

Описание работы web-сервиса

Максимова Наталья Александровна
Смоленский государственный университет
к.п.н., доцент

Аннотация

Сервис-ориентированные приложения – это распределенные приложения, функциональные элементы которых – веб сервисы - исполняются на разных серверах и формируют единое целое, управляемое клиентом. Web-Сервис – это приложение, которое доступно через Интернет, и предоставляющее услуги определенного вида, форма которых независима от поставщика (так как в них применяются универсальные платформы функционирования и форматы данных). Имеют место несколько разного вида технологий, которые поддерживающих теорию распределенных объектных web-систем.

Ключевые слова: сервис-ориентированные приложения, теория распределенных объектных web-систем, параллельное программирование.

Maksimova Natal'ja Aleksandrovna
Smolensk State University
PhD, associate Professor

Abstract

Service-oriented applications is distributed applications, functional elements which web services are executed on different servers and form a single unit, controlled by the client. Web Service is an application, which is available via the Internet and providing services of a certain type, the shape of which provider-independent (since they use a universal platform operation and data formats). There are several different types of technologies that support a theory of distributed object web system.

Keywords: service-oriented application, the object theory of distributed web-based systems parallel programming.

Для параллельной реализации операции Join может быть использован программно-аппаратный комплекс на основе вычислительной сети: локальная сеть, интранет, интернет. Архитектура такого комплекса (схема передачи данных между вычислительными узлами) показана на рисунке 1 [1,2].

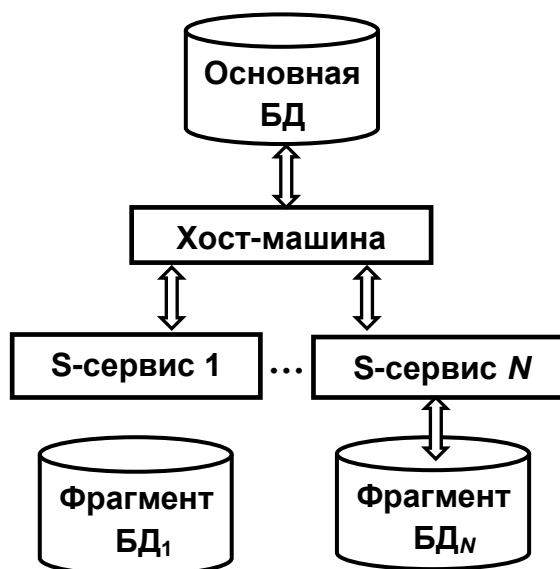


Рис. 1. Программно-аппаратный комплекс для обработки симметрично горизонтально распределенных данных

Основная БД располагается на узле-клиенте (хост-машине), который взаимодействует по данным и управлению с узлами-серверами, реализующими обработку фрагментов WEB-сервисами (на рисунке они названы S-сервисами). S-сервис содержит необходимые для работы API-контроллеры, обрабатывающие GET и POST запросы от клиента (хост-машины). Реализованы API-контроллеры трех типов, со следующими функциями:

Тип 1. Инициализации фрагмента БД.

Тип 2. Прием, распаковка и добавление в принятого по сети блока.

Тип 3. Выполнение операции Join над таблицами-операндами БД-фрагмента.

По завершении всех действий, связанных с выполнением операции Join, хост-машина принимает результаты от всех S-сервисов [3,4,5].

Операнды каждой пары генерировались таким образом, что содержали классы эквивалентности с одинаковыми значениями экземпляров множества ключей, но различными количествами записей. На каждом шаге четыре пары одновременно обрабатывались четырьмя параллельными процессами, результаты которых объединялись в один файл-результат (рисунок 2).

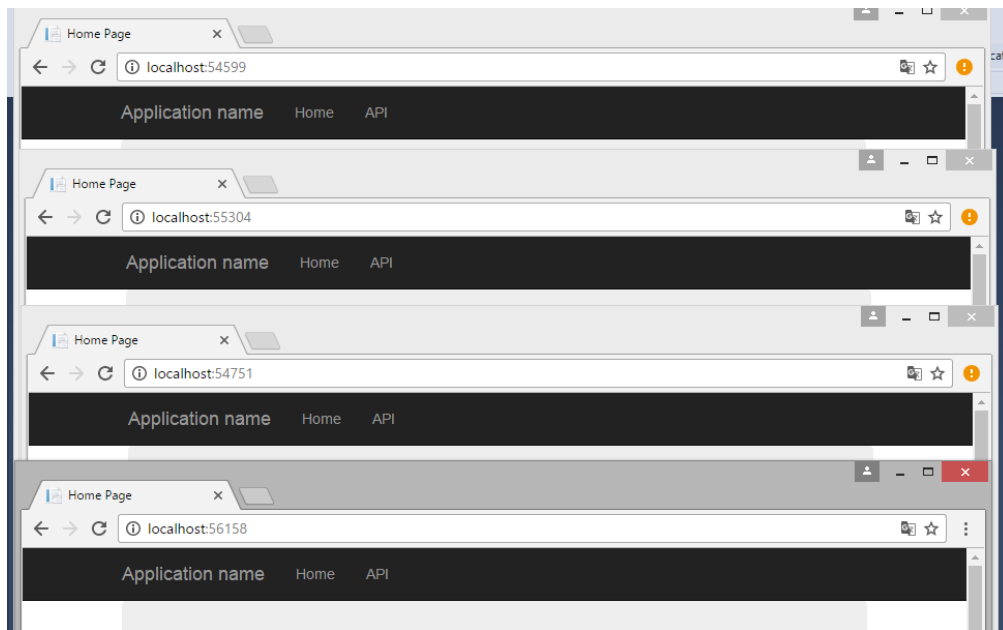


Рисунок 2. Подключение машин серверов

Фрагменты таблиц, сформированных на сервере 1 и 2 представлены на рисунке 3.

	A
1	AAABA
2	AAABA
3	AAABA
4	AAABA
5	AAABA
6	AAABA
7	AAABA
8	AAABA
9	AAABA
10	AAABA
11	AAABA
12	AAABA
13	AAABA
14	AAABA
15	AAABA
16	AAABA
17	AAABA
18	AAABA
19	AAABA
20	AAABA
21	AAABA
22	AAABA
23	AAABA
24	AAABA
25	AAABA
26	AAABA
27	AAABA

	A
1	AABBB
2	AABBB
3	AABBB
4	AABBB
5	AABBB
6	AABBB
7	AABBB
8	AABBB
9	AABBB
10	AABBB
11	AABBB
12	AABBB
13	AABBB
14	AABBB
15	AABBB
16	AABBB
17	AABBB
18	AABBB
19	AABBB
20	AABBB
21	AABBB
22	AABBB
23	AABBB
24	AABBB
25	AABBB
26	AABBB
27	AABBB

Рисунок 3. Фрагменты таблиц

Очевидно, что наборы данных формируются разные. После запуска данные распределяются следующим образом. На сервере 1 – 5014, сервере 2 – 5018, сервере 3 – 4999, сервере 4 – 4936 [6].

Основное окно запуска программы представлено на рисунке 4.

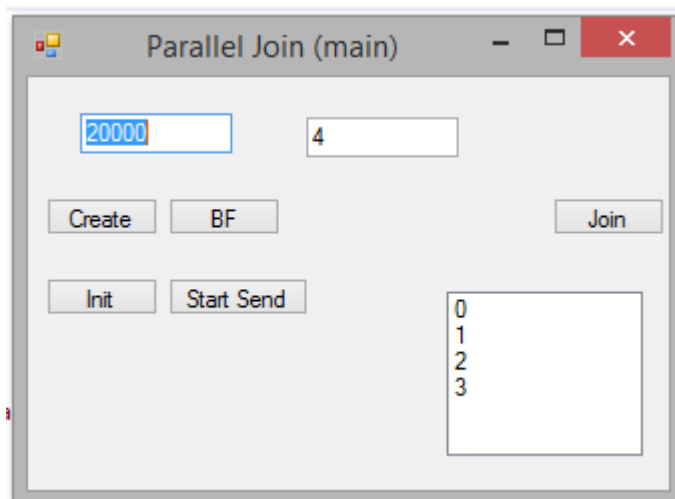


Рисунок 4. Основное окно

На данном этапе мы задаем количество данных и количество процессор (серверов) для их обработки и запускаем процесс создания основной базы, нажав на Create. После завершения данного процесса запускаем на выполнения алгоритм бустрофедона (приложение 2). В результате у нас происходит формирование четырех таблиц ABIST1, ABIST2, ABIST3, ABIST4 (рисунок 5,6,7,8). В данных таблицах находятся уже картежи данных.

	A	mA	mB	mm
▶	ABBAB	618	601	371418
	BAAAB	651	631	410781
	BAABA	590	648	382320
	ABVBA	671	593	397903
*	NULL	NULL	NULL	NULL

Рисунок 5. Таблица ABIST1

	A	mA	mB	mm
▶	ABBBB	587	575	337525
	BBBBB	690	650	448500
	AABAA	608	613	372704
	AAAAB	596	680	405280
	AAAAA	615	624	383760
	BBAAA	623	637	396851
*	NULL	NULL	NULL	NULL

Рисунок 6. Таблица ABIST2

	A	mA	mB	mm
▶	BBBBA	613	575	352475
	ABABA	635	680	431800
	ABAAA	583	643	374869
	AABBA	626	646	404396
	ABVAA	647	595	384965
	BAAAA	632	624	394368
*	NULL	NULL	NULL	NULL

Рисунок 7. Таблица ABIST3

	A	mA	mB	mm
▶	BBABB	572	627	358644
	BABBA	665	635	422275
	ABAAB	631	595	375445
	ABABB	624	647	403728
	BBBAB	642	602	386484
	AABBB	631	623	393113
*	NULL	NULL	NULL	NULL

Рисунок 8. Таблица ABIST4

После этого мы отсылаем данные наборы данных на серверы для обработки. Время обработки рисунок 9.

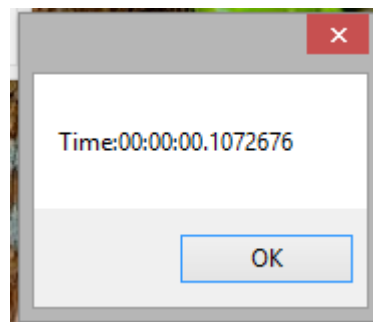


Рисунок 9. Время обработки запроса

На рисунках 10-14 представлены варианты ответа программы и время обработки данных

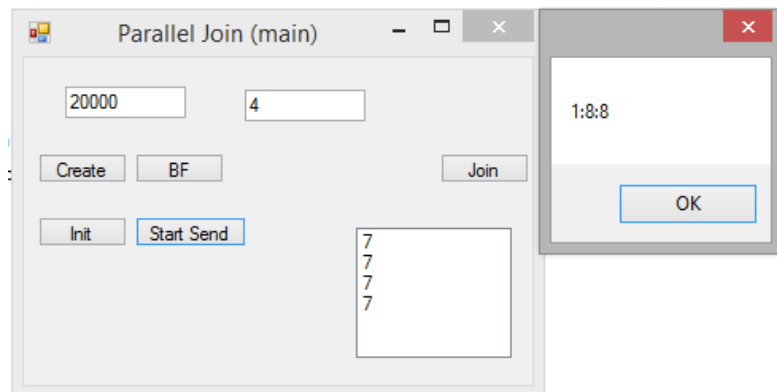


Рисунок 10. Первый сервер

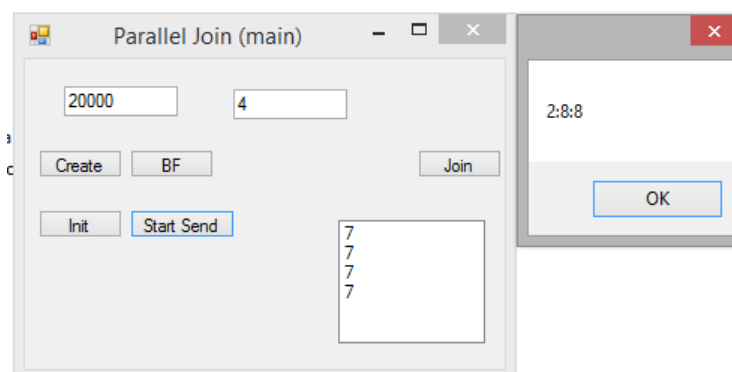


Рисунок 11. Второй сервер

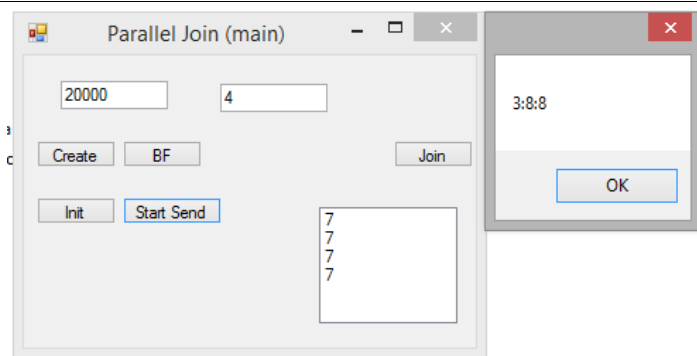


Рисунок 12. Третий сервер

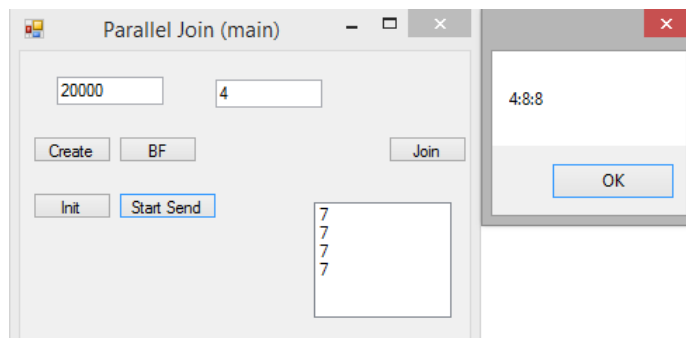


Рисунок 13. Четвертый сервер

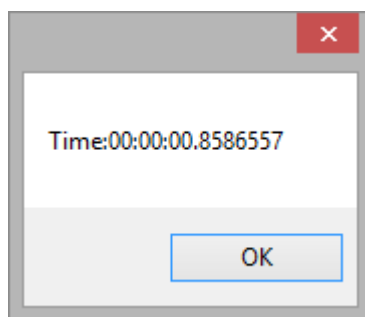


Рисунок 14. Время обработки запроса

Разработанная архитектура была реализована на локальной сети. Экспериментальный анализ показал, что время симметричного горизонтального распределения двух таблиц-операндов большого объема незначительно (не более чем на 5%) увеличивает время параллельного выполнения операции Join, которое в разы меньше времени ее последовательного выполнения.

За последние годы, распределенные и параллельные БД стали реальностью. Они предоставляют функциональность централизованных БД, но в такой среде, где данные распределены между компьютерами, связанными сетью, или между узлами многопроцессорной системы. Распределенные БД допускают естественный рост и расширение баз данных путем простого добавления в сеть дополнительных машин. Подобные системы обладают более привлекательными характеристиками «цена/производительность», благодаря современным прогрессивным сетевым технологиям. Параллельные СУБД – это, пожалуй, единственный

реалистичный подход для удовлетворения потребностей многих важных прикладных областей, которым необходима исключительно высокая пропускная способность баз данных. Поэтому при проектировании параллельных и распределенных СУБД следует предусмотреть в них соответствующие протоколы и стратегии обработки, направленные на достижение высокой производительности.

Сервис-ориентированная архитектура (СОА) — одно из последних решений многих проблем современных средних и крупных систем, которое поддерживают и развивают практически все современные поставщики программного обеспечения. Несмотря на небольшой возраст технологии (первые стандарты появились в 2003 году), СОА нашла применение, как в корпоративном, так и государственном секторах.

Библиографический список

1. Максимова Н.А. Сервис-ориентированная архитектура: от концепции к применению // NovaInfo.Ru. 2016. Т. 4. № 44. С. 19-22.
2. Максимова Н.А. Разработка приложений на основе сервис-ориентированных систем // NovaInfo.Ru. 2016. Т. 3. № 46. С. 41-44.
3. Мунерман В.И. Объектно-ориентированная модель массовой обработки данных // Системы высокой доступности. 2011. Т. 7. № 4. С. 72-74.
4. Мунерман В.И. Реализация обработки больших объемов данных на симметричных мультипроцессорных системах // Системы высокой доступности. 2013. Т. 9. № 2. С. 036-039.
5. Мунерман В.И., Мунерман Д.В. Анализ обработки больших объемов данных на симметричных мультипроцессорных системах // Системы компьютерной математики и их приложения. 2013. № 14. С. 87-91.
6. Мунерман В.И., Мунерман Д.В. Алгебраический подход к построению программно-аппаратных комплексов для повышения эффективности массовой обработки данных // Современные информационные технологии и ИТ-образование. 2015. Т. 2. № 11. С. 391-396.