициирующему узлу результат выполнения и проделывается откат транзакции ROLLBACK.

Шаг 4. В случае, получения инициирующим узлом транзакции положительных ответов от всех узлов, участвовавших в распределенной транзакции, узел посылает сообщения фиксации COMMIT этим узлам.

Шаг 5. В случае, неполучения инициирующим узлом транзакции положительных ответов от всех узлов, участвовавших в распределенной транзакции, узел посылает сообщения отката ROLLBACK этим узлам.

Как видно для увеличения производительности выполнения запросов нужно максимально уменьшить взаимодействие между узлами. На сегодняшний день стоимость оперативной памяти с каждым днем удешевляется, что приводит к возможности наращиванию ее на серверах. Благодаря этому все больше используются большие объемные буфера для кэширования данных на серверах.

Шаг 1. В каждом узле РБД введется СНИ (системный номер изменений) это глобальный идентификатор изменений, наряду с СНИ также и имеется ОСНИ (объектный системный номер изменений) – идентификатор изменений объекта.

Шаг 2. На узлах создаются таблицы запросов над фрагментируемыми объектами системы. Эти таблицы содержат информацию для определения ОСНИ всех объектов различных узлов содержащихся в запросе. Создаются таблицы запросов над фрагментируемыми объектами.

Шаг 3. Узел, инициирующий транзакцию проверяет на наличие желаемого исполняемого запроса Query(query_identification) в таблице определения запросов фрагментируемых объектов. Также нужно учесть, что в зависимости от размера буфера и настроенных

правил очистки буфера, данные нашего запроса могут быть очищены с кэш буфера системы. В случае нахождения данного запроса: если существует (Query(query_identification)), то проверяется состояние кэшируемых данных запроса в буфере и в случае успешного их расположения в буфере идет Шаг 4, в случае же не нахождения запроса или уже очистки нужных кэшируемых данных, то идет на Шаг 5.

Шаг 4. Посылается запрос всем остальным узлам, участвующих в распределенной транзакции, на получение ОСНИ фрагментируемого объекта Query_new(Node_OSCN). Получив ОСНИ фрагментируемого объекта, сравнивает с имеющимися у себя ОСНИ фрагментируемых объектов в таблице связки фрагментируемых запросов с узлами Query(Node_OSCN).

Если Query(Node_OSCN) = Query_new(Node_OSCN), то посылке на исполнение данного запроса и ожидания ответа от данного узла нет необходимости. Если же Query(Node_OSCN) <> Query_new(Node_OSCN), то посылает запрос на исполнения. При получении ответа, данные кэшируются и новый обновленный ОСНИ заносится в таблицу связки фрагментируемых запросов с узлами.

Шаг 5. Посылается запрос на исполнения всем необходимым узлам РС. При получении ответа данные запросы кэшируются, исполненный запрос заносится в таблицу определения запросов фрагментируемых объектов и ОСНИ заносится в таблицу связки фрагментируемых запросов с узлами.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

В настоящей диссертационной работе получены следующие основные результаты:

1. Рассмотрен и проведен анализ имеющихся работ в данном направлении, рассмотрение, которых имеет теоретический и практический интерес. Рассмотрены современные технологии и методы, используемые для поддержания фундаментальных правил распределенных систем. Проделаны экспериментальные сравнительные анализы влияния на производительность распределенной системы при использовании различных технологий.

2. Рассмотрены общие принципы распределенных баз данных. Проанализирован метод восстановления баз данных для обеспечения отказоустойчивости РБД. Обсуждены недостатки метода

“логирования для восстановления” при загрузке большого объема данных. Рассмотрены недостатки утилиты SQL*Loader при методе “прямой загрузки”. Разработан экономичный метод логирования для ускорения процесса восстановления базы данных при отказах.

3. Рассмотрен алгоритм ARIES восстановления узлов в распределенной базе данных. Предложен метод увеличения производительности системы при расположении UNDO табличного пространства в ОЗУ. Приведены результаты эксперимента, а также график работы системы при расположении UNDO как в оперативной памяти, так и на жестком диске.

4. Рассмотрена отказоустойчивость РС. Рассмотрены методы репликации применяемое в РС, а также метод снятие снимков позволяющий обработку данных в том виде, в котором они были в определенный момент времени в прошлом. Рассмотрен алгоритм инкрементального снятия бэкапа в Oracle и файл записи изменяющихся блоков данных. На основании данных технологий разработан алгоритм, для повышения снятия снимков при использовании файла записи измененных блоков.

5. Разработан алгоритм асинхронной репликации данных, увеличивающий производительность системы распределенных баз данных. Разработано программное обеспечение (шлюз), для обеспечения асинхронной репликации в неоднородной распределенной базе данных. Анализированы современные технологии корпорации Oracle для создания репликации в распределенных базах данных. Проведен эксперимент по влиянию на производительность первичного узла при включении асинхронной репликации.

6. Разработан алгоритм фрагментации данных в распределенных базах данных, увеличивающий эффективность системы. Рассмотрены современные технологии корпорации Oracle для создания фрагментации в распределенных базах данных. Проведен эксперимент по влиянию на эффективность распределенной системы при использовании фрагментации.

7. Предложенные алгоритмы были использованы и реализованы при построении распределенной системы процессингова центра Азерикард, являющимся полностью сертифицированным Международными платежными системами MasterCard, Visa, American Express и Diners Club.

Основные положения диссертации опубликованы в следующих работах:

1. **Самедов С.Б.** Исследование проблемы отказоустойчивости распределенных систем / XIV Республиканская научная конференция докторантов и молодых ученых, Баку, 2010, с. 66.
2. **Самедов С.Б.** Модифицированный метод логирования для восстановления базы данных в отказоустойчивых распределенных системах // Научно-технический журнал «Информационные технологии моделирования и управления», Воронеж, 2009, № 7(59), с. 968-971.
3. **Самедов С.Б.** Механизмы обеспечения отказоустойчивости в распределенных базах данных / Республиканская научная конференция посвещенная 90-ю БГУ «Современные проблемы математики, информатики и экономики», Баку, 2009, с. 147-152.
4. **Самедов С.Б.** Технология DATA GUARD в распределенных системах / 1-ая Международная научно-практическая конференция «Информационные системы и технологии в таможенной сфере», Баку, 21-22 декабря, 2009, с. 60-62.
5. **Самедов С.Б.** Технологии обеспечения отказоустойчивости в распределенных системах / XV Республиканская научная конференция докторантов и молодых ученых, Баку, 2010, с. 190-191.
6. **Самедов С.Б.** Повышение эффективности распределенных систем при репликации данных с первичных узлов на вторичные узлы // Научно-технический журнал «Информационные технологии моделирования и управления», Воронеж, 2010, № 6(65), с. 829-836.
7. **Самедов С.Б.** Кластерные технологии баз данных / Международная научная конфереция «Современные проблемы информатизации в анализе и синтезе программных и телекоммуникационных систем», Сборник трудов выпуск 15, Воронеж, 2010, с. 403-404.
8. **Самедов С.Б.** Повышение эффективности распределенных систем транзакционной обработки информации // Вестник Бакинского Университета, серия физико-математических наук, Баку, 2011, № 1, с. 106-112.

9. **Самедов С.Б.** Повышение производительности в однородных и неоднородных распределенных базах данных при репликации данных / 3-ая международная, научно-практическая конференция «Информационные системы и технологии в таможенной сфере», Баку, 2011, с. 41-42.
10. **Самедов С.Б.** Асинхронная репликация данных для увеличения производительности в однородных и неоднородных распределенных базах данных // Известия НАНА, серия физико-технических и математических наук, том XXXII информатика и проблемы управления, Баку, 2012, № 3, с.68-75.
11. **Самедов С.Б.** Комплексный подход к созданию распределенных систем // Вестник Бакинского Университета, серия физико-математических наук, Баку, 2011, № 4, с. 101-110.
12. **Самедов С.Б.** Повышение производительности работы распределенных систем базы данных / XVI Республиканская научная конференция докторантов и молодых ученых, Баку, 2011, с. 157-159.
13. **Самедов С.Б.** Повышение производительности в распределенных системах при использовании фрагментации данных // Институт Информационных Технологий, научно-практический журнал «Проблемы информационных технологий», Баку, 2012, № 2(4), с. 31-37.
14. **Самедов С.Б., Алиев А.А.** Разработка алгоритмов, повышающих производительность распределенных систем обработки информации // «Системы управления и информационные технологии», научно-технический журнал, перспективные исследования, Москва, 2012, № 2.1(48), с.175-179.
15. **Samadov S.B.** Increase the effectiveness of distributed database systems / Eighth international scientific-practical conference, «Internet-Education-Science» (IES), Vinnytsia, Ukraine, 2012, p. 131.

Роль соискателя в трудах, опубликованных в соавторстве:
[14]-разработан алгоритм и программное обеспечение

S.B.SƏMƏDOV

Paylanmış sistemlərin məhsuldarlığını artıran alqoritmlərin və proqram vasitələrinin işlənməsi və tədqiqi

X Ü L A S Ə

Dissertasiya işində paylanmış sistemlərin səmərəliliyini artıran alqoritmlər və proqram təminatı işlənib. Paylanmış sistemlər olmadan müasir informasiya texnologiyalarını təsvir etmək mümkün deyil.

Müasir şəbəkələrə qarşı irəli sürülən tələblər, müasir kompüter şəbəkələrinin səmərəliliyinin əsas xüsusiyyətləri gözdən keçirilib. Paylaşnmış sistemlər öyrənilib, informasiyanının emalının paylanmış sistemlərinin əsas qurulma prinsipləri araşdırılıb, paylanmış sistemlərin fundamental qaydaları öyrənilib. Hər bir qayda araşdırılıb və paylanmış sistemin səmərəliliyinə təsir göstərən qaydalar ayrılıb.

Paylanmış sistemlərdə informasiyanın emalının effektivliyini artıran alqoritmlər işlənib. Verilənlərin fraqmentləşdirmə və replikasiya alqoritmləri və paylanmış sistemlərin səmərəliliyini artıran proqram təminatı hazırlanıb. Təcrübələr aparılıb və alınan nəticələr təhlil olunub.

Alınmış elmi nəticələr Azərbaycanda MasterCard, Visa, American Express və Diners Club Beynəlxalq Ödəmə Sistemləri tərəfindən tamamilə sertifikatlaşdırılmış AzeriKard prossesinq mərkəzində isitismar olunan proqram təminatında istifadə edilmişdir.

S.B.SAMADOV

Investigation and development algorithms and software improves the performance of distributed systems

S U M M A R Y

In this work developed algorithms and software that enhance the performance of distributed systems without which it is very difficult to imagine a modern information technology.

Consideration of the requirements for today's networks, the basic performance characteristics of the modern computer networks. The distribution of the system, investigated the general principles for distributed data processing systems have been studied fundamental rules of distributed systems. Investigated each rules and regulations affecting the identified performance distributed system.

An algorithm to improve the efficiency of processing distributed information systems. The algorithms fragmentation, replication of data and software, enhancing the performance of distributed systems. Experiments was done and analyzed these results.

Archived scientific results have been applied in the software of AzeriCard processing center. AzeriCard is fully certified by MasterCard, Visa, American Express and Diners Club International Payment Systems.

AZƏRBAYCAN MİLLİ ELMLƏR AKADEMİYASI
İNFORMASİYA TEXNOLOGİYALARI İNSTİTUTU

Əlyazması hüququnda

SAMİR BƏXTİYAR OĞLU SƏMƏDOV

PAYLANMIŞ SİSTEMLƏRİN MƏHSULDARLIĞINI ARTIRAN ALQORİTMLƏRİN VƏ PROQRAM VASİTƏLƏRİNİN İŞLƏNMƏSİ VƏ TƏDQİQİ

İxtisas: 3338.01– sistemli analiz, idarəetmə və informasiyanın işlənməsi

Texnika üzrə fəlsəfə doktoru alimlik dərəcəsi almaq üçün təqdim edilmiş dissertasiyanın

AVTOREFERATI

Bakı - 2013