

УДК 004.424

К.А. Рухлис, А.Е. Дорошенко

К ВОПРОСУ О ПРОИЗВОДИТЕЛЬНОСТИ РАСПРЕДЕЛЕННЫХ ХРАНИЛИЩ ОБЪЕКТОВ ДАННЫХ В ПАМЯТИ ОЗУ

Проанализированы основные преимущества современных распределенных хранилищ объектов данных в памяти ОЗУ. Представлены описания основных платформ таких систем, приведён анализ их характеристик. Проведены группы экспериментов по замеру производительности нескольких систем для типичных приложений. Проанализированы результаты экспериментов.

Введение

Одной из основных проблем при обработке больших объёмов данных на вычислительной технике всегда была производительность системы ввода-вывода хранилищ данных. Исторически, для хранения больших объёмов данных использовались хранилища на основе магнитных лент/жёстких дисков. Такие хранилища обладают сравнительно низкой стоимостью хранения данных. При этом производительность подсистемы ввода-вывода остаётся бутылочным горлышком у этого типа устройств.

С распространением 64-х битных многоядерных систем стоимость оперативной памяти значительно снизилась, а объёмы многократно увеличились. Это делает идею использования ОЗУ для хранения операционных данных вместо жёстких дисков очень привлекательной, учитывая тот факт, что скорость доступа к ОЗУ в 1000 раз превышает скорость доступа к HDD [1]. Группировка узлов с большими объёмами ОЗУ в единую вычислительную систему-кластер фактически полностью исключает потребность приложений в доступе к медленным устройствам ввода-вывода. Все эти факторы породили создание новых подходов и новых решений кластеризации – распределённых хранилищ объектов данных (*In-Memory Compute Grid* и *In-Memory Data Grid, IMDG*) [2], ещё называемые дата грид.

Ключевыми особенностями, способствующими распространению IMDG, являются [3, 4]:

- скорость доступа к данным для

чтения/записи – скорость доступа к ОЗУ в 1000 раз превосходит скорость доступа к лучшим HDD;

- масштабируемость – в большинстве случаев в лучших реализациях IMDG масштабируемость фактически линейна;

- простота внутреннего представления данных – подавляющее большинство IMDG платформ хранят данные в виде пары <ключ, значение>. Отсутствие сложностей в хранении, комплексных иерархических структур дают значительный прирост производительности и удобства разработки прикладных приложений, позволяя программистам хранить данные в наиболее удобном для прикладных задач виде;

- поддержка **Transactional ACID** принципа – современные системы IMDG используют оптимизированное 2-х фазное подтверждение операций (2-phase-commit) при изменении данных для гарантирования целостности и непротиворечивости данных. В этом их ключевое отличие от NoSQL баз данных, которые используют принцип “Согласованность в конечном счете (Eventual consistency)” при изменении данных (что позволяет NoSQL иметь в какой-то момент времени несогласованные данные).

- секционирование данных (Partitioning) – у лучших реализаций IMDG систем хранение пар <ключ, значение> оптимизировано для минимизации пересылки данных между разными узлами кластера.

В свою очередь реализации IMDG, в основном, различаются [4, 5]:

- механізмами реалізації **Transactional ACID**;

- засобами первичної завантаження даних (прогрів кеша);

- наявністю і реалізацією паралельного секціонування;

- політиками вивантаження даних (eviction policies) – при переповненні кеша, при відсутності потреби в самих даних на протязі якого-то моменту і т. д.);

- підтримкою SQL-like запитів.

Вибір той чи іншої реалізації залежить в першу чергу від програм, які будуть оперувати зберіганими в IMDG даними.

1. Описання задачі, для якої потрібався вибір конкретної платформи IMDG

Одним з основних застосувань IMDG систем є клас програм, де основною операцією введення-виведення є читання, а не запис. Прикладами таких програм є пошукові системи, системи агрегації і вирахувань на множині даних. Фактично, IMDG в даному випадку використовується як кеш 2-го рівня. Схематично такі програми можна представити наступним чином (рис. 1).

Для порівняння продуктивності був використаний набір даних з 20 мільйонів унікальних записів. Кластер IMDG складається з 4 вузлів. Як постійне сховище використано Oracle 11g. Кожен фізичний вузол має наступні характеристики:

- процесор: Intel Xeon 2.4GHz 2 Cores;
- ОЗУ: 16GB;
- операційна система: CentOS x64;
- Java: 8.x.

Інтерфейс з клієнтською програмою побудований на базі REST запитів. Загальна схема кластера показана на рис. 2.

Тестування проводилося в 2-х режимах – з “прогретим кешем” і з “холодним стартом”. В випадку з прогретим



Рис. 1. Общая структурная схема приложения IMDG

кешем-записи предварительно во время старта узлов в соответствии с разбивкой вычитывались в кеш. Так, что во время запросов от клиентского приложения обращения к БД сведены к минимуму. В случае холодного старта – после старта узлов в их кеше не содержатся данные. Данные в этом случае попадают в кеш при первом обращении к соответствующей записи.

Клиентское тестовое приложение состоит из конфигурируемого количества независимых клиентских потоков с запросами к IMDG на загрузку 500 взаимосвязанных записей. В процессе тестирования мы использовали два набора тестов – один, когда количество клиентских потоков составляет 3 экземпляра; и второй, когда мы измеряли обращения к IMDG от 5 до 20 клиентских потоков с 5000 запросов одновременно. Для сглаживания возможных инфраструктурных накладок для каждой из платформ тестирование запускалось 5 раз. Результаты – среднее арифметическое из 5 запусков для каждого из тестовых сценариев.

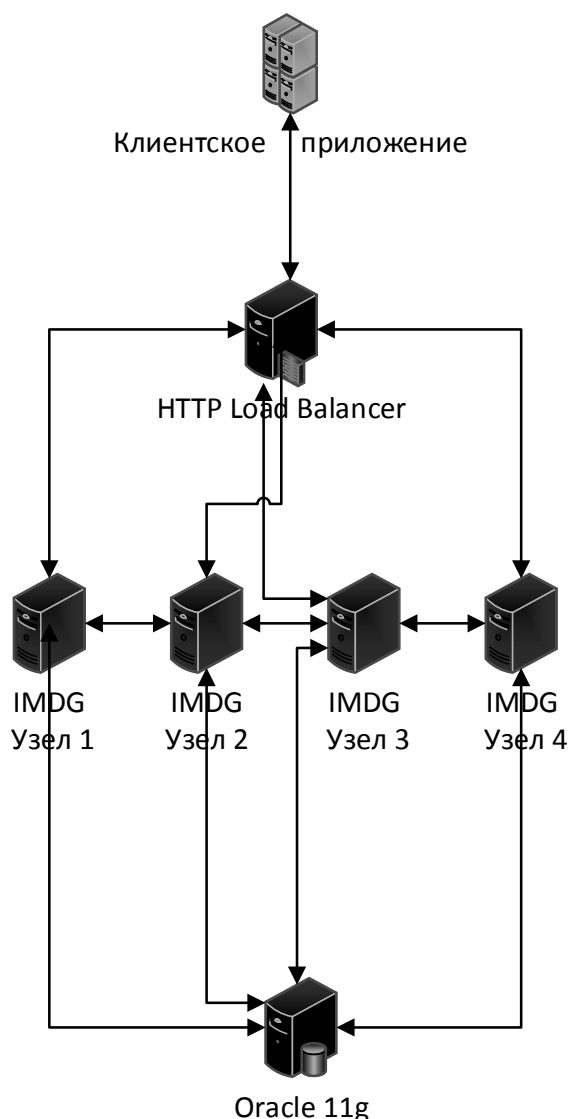


Рис. 2. Общая схема кластера

2. Описание выбранных для тестирования платформ

В качестве целевых IMDG платформ тестировались: RH Infinispan, Oracle Coherence, Ehcache (Terracotta commercial version) и HazelCast. Все эти реализации IMDG полностью удовлетворяют поставленной задаче, имеют достаточно длительную историю развития и популярность.

RH Infinispan [6, 7] представляет собой распределённую систему хранения пар <ключ, значение> с реализацией всех вышеописанных принципов IMDG. Ключевыми особенностями Infinispan являются:

- максимальная масштабируемость;

- гарантированная доступность его сервисов;
- максимальное использование многопоточности на современных CPU;
- развитые неблокирующие (lock-free) механизмы репликации данных.

Infinispan часто используется не только в качестве IMDG платформы, но и как обычный распределённый кеш, высокопроизводительное объектное хранилище, NoSQL хранилище пар <ключ, значение>. Существует также и коммерческая версия Red Hat JBoss Data Grid, которая отличается наличием коммерческой технической поддержки Red Hat, возможностью получать самые свежие и последние версии, возможностью адаптации под клиента.

Oracle Coherence [8] является ведущей коммерческой IMDG платформой. Кроме стандартного набора функций IMDG эта реализация включает в себя развитые средства мониторинга IMDG кластера, возможности гибкой настройки очищения/сброса кеша, гарантированную доступность сервисов платформы не зависимо от нагрузки, богатые возможности горячей замены узлов.

Ehcache [9, 10] – один из старейших фреймворков для создания кеша под Java. Первые версии этого кеша предназначались для использования в качестве кеширующей надстройки над различными Java ORM библиотеками для ускорения операций с часто используемыми сущностями без необходимости обращения к физической СУБД. С тех пор Ehcache был значительно функционально расширен, появились возможности использования миллионов кешей в рамках Ehcache кластера, добавились средства гарантирования доступности сервисов, несколько различных вариантов репликации данных, динамическое конфигурирование/реконфигурирование кешей, плюс, зеркалирование узлов, гибкая настройка непротиворечивости данных и средств для достижения этого. Кроме того, Ehcache – единственная платформа, которая поддерживает пользовательские надстройки/замещения для любого своего модуля.

HazelCast [11, 12] – предназначен для создания высокопроизводительных кластерных систем на Java. IMDG – всего лишь часть этой платформы. Среди отличительных черт HazelCast можно выделить:

- распределённую реализацию java.util.concurrent.ExecutorService и java.util.concurrent.locks.Lock;
- высокоэффективную реализацию принципа publish/subscribe уведомлений;
- транзакционность и соответствующую интеграцию с JEE контейнерами;
- механизм распределённых слушателей/событий;
- возможность пользовательским приложениям реагировать на изменение состава узлов кластера;
- динамическую кластеризацию HTTP сессий;
- динамическое масштабирование до сотен серверов;
- динамическое секционирование данных и их резервное хранение;
- динамическое восстановление работоспособности;
- компактность и лёгкость интеграции в любую существующую инфраструктуру;
- полностью внутренне многопоточный;
- оптимизирован для применения на современных CPU x64 и для работы с большими объёмами малых данных.

3. Описание результатов тестирования

В процессе построения экспериментального кластера на каждой из вышеперечисленных платформ оценивались несколько параметров не связанных с производительностью самой платформы в тестовой задаче. Среди них – удобство установки и развёртывания, нагрузка на CPU во время работы, возможности динамического конфигурирования параметров кластеров и т. д. Во время тестирования все

платформы правильно секционировали данные и нагрузка на узлы была равномерной. Результаты сравнения по этим параметрам приводятся в табл. 1.

Таблица 1. Сводная таблица характеристик тестируемых IMDG платформ. 100 % – максимальный показатель

Свойство\ Платф-ма	Infini span	Coherence	Ehcache	Hazel Cast
Инсталляция	70%	85%	75%	90%
Конфигурирование	50%	70%	65%	70%
Поддержка	60%	70%	60%	70%
Средства репликации	70%	90%	70%	75%
Нагрузка на CPU, максимум	80%	85%	75%	80%

Было проведено 3 группы тестов:

1. Первым тестом стал запуск 3 клиентских потоков с запросом со стороны каждого клиента пятисот записей. Тесты выполнялись на “холодном кеше”, т. е. без предварительной загрузки данных из СУБД. Результаты приведены в табл. 2.

Таблица 2. Сводная таблица результатов первого теста

IMDG платформа	Запросов в секунду	Ошибок чтения, %	Попадание в кеш, %
Infinispan	289	1.5	57
Oracle Coherence	400	8	99
Ehcache	280	5	80
HazelCast	292	1.3	95

Как можно увидеть из этих результатов – лучшую производительность на чтение в условиях холодного кеша проде-

монстрировала платформа Oracle Coherence.

2. Для проверки на сколько изменится производительность платформ с предварительно прогретым кешом при прочих равных условиях проведём второй тест (табл. 3).

Таблица 3. Сводная таблица результатов второго теста

IMDG платформа	Запросов в секунду	Ошибок чтения, %
Infinispan	488	0.02
Oracle Coherence	643	0.015
Ehcache	402	0.15
HazelCast	503	0.02

Как можно увидеть, с “прогретым кешем” производительность операций случайного чтения всех датагрид платформ выросла в среднем в 1.5 раза. Лучше всего, как и в прошлый раз, показал себя Oracle Coherence.

3. Для тестирования производительности операций чтения с переменным количеством клиентских запросов мы провели третий тест. Во время теста мы перезапускали клиентское приложение с различным количеством одновременных клиентских потоков, начиная с 5, с шагом в 5. Каждый поток делает запрос на загрузку 5000 записей. Кеш на всех датагрид платформах был предварительно прогрет при старте. Результаты можно увидеть на следующем графике (рис. 3).

Как можно увидеть из диаграммы, все платформы справились с нагрузкой. Производительность операций чтения слабо изменялась в зависимости от количества одновременных клиентских потоков. Лучше всего сбалансировать нагрузку смогли Coherence (в районе 647–650 запросов в секунду) и HazelCast (в районе 517–520 запросов в секунду).

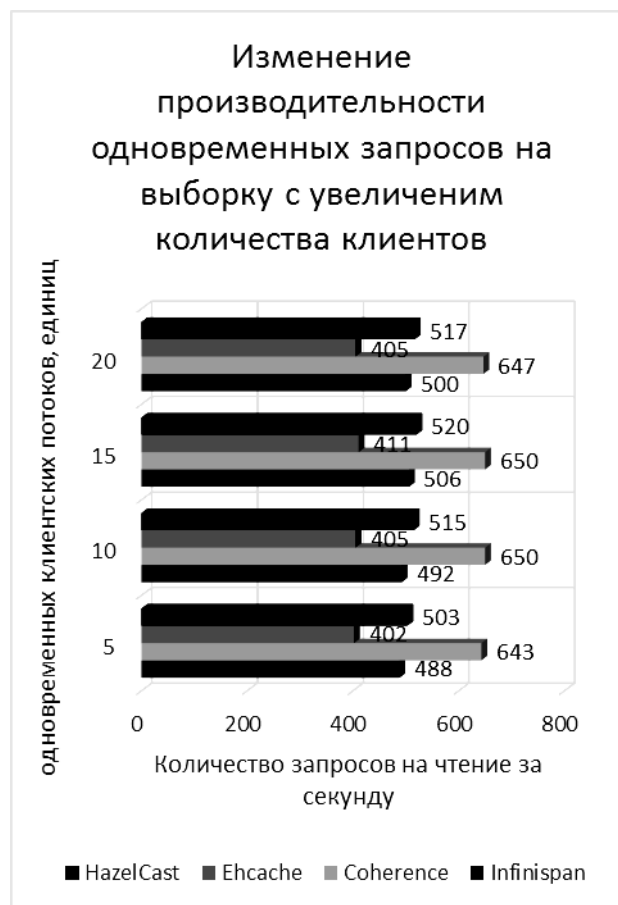


Рис. 3. Изменение производительности запросов на чтение кластеров IMDG в зависимости от количества клиентских потоков

Заключение

Современные платформы In-Memory Data Grid – высокопроизводительные системы, успешно заменяющие реляционные и NoSQL СУБД в задачах с требованиями максимально быстрых запросов к данным. Эти системы нашли повсеместное использование в таких сферах, как поисковые запросы данных в интернет, анализ и агрегация научных данных, приложениях электронной коммерции (например, для хранения и быстрого доступа к состоянию склада по товарам). Современные IMDG платформы научились автоматически балансировать нагрузку, оптимально секционировать данные, гарантируют целостность данных, самовосстановление отдельных узлов и т. д. Конкретно под нашу задачу из протестированных датагрид систем лучше всего подходят Oracle Coherence и HazelCast. Эти си-

стемы демонстрируют наилучшую масштабируемость, оптимальную автобалансировку нагрузки, а также оказались наиболее оптимальными при выполнении запросов к данным, которые отсутствуют в кеше.

1. *Falling RAM Prices Drive In-Memory Database Surge*: <http://blogs.sap.com/innovation/big-data/ram-prices-drive-in-memory-surge-020097>.
2. *Taxonomy, Definitions and Vendor Landscape for In-Memory Computing Technologies*: <http://www.gartner.com/document/2608719>.
3. *What IT Leaders Need to Know About In-Memory Data Grids*: <http://www.gartner.com/document/code/231619>.
4. *Plattner H., Zeier A. In-Memory Data Management: Technology and Applications*. 2nd ed., Springer, 2012. – 267 p.
5. *Di Stefano M. Distributed Data Management for Grid Computing*. Wiley, 2005. – 312 p.
6. *RH Infinispan*: <http://infinispan.org/>
7. *Marchioni F. Infinispan Data Grid Platform*. Packt Publishing, 2012. – 150 p.
8. *Oracle Coherence In-Memory Data Grid*: <http://www.oracle.com/ru/products/middleware/cloud-app-foundation/coherence/overview/index.html>.

9. *Ehcache*: <http://ehcache.org/>
10. *Wind D. Instant Effective Caching with Ehcache*. Packt Publishing, 2013. – 86 p.
11. *Hazelcast*: <http://hazelcast.com/>
12. *Johns M. Getting Started with Hazelcast*. Packt Publishing, 2013. – 136 p.

Получено: 12.05.2015

Об авторах:

Рухлис Константин Александрович,
младший научный сотрудник,

Дорошенко Анатолий Ефимович,
доктор физико-математических наук,
профессор, заведующий отделом теории
компьютерных вычислений.

Место работы авторов:

Институт программных систем
НАН Украины,
03680, Киев-187,
Проспект Академика Глушкова, 40.
Тел.: (044) 526 3559.
E-mail: ukkr@yandex.ru,
dor@isofts.kiev.ua