

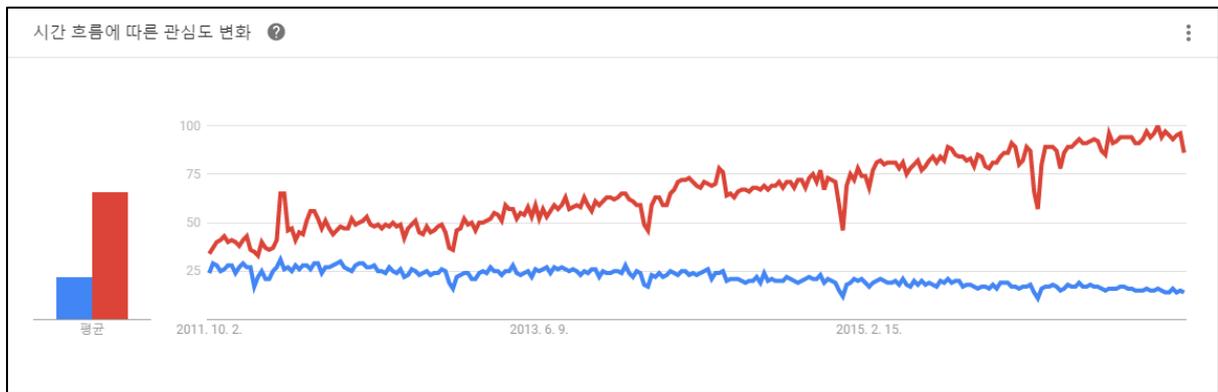
캐시서버인 Redis 를 MySQL/MariaDB 플러그인으로 개발

# MySQL Redis Plugin

공개 SW 개발자 Lab 오픈소스프론티어 2 기 이형채

MySQL 5.6 이전까지는 Memcached가 UDF(User Defined Function) 형태로 제공되었으나, MySQL 5.6부터는 플러그인(Plugin) 형태로 제공되었다. 이 시점에서 Memcached + MySQL 보다는 Redis + MySQL이 더 많이 사용되기 시작했다. 불과 몇 년 사이에 Key-Value Database 1위와 2위 자리가 뒤바뀌었고, 그 격차는 더 심화되었다.

그에 반해서 Oracle MySQL, MariaDB, Percona는 Redis Plugin을 개발하지 않고 있다. 이는 예전처럼 Memcached, Redis Key-Value Database를 빌트인 캐시서버(Built-in Cache Server)로 쓰기보다는 분산캐시(Distributed Caced)를 별도의 레이어(Layer)로 처리하는 형식으로 바뀌었기 때문이다. 그럼에도 불구하고 아직까지도 Plugin 형태로 쓰고 있는 사용자와 서비스들이 있으며 Memcached보다는 Redis 사용자가 월등히 많아졌기 때문에 Redis Plugin 개발은 필요하다.



[그림 1] 구글트렌드 Memcached (blue) vs Redis (red)

## [ 목차 ]

- 1 분산캐시 (Distributed Cache)
- 2 Memcached 소개
- 3 Redis 소개
- 4 프로젝트 개요
- 5 프로젝트 현황
- 6 적용분야
- 7 Open Source DBMS 기술 동향
  - Oracle MySQL
  - MariaDB
  - Percona
  - PostgreSQL

## 1 분산 캐시 (Distributed Cache)

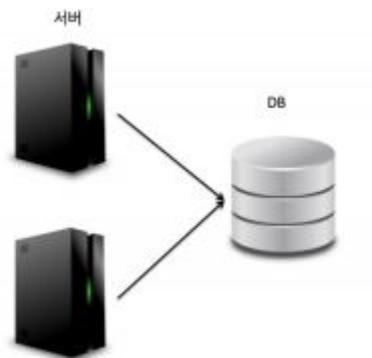
### 분산 캐시 시스템 <sup>1)</sup>

일반적으로 캐시라 하면 데이터나 값을 미리 복사해 놓는 임시 장소를 가리킨다. 계산 또는 저장된 값을 읽기 위해 특정 장소에 접근하는 시간이 오래 걸릴 경우, 해당 소요 시간을 줄이기 위해서 캐시가 만들어졌다. 즉, 캐시는 빠른 접근을 가능하게 하는 효율성에 집중되어 있다.

캐시는 CPU 성능을 높이기 위한 L1, L2, L3 캐시를 사용하는 CPU 캐시, 디스크의 내용을 RAM에 저장하는 DISK 캐시, web 브라우저의 캐시나 iOS, Android와 같은 미들웨어, 애플리케이션에서 사용하는 단말 애플리케이션 단위의 캐시, DB나 웹 서버, 대용량 서버에서 사용하는 분산 캐시 등으로 크게 나눌 수 있을 것이다.

분산 환경 또는 서버 환경에서 개발하면서 만들고 경험했던 캐시 시스템을 소개하고, 분산 캐시의 대표 주자인 Redis와 Memcached 등이 만들어지게 되는 배경을 살펴보고자 한다.

웹 서비스와 같은 일반적이고 기초적인 서버, DB의 아키텍처는 [그림 2]와 같다. L4와 같은 네트워크 장치와 방화벽과 요청에 대한 내용은 그림에서 제외했다. 서버에서 요청을 받고 처리하고 처리된 정보를 DB에 저장하거나 서버로 데이터를 전달한다.



[그림 2] 기초적인 서버, DB 연동 아키텍처

## 2 Memcached 소개

---

<sup>1</sup> <웹진 122 호 : 공학 트렌드> 대용량 서비스를 지탱하는 분산 캐시 시스템 Part 1 - <http://www.sw-eng.kr/member/customer/Webzine/BoardView.do?boardId=00000000000000022837&currPage=1&searchPrefacId=&titOrder=&writeOrder=&regDtOrder=&searchCondition=TOT&searchKeyword=redis>

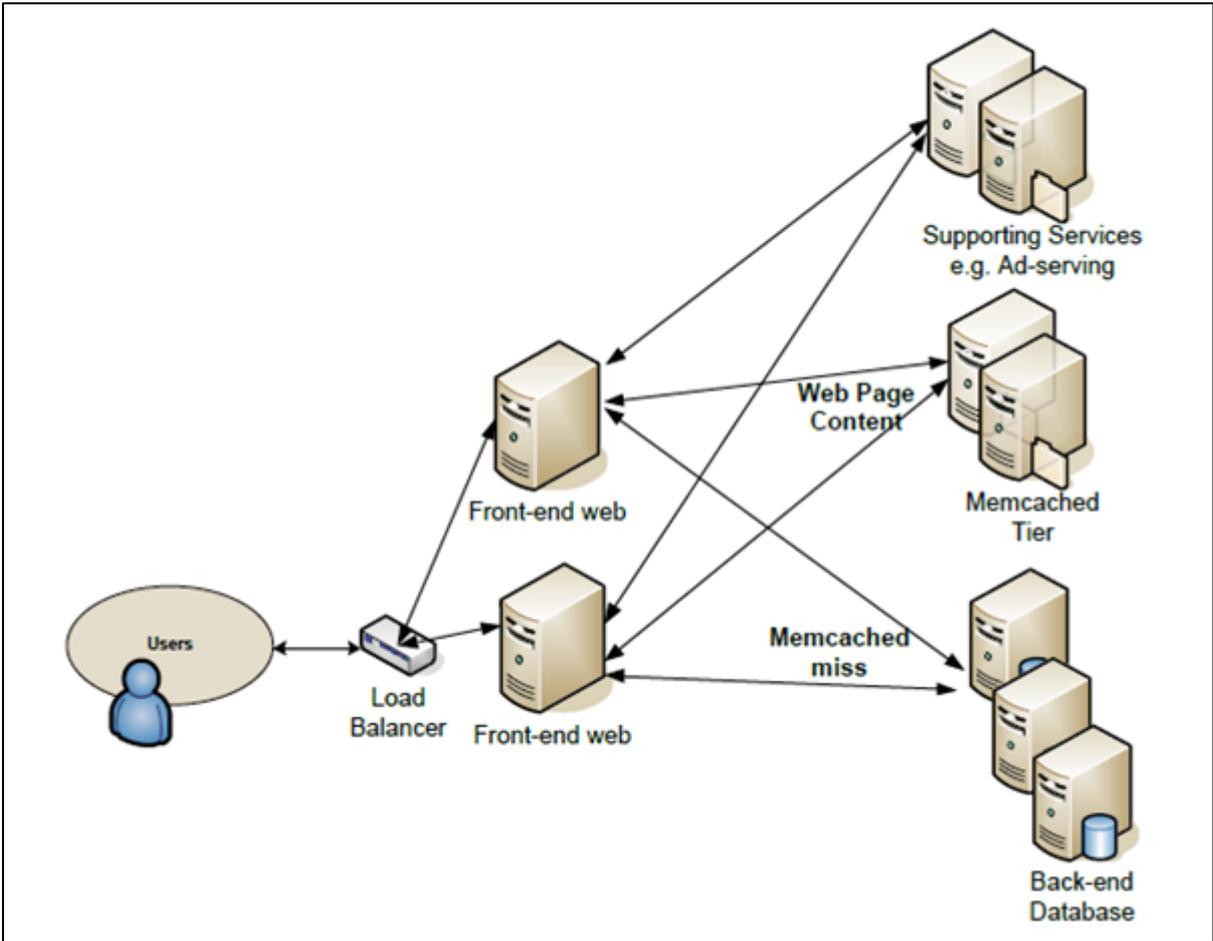
## Memcached 및 웹 서비스 소개 <sup>2)</sup>

Memcached는 Facebook, Twitter, Reddit, YouTube와 같은 클라우드 및 웹 서비스 제공 회사에서 사용하는 key-value 메모리 캐시로, 웹 데이터를 소비자에게 서비스하는데 있어 지연 시간을 줄이고 데이터베이스 및 컴퓨팅 서버에 대한 증설을 줄여준다. Latency를 줄이는 것 외에도 Memcached의 확장성 있는 아키텍처(scale-out)는 Memcached 서버를 간단하게 추가만 하여 처리량을 높일 수 있다. 그러나 코어 수가 4개를 넘으면 성능 저하가 발생하기 때문에 수직 scalability(scale-up)에는 문제가 있다.

Memcached는 Key-Value 쌍으로 이뤄진 간단한 데이터 타입을 저장하며, NoSQL 데이터베이스와 유사하지만 NoSQL처럼 영구적(persistent)이지는 않다. Memcached는 모든 Key-Value 쌍을 메모리에 저장하므로 서버장애나 오류가 발생했을 때 저장된 데이터가 모두 손실된다. 이 글에서는 키(key)와 해당 키의 값(value)을 지칭할 때 '캐시 항목(cache item)'이라는 용어를 사용한다. 이때 키는 고유한 값이다. 웹 서비스 아키텍처에서 Memcached 애플리케이션은 그림 2에서 보는 바와 같이 프론트엔드(Front-end) 웹 서버(또는 계층 (tier))와 백엔드(Back-end) 데이터베이스 사이에 위치한다.

---

<sup>2</sup> 네이버 D2 - <http://d2.naver.com/helloworld/151047>



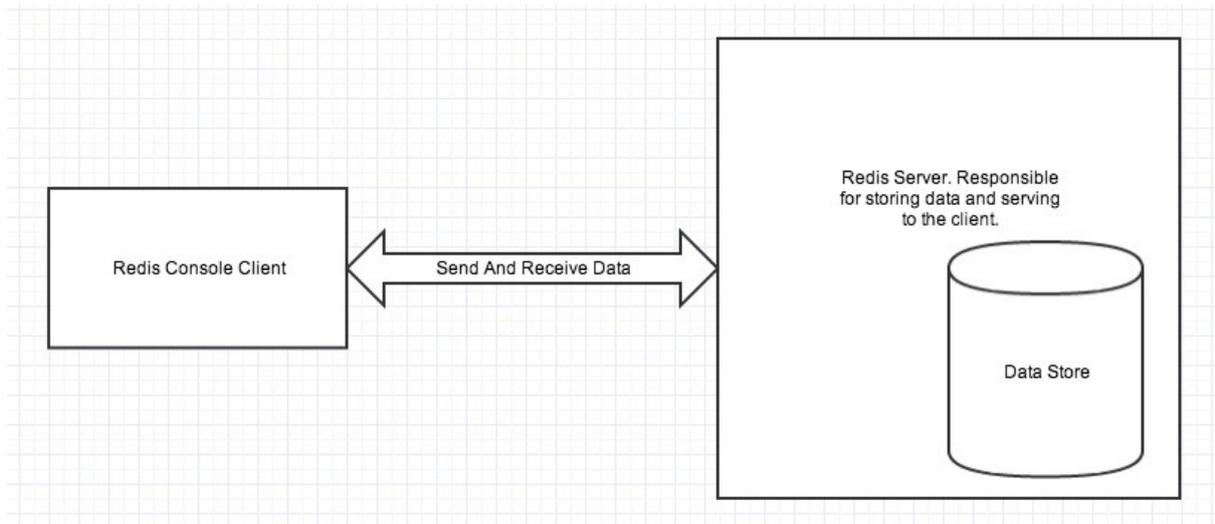
[그림 3] 프론트엔드 웹 계층 및 백엔드 데이터베이스 사이에 위치하는 Memcached 서버

### 3 Redis 소개

#### Redis 캐시<sup>3)</sup>

Redis는 REmote DIctionary Server의 약자이며, 웹 로그 분석 시스템의 성능을 높이기 위해서 개발되었다. BSD 라이선스(Three Clause BSD License) 오픈소스 기반이다. DB-Engine 선호도 랭킹에 따르면 선호도 1위를 기록하고 있다. Redis 홈페이지는 심플한 화면 구성을 가지고 있다. 문서와 다운로드 링크, 명령어, 언어별 클라이언트, 이슈 트래킹 화면, 지원 정보, 라이선스 등을 포함하고 있다.

<sup>3</sup> <웹진 123 호 : 공학 트렌드> 대용량 서비스를 지탱하는 분산 캐시 시스템 Part 2- <http://www.sw-eng.kr/member/customer/Webzine/BoardView.do?boardId=00000000000000022844>



[그림 4] Redis Single Instance Architecture <sup>4)</sup>

## 4 프로젝트 개요

### *MySQL Redis Plugin*

MySQL 데이터베이스 서버는 세계에서 제일 널리 사용되는 오픈 소스 데이터베이스다. 매우 효율적인 아키텍처를 채택하여 빠른 성능과 쉬운 사용자 구성을 가능하게 해준다. 소프트웨어 내적으로 다량의 코드 재사용과 풍부한 기능 생산에 대한 최소화 기법은 최고의 속도와 압축성(compactness), 안정성과 용이한 deployment를 보장해주는 데이터베이스 관리 시스템을 탄생시켰다. 테이블 핸들러(table handler)에서 코어 서버(core server)를 분리하는 독창적인 방식은 정교한 트랜잭션 제어(transaction control) 또는 트랜잭션이 필요 없는 초고속의 디스크 액세스(disk access)를 구현하도록 해주었고, 이와 같은 기능은 모든 상황에 제일 적합한 방식이다. <sup>5)</sup>

Redis는 오픈 소스이며 네트워크화(network), 인메모리 데이터베이스(in-memory database), 선택적 영구성(durability) key-value 데이터 저장소(key-values data store)이다. DB-Engines.com의 월간 랭킹에 따르면, 레디스는 가장 인기 있는 key-value 저장소이다.

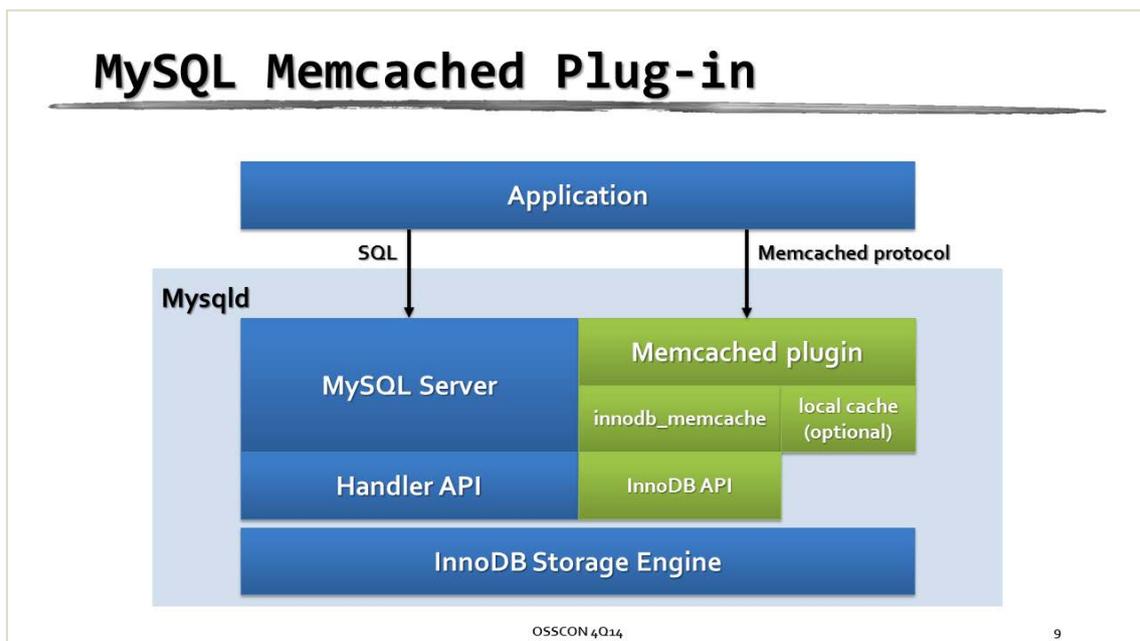
최근에는 데이터 증가로 인한 부하 때문에 MySQL 샤딩(sharding)만으로는 transaction(트랜잭션)을 감당하기 어려운 경우가 많다. 그래서 MySQL을 단독으로 쓰기보다는 어플리케이션

<sup>4)</sup> <http://qimate.com/overview-of-redis-architecture/>

<sup>5)</sup> MySQL 소개 - <http://www.linux.co.kr/database/mysql/product/MySQL/index.htm>

(application) 또는 프록시(proxy) → MySQL 사이에 캐시서버(cache server) 또는 인메모리 key-value 저장 (in-memory key-values store) 방식을 통해서 DBMS 부하를 줄이고, 응답시간(response time)을 높이는 방식을 많이 채택하고 있다. 그 중에서도 Memcached와 Redis를 Google, Twitter, Facebook, LINE, Daum Kakao 등과 같은 글로벌 서비스 업체들이 사용하고 있다.

Memcached는 예전부터 MySQL 유저들이 많이 사용했다. MySQL 5.5 버전까지는 사용자정의함수(UDF) 혹은 데몬 플러그인(daemon plug-in)이 제공되다가 MySQL 5.6 버전부터는 InnoDB 엔진에 통합된 Plug-in을 제공하고 있다.



[그림 5] MySQL Memcached Plug-in Architecture <sup>6)</sup>

최근 cluster architecture 보다는 shading architecture가 더 각광받고 있는 추세에 부합하여 Redis는 Memcached 보다 더 많이 쓰이고 있다. 그런데 그에 반해서 MySQL Plug-in이 존재하지 않는다. 따라서 UDF 형태로 사용하거나, 개별적으로 구현하여 사용하고 있다. 이는 MySQL + Memcached 조합의 초창기 구성방식이다.

최근에 각광받고 있는 Redis의 Plug-in을 개발함으로써 이런 단점을 보완하고, 캐시 아키텍

<sup>6</sup> 4Q14 OSSCON - <http://www.slideshare.net/LeeHyeongchae/20141105-osscon-mysqlredisplugin-41155534>

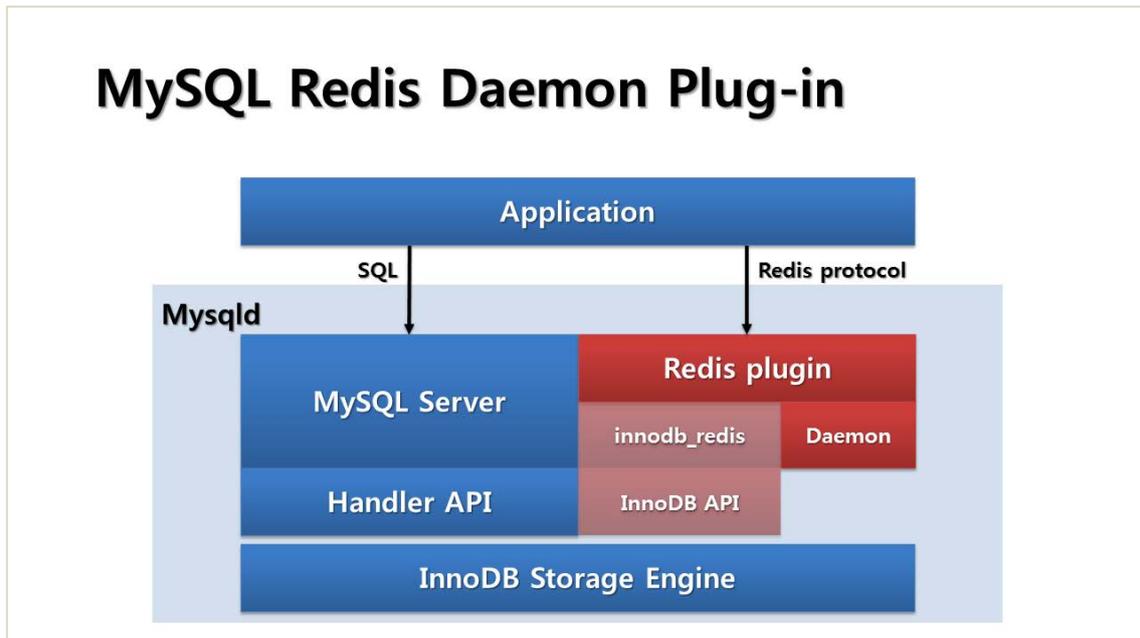
처의 유연함, 개발 및 관리의 편의성을 높이고자 한다.

향후에는 MySQL, MariaDB에 소스를 기여(contribution)하여 GA(General Availability) 버전에 포함되길 바란다.

## 5 프로젝트 현황

프로젝트 초기에 예상했던 것보다 MySQL Memcached Plug-in 소스분석과 InnoDB 통합 형태의 소스분석에 시간을 많이 소비하였다. 이는 Memcached Plug-in 소스와 Membase's Memcached <sup>7)</sup> 소스의 차이, 불확실한 Plug-in 소스 버전정보에서 기인한다. 또한 소스에도 많은 부분에서 수정이 가해졌다.

현재는 Daemon Plug-in 개발 완료, InnoDB Plug-in 베타 릴리즈 완료 후 QA와 디버깅을 진행 중이다.



[그림 6] MySQL Redis Plugin Architecture <sup>8)</sup>

## 6 적용분야

Redis와 MySQL/MariaDB를 사용하는 분야에는 대부분 적용가능하고 구글, 트위터, 링크드인, 페이스북, 라인, 카카오톡 등과 같이 오픈 소스 데이터베이스를 널리 사용하고 있는 기업에서

<sup>7)</sup> Couchbase - [http://en.wikipedia.org/wiki/Couchbase\\_Server](http://en.wikipedia.org/wiki/Couchbase_Server)

<sup>8)</sup> 4Q14 OSSCON - <http://www.slideshare.net/LeeHyeongchae/20141105-osscon-mysqlredisplugin-41155534>

많은 활용이 예상된다. 또한, Redis는 DB-Engines Raking of Key-value Stores <sup>9)</sup> 에서 순위가 1위이며 2위인 Memcached와 대비해도 3배 이상 높은 점수인 만큼 인기 있는 Key-value Stores이다.

Rank	Dec 2014	DBMS	Database Model	Score	Dec 2014
1	1	Oracle	Relational DBMS	1497.55	+37.76
<b>2</b>	<b>2</b>	<b>MySQL</b>	<b>Relational DBMS</b>	<b>1298.54</b>	<b>+29.96</b>
3	3	Microsoft SQL Server	Relational DBMS	1123.16	-76.891
4	5	MongoDB	Document Store	301.09	+54.87
<b>5</b>	<b>4</b>	<b>PostgreSQL</b>	<b>Document Store</b>	<b>280.09</b>	<b>+26.09</b>
<b>23</b>	<b>27</b>	<b>MariaDB</b>	<b>Relational DBMS</b>	<b>27.74</b>	<b>+10.41</b>
<b>96</b>	<b>82</b>	<b>Percona</b>	<b>Relational DBMS</b>	<b>1.58</b>	<b>+0.38</b>

[표 1] DB-Engines Ranking <sup>10)</sup>

## 7 Open Source DBMS 기술 동향

최근 몇 년 사이에 글로벌 웹 서비스들이 성공을 거두면서 많은 오픈 소스 소프트웨어가 개발되었고, 각광받고 있다. 대부분의 서비스들이 관계형 데이터베이스시스템 (Relational DBMS)를 기반으로 데이터를 저장 및 처리하고 있다.

2015년 가트너 매직 쿼드런트(Gartner magic Quadrant)에는 Percona(+TokuDB)가 추가되었고, Oracle(+MySQL), MariaDB, EnterpriseDB(+PostgreSQL)를 포함해서 Big4 Open Source Relational DBMS가 리스트 업 되었다. 또한 Big4를 지원하는 AWS(Amazon Web Service)가 신규 LEADERS로 추가된 점도 주목할만하다.

<sup>9)</sup> DB-Engines Ranking of Key-value Stores - <http://db-engines.com/en/ranking/key-value+store>

<sup>10)</sup> DB-Engines Ranking (2015.12.07) - <http://db-engines.com/en/ranking>



[그림 7] 2015 Gartner's Magic Quadrant for Operational Database Management Systems <sup>11)</sup>

2015년 DB-Engines Ranking은 전년도에 비해서 크게 변화한 점은 찾아보기 어렵고, MongoDB가 PostgreSQL을 앞질렀을 정도로 NoSQL 사용이 일반화되었다는 점을 주목할만하다.

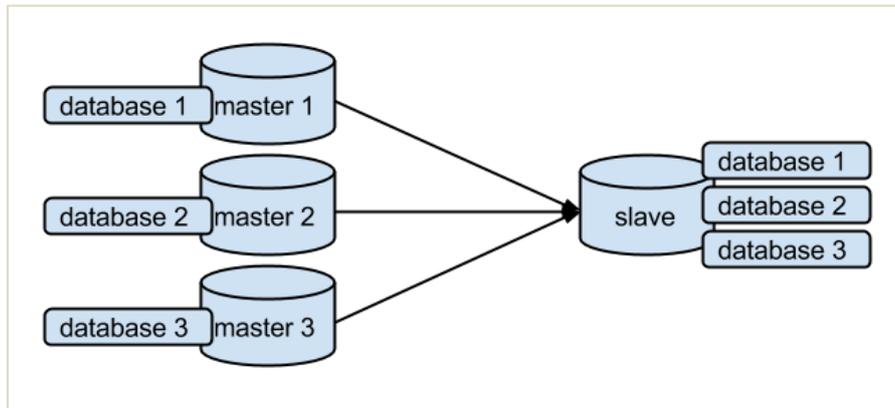
### - Oracle MySQL

현재 최신 버전은 MySQL 5.7이고, MySQL 5.7 GA 버전은 2015년 12월 5일에 릴리즈된 상태이다. MySQL 5.7 버전은 5.6 버전과 대비해서 성능(performance)과 확장성(scalability)이 개선되었다. 그러나 Read 성능이 좋아졌을 뿐이고, Write 성능 개선은 크지 않다.

개선 내용을 살펴보면, 이중화 개선(Replication Enhancements), InnoDB 엔진 개선(InnoDB Enhancements), 트리거(Triggers), 성능 개선(Performance Improvements), 최적화 개선(Optimizer Improvements), MySQL 테스트 개선(MySQL Test Suite Enhancements), 보안 개선(Security Enhancements)등을 크게 볼 수 있다. 그 중에서 주목할 만한 내용은 Multi-Source

<sup>11</sup> <https://www.gartner.com/doc/3147919/magic-quadrant-operational-database-management>

Replication 지원, Transportable Tablespace 지원, Fusion-I/O atomic-write 지원, Online DDL 개선 등이 있다.



[그림 8] Multi-Source Replication

#### - MariaDB

MariaDB는 오픈 소스의 관계형 데이터베이스 관리 시스템(RDBMS)이다. MySQL과 동일한 소스 코드를 기반으로 하며, GPL v2 라이선스를 따른다. 오라클 소유의 현재 불확실한 MySQL 라이선스 상태에 반발하여 만들어졌으며, 배포자는 몬티 프로그램 AB(Monty Program AB)와 저작권을 공유해야 한다. 이것은 MySQL과 높은 호환성을 유지함과 동시에, MySQL APIs 명령에 정확히 매칭하여 라이브러리 바이너리와 상응하게 하여 교체 가능성을 높이고자 함이다. 마리아 DB는 새로운 저장 엔진인 아리아(Aria)뿐만 아니라, InnoDB를 교체할 수 있는 XtraDB 저장 엔진을 포함하고 있다. 이것은 트랜잭션과 비트랜잭션 엔진 그리고 미래에 나올 MySQL 판에 대응하고자 함일 것이다. <sup>12)</sup>

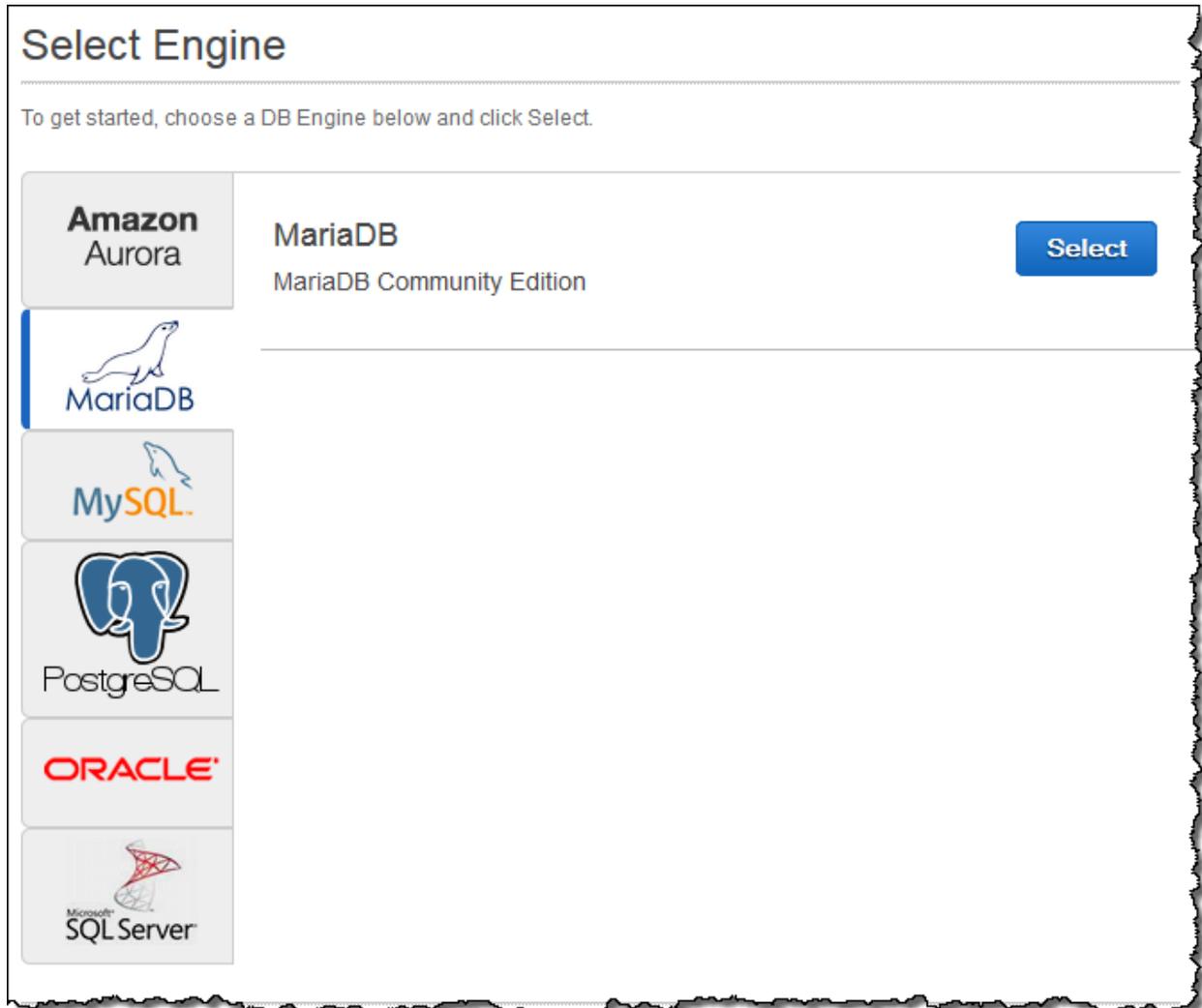
MariaDB는 5.5 버전과 10.x 버전으로 나눈다. MariaDB 5.5 버전은 MySQL과의 호환성을 유지한 버전이고, MariaDB 10.x는 MySQL과는 차별성을 두기 위한 버전이다. 결국 MariaDB 5.5 버전은 MySQL 5.5와 대비해서 큰 변화는 없다. 단지, 더 많은 스토리지 엔진(storage engines)을 제공하며, 일부분 수정이 반영되어서 성능개선이 이루어진 점이 다를 뿐이다. 최근에는 MariaDB 5.5보다는 MariaDB 10.x를 주력으로 밀고 있다.

MariaDB 10.x도 MariaDB 5.5의 branch이기 때문에 MySQL 5.5나 MySQL 5.6과 대비해서 큰 변화는 이루어지지 않고 있다. 추가적으로 병렬 이중화(parallel replication), 멀티소스 이중화(multi-source replication), 롤(roles), PCRE(Perl Compatible Regular Expressions), KILL QUERY ID, DELETE, RETURNING 등을 지원한다.

<sup>12)</sup> MariaDB - <http://en.wikipedia.org/wiki/MariaDB>

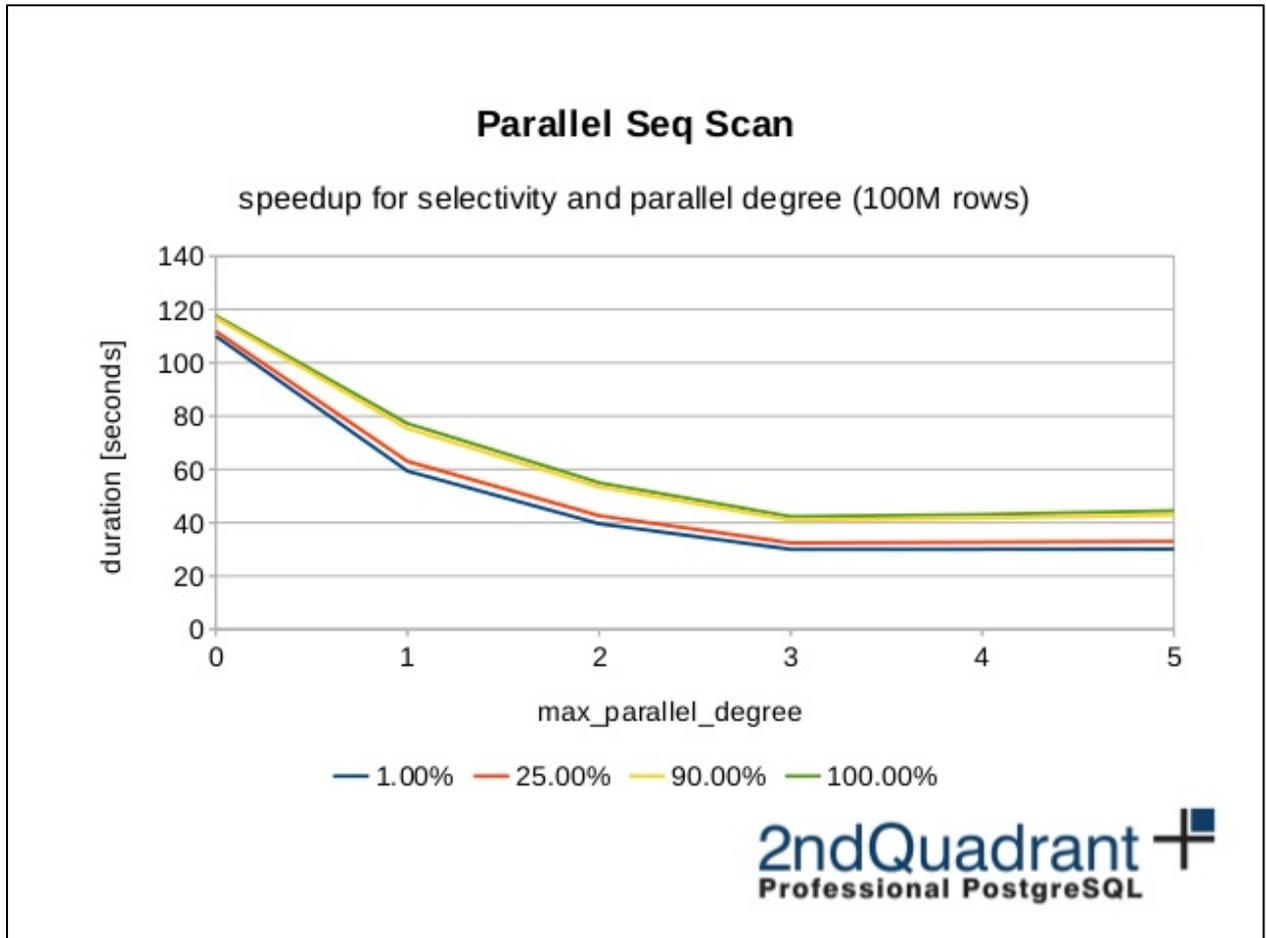
MariaDB 는 MySQL Cluster 대신에 MariaDB Galera Cluster <sup>13)</sup> 를 지원한다.

2015 AWS Re:Invent에서 MariaDB Service를 발표하였다. 이로써 AWS RDS for MySQL과도 경쟁할 수 있게 되었다.



[그림 9] AWS RDS for MariaDB 10.0.17

<sup>13)</sup> MariaDB Galera Cluster - <https://downloads.mariadb.org/mariadb-galera>



[그림 10] PostgreSQL 9.5 Parallel Sequential Scan

#### - Percona

Percona는 Peter Zaitsev과 Vadim Tkachenko이 2006년에 설립하였으며 MySQL 전문 컨설팅, 지원, 관리 서비스 업체이다. Percona Server는 XtraDB, InnoDB 엔진을 주력으로 하고 있고 올해 TokuTek을 인수하여 TokuDB에 대한 주도권도 가져가게 되었다. Percona는 MySQL, MariaDB에 비해서 성능부분에 반영을 빠르게 하고 있다.

Percona Toolkit은 MySQL, MariaDB와도 호환되기 때문에 대부분의 기업들과 서비스들이 사용하고 있을 정도로 유명하다.

#### - PostgreSQL

PostgreSQL은 객체-관계형 데이터베이스 관리 시스템(ORDBMS)의 일종이다. BSD 허가권으로 배포되며 오픈 소스 개발자 및 관련 회사들이 개발에 참여하고 있다. <sup>14)</sup>

<sup>14)</sup> PostgreSQL - <http://en.wikipedia.org/wiki/PostgreSQL>

PostgreSQL은 캘리포니아대학교 버클리 분교에서 시작된 Ingres 프로젝트로부터 시작되었다. 프로젝트 리더인 Michael Stonebraker는 1982년 Ingres의 상용화를 위해 학교를 떠났었다. 그 후, 1985년에 다시 학교로 돌아온 그는 1980년대 초반부터 급증하게 된 당시의 데이터베이스 시스템의 문제점들을 해결하고자 "post-Ingres(후기-Ingres)" 프로젝트를 시작했다. 상용화 된 Ingres와는 다르게 post-Ingres에 대한 아이디어를 공유하기 위해 Ingres의 코드 일부를 제외하고 분리시켰다. 1986년부터 개발팀은 데이터베이스 시스템의 기본적인 사항에 대해 몇 가지 논문을 제출하고 이어서 1988년까지 실제로 운영이 가능한 프로토타입을 완성하게 된다.

최신 GA 버전은 PostgreSQL 9.6 이다. PostgreSQL 은 8.x버전대에서 9.x 버전으로 업그레이드 되면서 성능과 기능이 대폭 개선되었다. <sup>15)</sup>

PostgreSQL은 MySQL보다 더 커뮤니티(Community) 중심적인 개발 프로세스를 가지고 있으며 성능 및 기능 개선에 필요한 TODO 리스트를 중심으로 개발하고 있다. <sup>16)</sup>

PostgreSQL 9.6에서는 "Parallel execution of sequential scans, joins and aggregates"와 "postgres\_fdw now supports remote joins, sorts, UPDATEs, and DELETEs" 등을 지원한다. PostgreSQL 10 로드맵에는 성능개선에 대한 부분이 많기 때문에 기대해 볼만하다.

---

<sup>15</sup> What's new in this release -

[https://wiki.postgresql.org/wiki/Category:What%27s\\_new\\_in\\_this\\_release](https://wiki.postgresql.org/wiki/Category:What%27s_new_in_this_release)

<sup>16</sup> PostgreSQL TODO - <https://wiki.postgresql.org/wiki/TODO>