



# Redis, MongoDB 그리고 MySQL

모바일 애플리케이션 서비스에서의 로그 수집과 분석



NoSQL

NoSQL?

NoRDB?

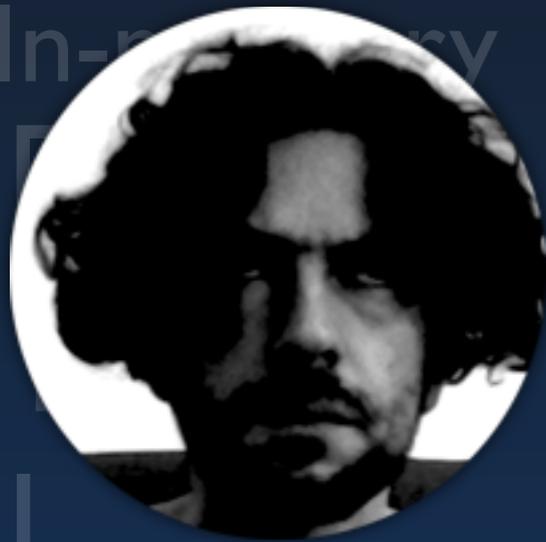
NoSQL = Not only SQL  
≠ Do not use SQL

via <http://nosql-database.org/>

In-memory Document Store  
Redis, MongoDB  
NoSQL

그리고

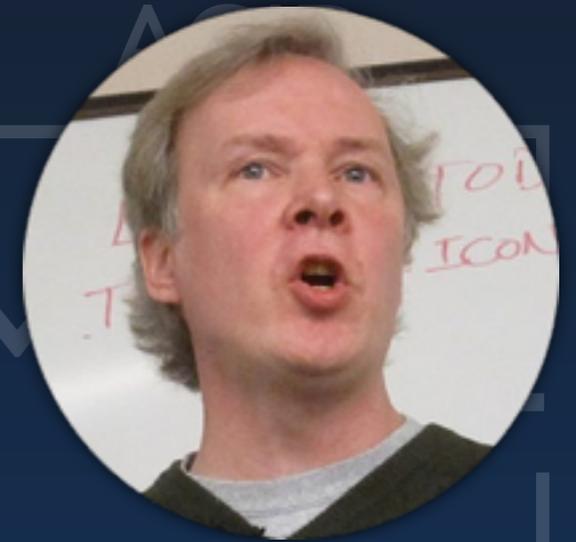
ACID  
MySQL  
RDBMS



Salvatore



Dwight



Monty



VS

MAY/JUNE 2010

ORACLE<sup>®</sup>

ORACLE.COM/ORAENR/MAGAZINE

# ORACLE MAGAZINE

John Fowler Speaks Oracle's Executive Vice President, Systems, on the Sun acquisition / 23  
BMW ORACLE Racing How Oracle Data Mining helped the team win the America's Cup / 45  
Java Development Kit 7 A new bytecode and linkage mechanism offer better support for dynamically typed languages / 63  
Frameworks Modularize your applications by using Oracle ADF libraries / 69  
Ask Tom Our technologist concludes his exploration of Edition-Based Redefinition / 85

## IRON MAN RETURNS

Tony Stark once saved his heart—and his life—with the Iron Man suit. Now he's found his heart and forged new life into his company, Stark Industries, by abandoning munitions manufacture for cutting-edge communications. Learn how Oracle helped with the transformation. >

### Sharing Resources in the Cloud

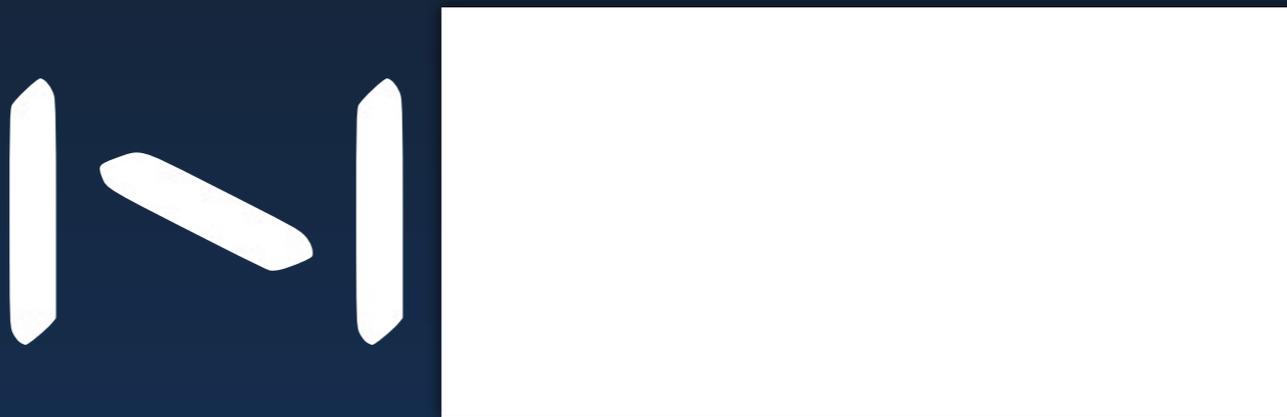
To maximize resources and ensure data security, more companies are turning to cloud computing. Oracle technologies lead the way. 

### Software. Hardware. Complete.

 Oracle's acquisition of Sun completes and improves the technology stack, optimizes integration, and simplifies operations. 



10년 전으로 돌아갑니다.



“로그를 보려면 한 달 정도 걸립니다.

그래도 어제 동접은 나오잖아요.”

“동접이 중요한게 아닙니다.

왜, 언제 이탈하는지 알아야죠.”

1일

최소 5일 (반복됨)

지표요청

로그가 있는가?

스키마 설계!

DBA가 반려

$1 + 5 = 6$ 일

1일

2일

3일

테이블 생성

프로토콜 추가/테스트

점검일 대기

서버 재시작

$$1 + 2 + 3 = 6\text{일}$$

최소 5일

1일

서버팀에 로그 요청

CSV로 받음

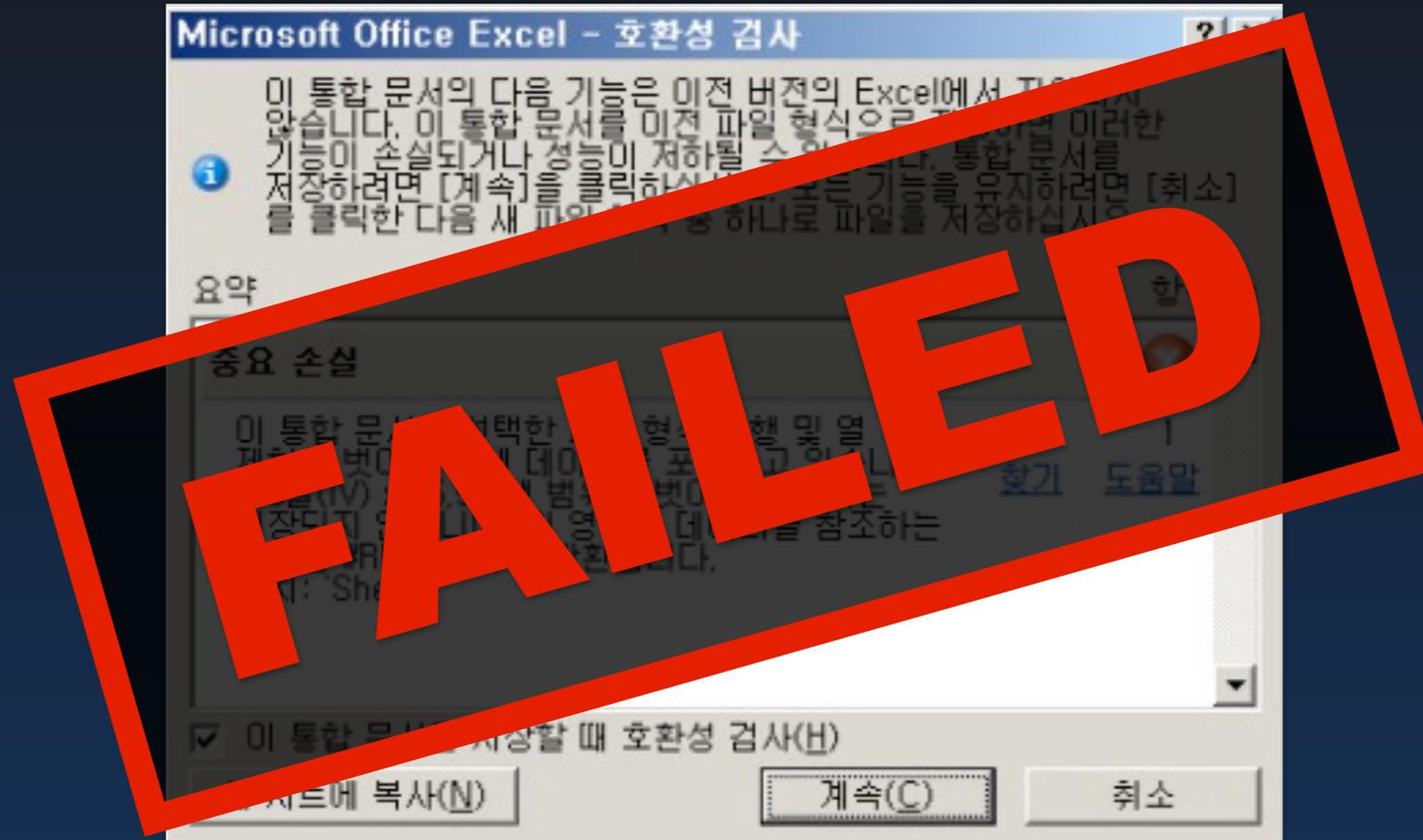
엑셀에 넣음

끝!

$$5 + 1 = 6\text{일}$$

좀 과장되게 얘기해서

$$6 + 6 + 6 = 18\text{일}$$



엑셀 65,536행 초과



뭐가 문제인지 모르겠어

2년 뒤

목마른 사람이 우물을 팝니다

# WBERS

- LAMP 기반 웹 서비스
- 스키마 저작 도구 ≍ phpMyAdmin
- HTTP / HTTPS로 로그 쌓기
- 클라이언트 덤프 수집 / 분석
- 자주 사용하는 스키마 / 쿼리 저장
- 일 / 월 / 버전 단위 테이블 파티셔닝



베르스ID \_\_\_\_\_

패스워드 \_\_\_\_\_

로그인

LANGUAGE KOREAN ▼

# 덤프 수집기 & 분석기

address	reason	id	callstack	register	version	modinfo	dumplink	pcinfolink	time	wberscount
7C812A6B	Unknown	...	7C812A6B 004599C2 0040803C 00437489 00433036 00435812 0043589F 77CF8709 77CF87EB 77CFB743 ...	...	120506_000c9430	with 35 dll(s)	get dump	----	2012-05-06 18:50:43	2
00477204	EXCEPTION_INT_DIVIDE_BY_ZERO	...	00477204 004294F0	...	110711_000d02c7	with 57 dll(s)	get dump	----	2012-05-06 14:15:08	1

## \* 쿼리 만들기

select

group

order

where

select

group & order

where

페이지당 결과 출력수

기본값으로 설정

# 쿼리 작성

1일

1일

지표요청

로그가 있는가?

스키마 설계

클라이언트 수정

1 + 1 = 2일

1일

로그 확인

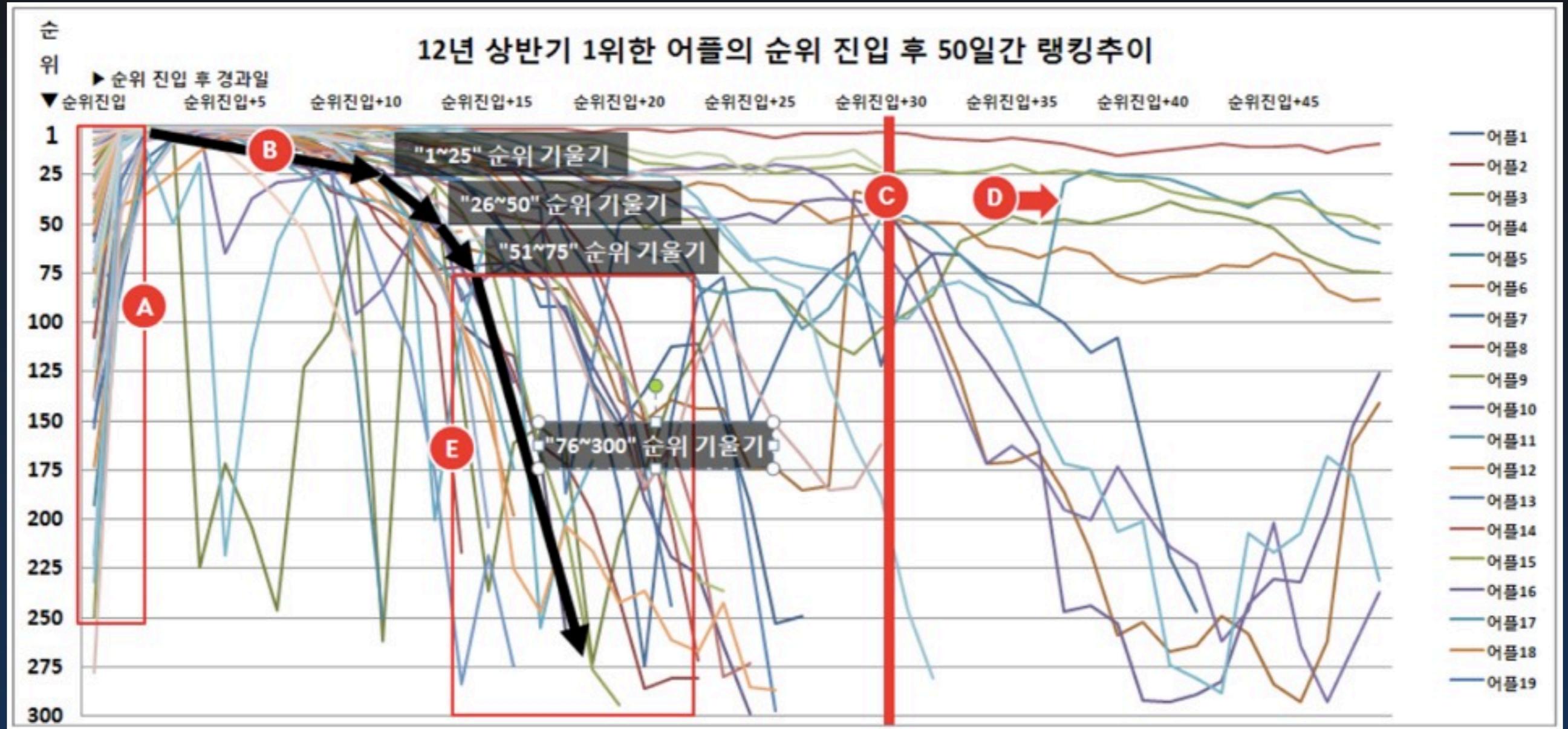
쿼리 작성

지표 도출

끝!

1일

거짓말 조금 보태서  
 $2 + 1 = 3$ 일



# 30일 내에 승부

# LOG

- 필요한 사람이 설계해야 하고,
- 필요한 시점에 수집해야 하고,
- 필요한 시기에 활용해야 한다.



그래도 동접이 떨어져요



망했어요 후후

괜찮아!

```
mysqldump -uroot -p -e  
--opt -c log | gzip -c  
log-20121114.sql.gz
```

로그를 잘 쌓으면 망해도 준치

다음 번 서비스 지표로 활용 가능

# 단점

- MySQL은 지속적으로 확장하기 어렵다.
- 어쨌든 스키마를 매번 만들어야 한다.
- 쌓는 중에 형식 변경 비용이 비싸다.



# 모바일 환경

- 항상 로그 전송이 가능한 상태는 아님.
- JSON 형태로 로컬 저장소에 보관하다,
- n개까지 모으거나, 필요한 경우 즉시 발송.
- 서버에서는 받은 후, 낱개로 풀어서 저장.

# 장애 대응

- HTTP / HTTPS 로 전송
- 200 OK 응답을 받지 못한 경우 장치에 보관
- IDC 이전시 효과적으로 활용

## Speaker's note

- 기존에 하던 방식과 비슷하지만, 서버의 응답을 확인해서 200 OK 를 받지 않으면 로컬 큐에서 해당 로그를 제거하지 않고, 다음에 다시 보낼 수 있도록 합니다.
- 이를 이용해, 올해 IDC를 이전하면서 약 5-6시간 정도의 장애가 있었는데 해당 기간의 로그를 잃어버리지 않고, 대부분 회수할 수 있었습니다.
- 물론, 네트워크가 활성화되지 않은 상태에서 한 번 켜고, 다시 켜지 않으면 해당 로그는 잃어버리지만, 감당할 수 있는 범위의 loss라고 판단했습니다.

# 로그 스토리지 검토

- 자원 = 개발자 + 시간 + 돈
- 자원이 부족했기에 제한된 선택만 가능
- 대용량 처리 능력보다 사용 편의성에 중점
- Hadoop, Cassandra 등은 **개발 역량 부족**



mongoDB

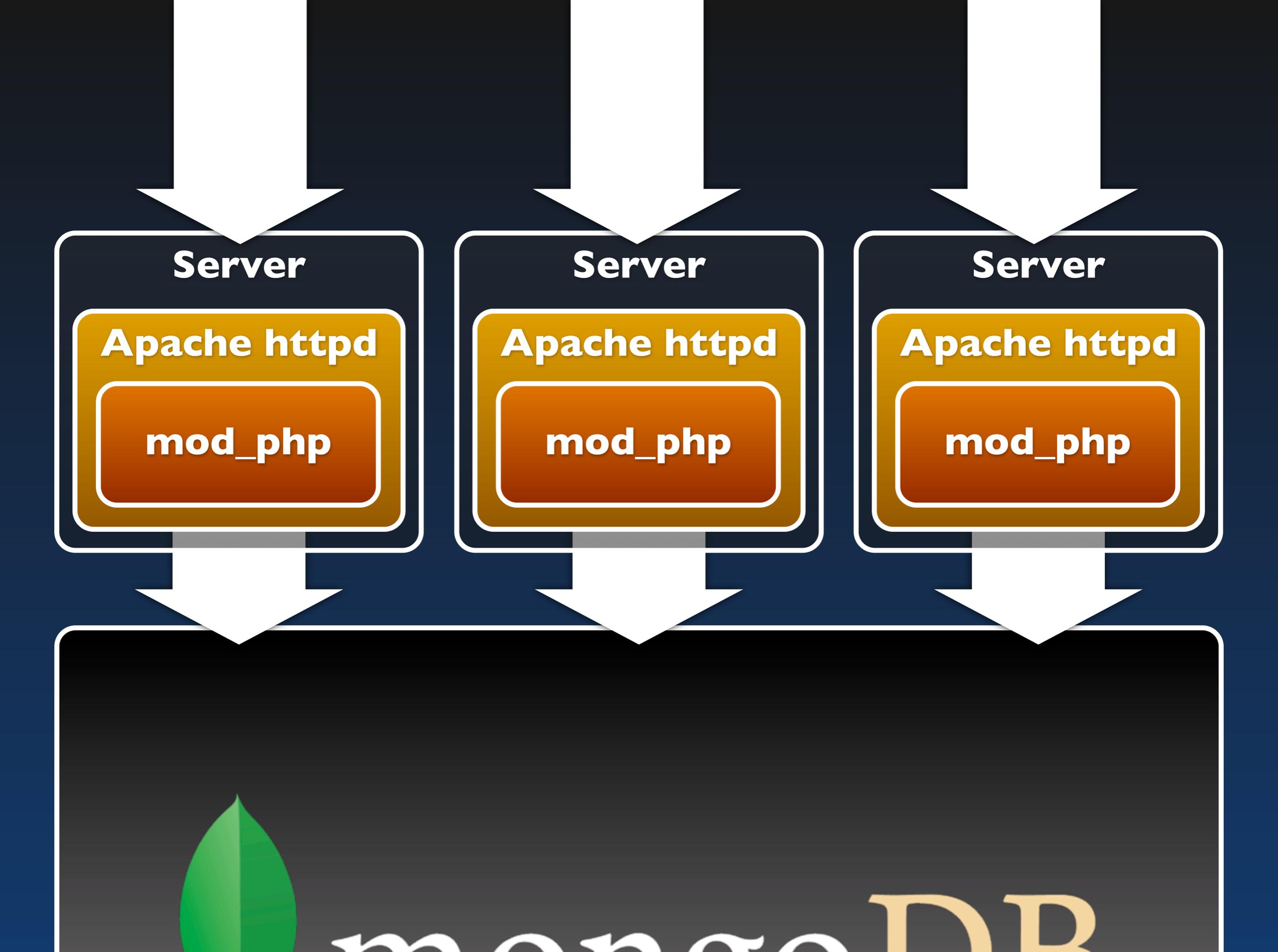


# 해결

- 기존 : MySQL은 지속적으로 확장하기 어렵다.
  - ReplicaSet, Sharding을 지원한다.
- 기존 : 어쨌든 스키마를 매번 만들어야 한다.
  - JSON 형식으로 자유롭게 저장 가능.
- 기존 : 쌓는 중에 형식 변경 비용이 비싸다.
  - 원한다면 언제든지 내용을 바꿀 수 있다.







**Server**

**Apache httpd**

**mod\_php**

**Server**

**Apache httpd**

**mod\_php**

**Server**

**Apache httpd**

**mod\_php**



**mongoDB**

Server

Apache httpd

mod\_php

Server

Apache httpd

mod\_php

Server

Apache httpd

mod\_php

**FAILED**

900+ connections

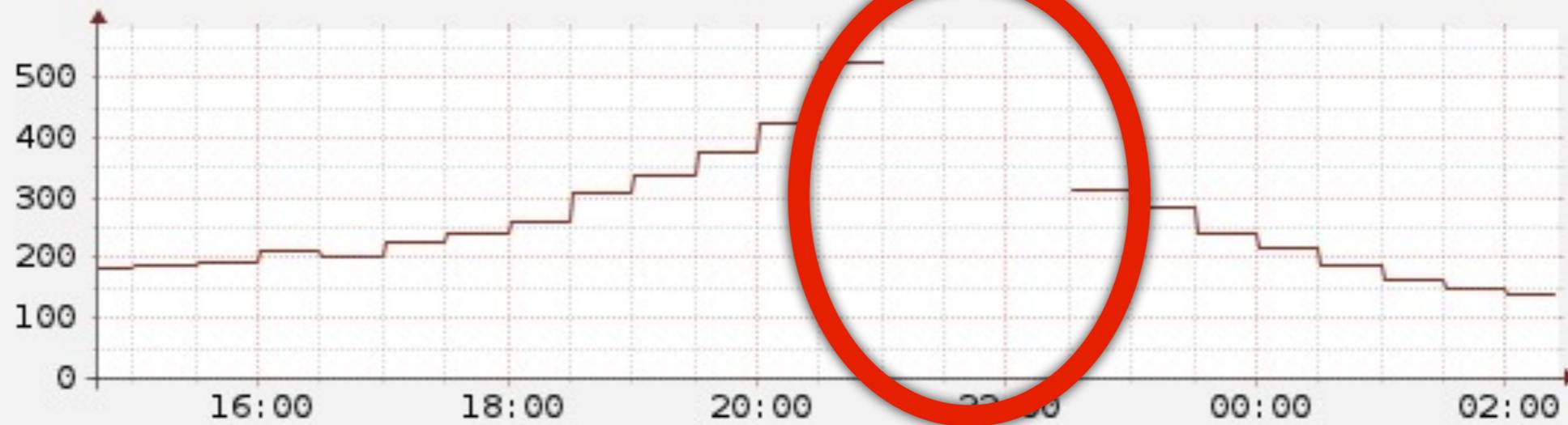


mongoDB

# Speaker's note

- 접속자가 늘어나 900 커넥션이 넘어가자 더 이상 동작 할 수 없는 상황에 이르게 됩니다.
- 락 작업과 쓰기에 바빠 더 이상의 접속을 허용하는 것이 무의미해집니다.
- MongoDB는 시스템 ulimit의 80%가 기본 커넥션 제한 값이고, 최대 2만개의 연결을 허용하지만, 천개쯤만 되어 도 장애 상태가 되었습니다.

### idc / orange1 - MongoDB Connections



From 2012/04/30 14:41:56 To 2012/05/01 02:24:02

Connected Clients Cur: 138.5 Avg: 256.0 Max: 525.6

Weekly (30 Minute Average)

RRDTOOL / TOBI OETIKER

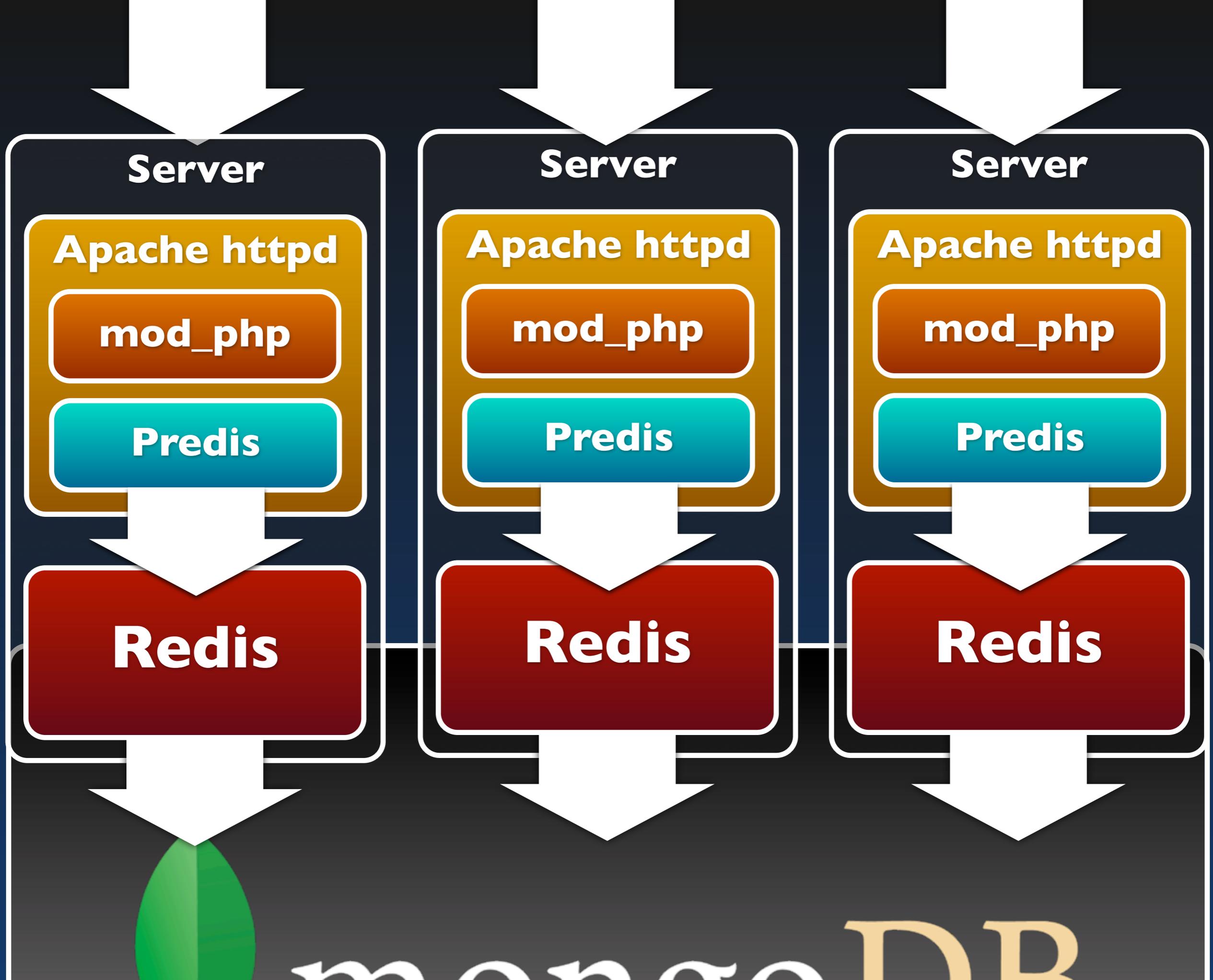
900+ connections

mongodb

- 
- The diagram illustrates a server architecture where three servers are running Apache httpd with the mpm\_prefork module and the mod\_php module. This configuration leads to a high number of connections, specifically 900+, which causes a global write lock issue in MongoDB.
- Apache httpd mpm\_prefork 사용
  - 프로세스마다 커넥션을 요구
  - 연결 폭주, 서비스 장애 발생
  - MongoDB Global write lock 문제

900+ connections

위 상황은 MongoDB 1.6.3 을 가정함



**Server**

**Apache httpd**

**mod\_php**

**Predis**

**Redis**

**Server**

**Apache httpd**

**mod\_php**

**Predis**

**Redis**

**Server**

**Apache httpd**

**mod\_php**

**Predis**

**Redis**



**mongo DB**

# Speaker's note

- 매 사용자 연결마다 커넥션을 맺는 것이 불합리하므로, 한 곳에 모아 일괄적으로 넣으면 어떨까 생각이 들어 redis를 도입했습니다.
- MySQL 같은 경우에도 여러 트랜잭션으로 insert 하는 것에 비해 한 트랜잭션에 몰아 넣는 것이 효율적이므로, 같은 방식으로 적용해봤습니다.
- 이에 php의 Predis 모듈을 활용하게 되었습니다.

Server

Server

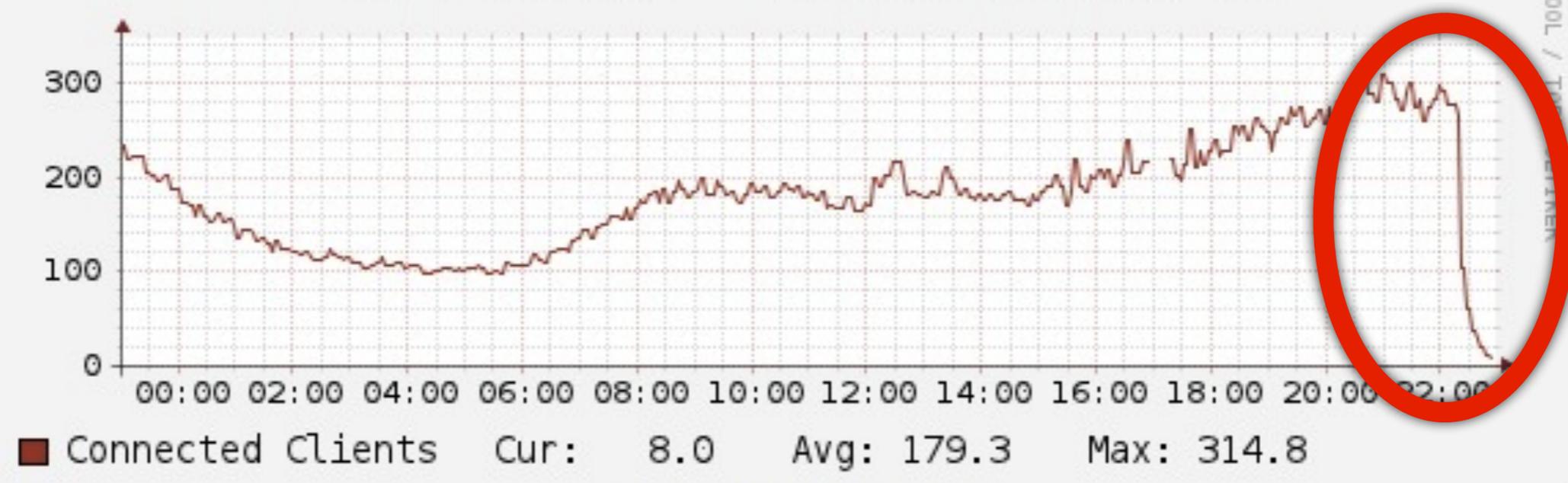
Server

Apache httpd

Apache httpd

Apache httpd

### idc / orange1 - MongoDB Connections



Daily (5 Minute Average)

# Redis

- In-memory Key-Value Storage by ANSI C
- Replication, Cluster 지원
- 다양한 데이터 타입 지원
  - strings, hashes, lists, sets, sorted sets



# Link

- @charsyam blog (very useful)
  - <http://charsyam.wordpress.com/category/cloud/redis/>
- Redis를 이용한 MongoDB 기반 로그 수집기 성능 개선
  - [http://blog.naver.com/ez\\_/140158788246](http://blog.naver.com/ez_/140158788246)
- Predis, using redis on PHP
  - [http://blog.naver.com/ez\\_/140158670703](http://blog.naver.com/ez_/140158670703)

- Redis를 일종의 버퍼로 사용
- 모아서 bulkInsert 가능
- 로그 수 파악을 위한 캐시 UPDATE 명령 절약
- 피크시 200 log/sec 에서 ~3MB의 메모리 사용

# Speaker's note

- percona에서 제공하는 모니터링 플러그인을 사용해서 모니터링합니다.
- <http://www.percona.com/software/percona-monitoring-plugins>
- cacti로 관찰한 것 뿐 아니라, mongostat 이나 mongod 의 웹 페이지를 통해 mongo의 상태를 보는 것도 좋습니다.
- 서비스/이벤트, 즉 로그의 종류에 따라 몇 개나 쌓았는지를 기록하는데, 이를 로그를 쌓으면서 같이 업데이트 합니다. 이 쿼리 또한 벌크로 넣으면서 절약할 수 있었습니다.

- MongoDB 버전어 따라 write lock 개선

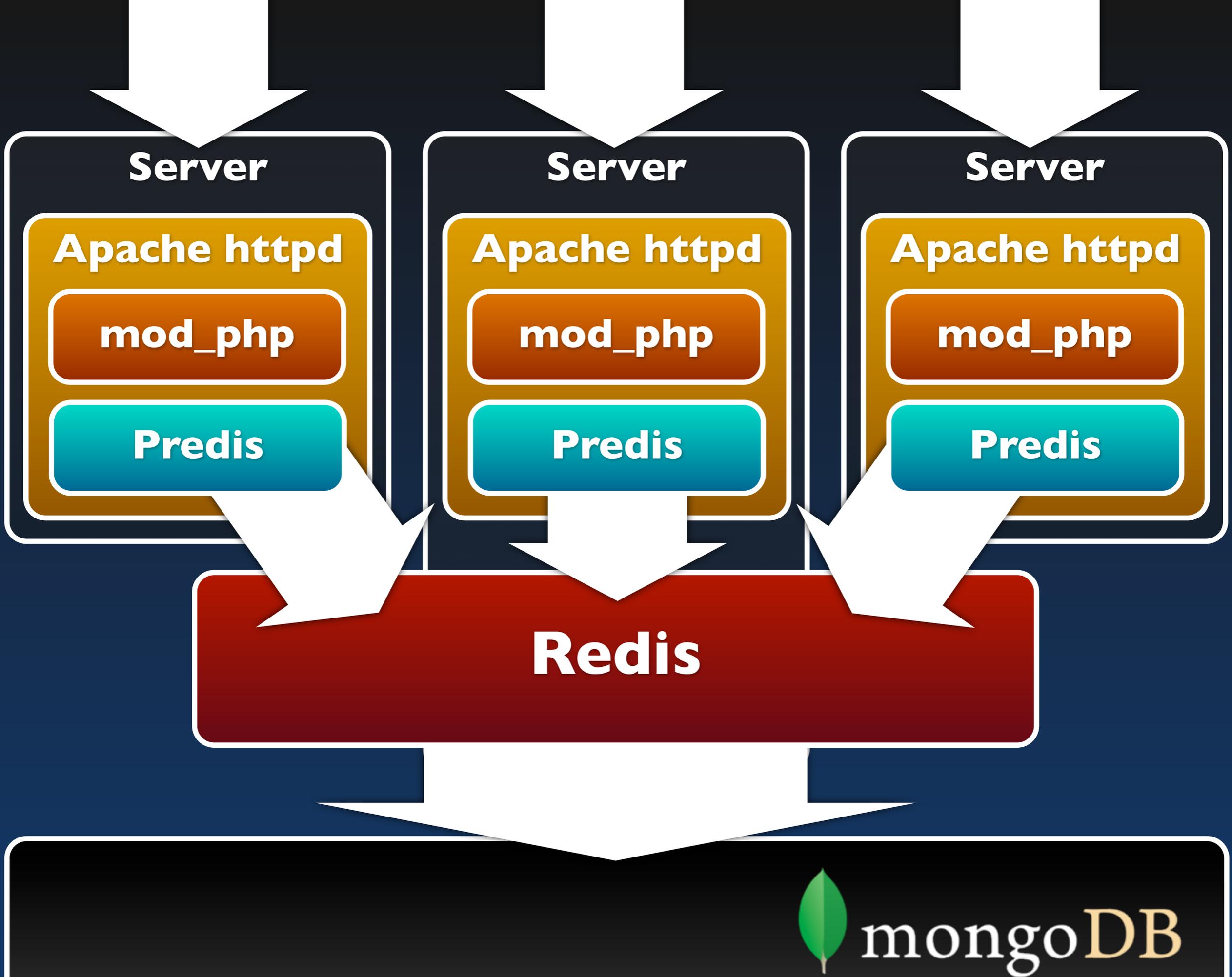
Version	Level	Description
1.6	Global	
2.0	Global	locking-with-yield
2.2	Database	
Future	Collection	

## Speaker's note

- 1.6 시절과 달리 MongoDB 는 많은 발전을 이뤄, redis / memcached 등의 레이어를 두고 쓰던 사용자들 중 일부는 2.0으로 업그레이드 하며 캐시 레이어를 제거했다는 글을 쓰기도 했습니다.
- locking-with-yield 기능을 통해 page fault 상황에서 쓰기 성능이 극적으로 개선된 벤치마크를 살펴볼 수 있습니다.
- 앞으로는 전역 lock, 데이터베이스 단위 lock을 넘어 콜렉션, 즉 테이블 수준의 락을 지원한다고 합니다. 언젠가는 RDB 수준의 object 단위 락도 기대할 수 있지 않을까 보고 있습니다.

# Link

- Removing Memcached because it's too slow
  - <http://blog.serverdensity.com/removing-memcached-because-its-too-slow/>
  - 번역 : <http://charsyam.wordpress.com/2012/06/08/>
- Goodbye global lock – MongoDB 2.0 vs 2.2
  - <http://blog.serverdensity.com/goodbye-global-lock-mongodb-2-0-vs-2-2/>
  - 번역 : <http://charsyam.wordpress.com/2012/05/24/>



**Server**

**Apache httpd**

**mod\_php**

**Predis**

**Server**

**Apache httpd**

**mod\_php**

**Predis**

**Server**

**Apache httpd**

**mod\_php**

**Predis**

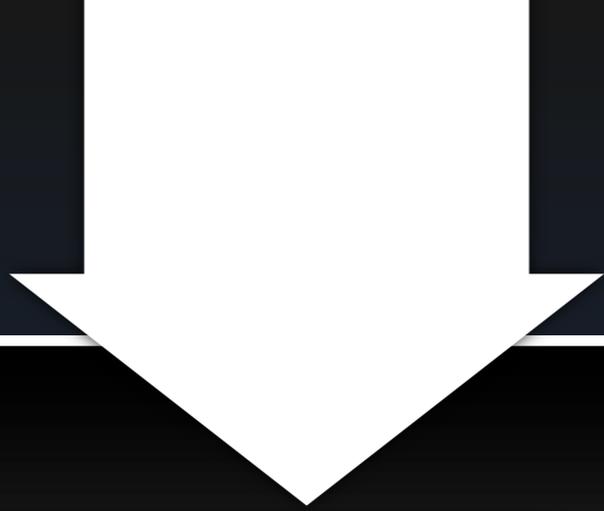
**Redis**



**mongoDB**

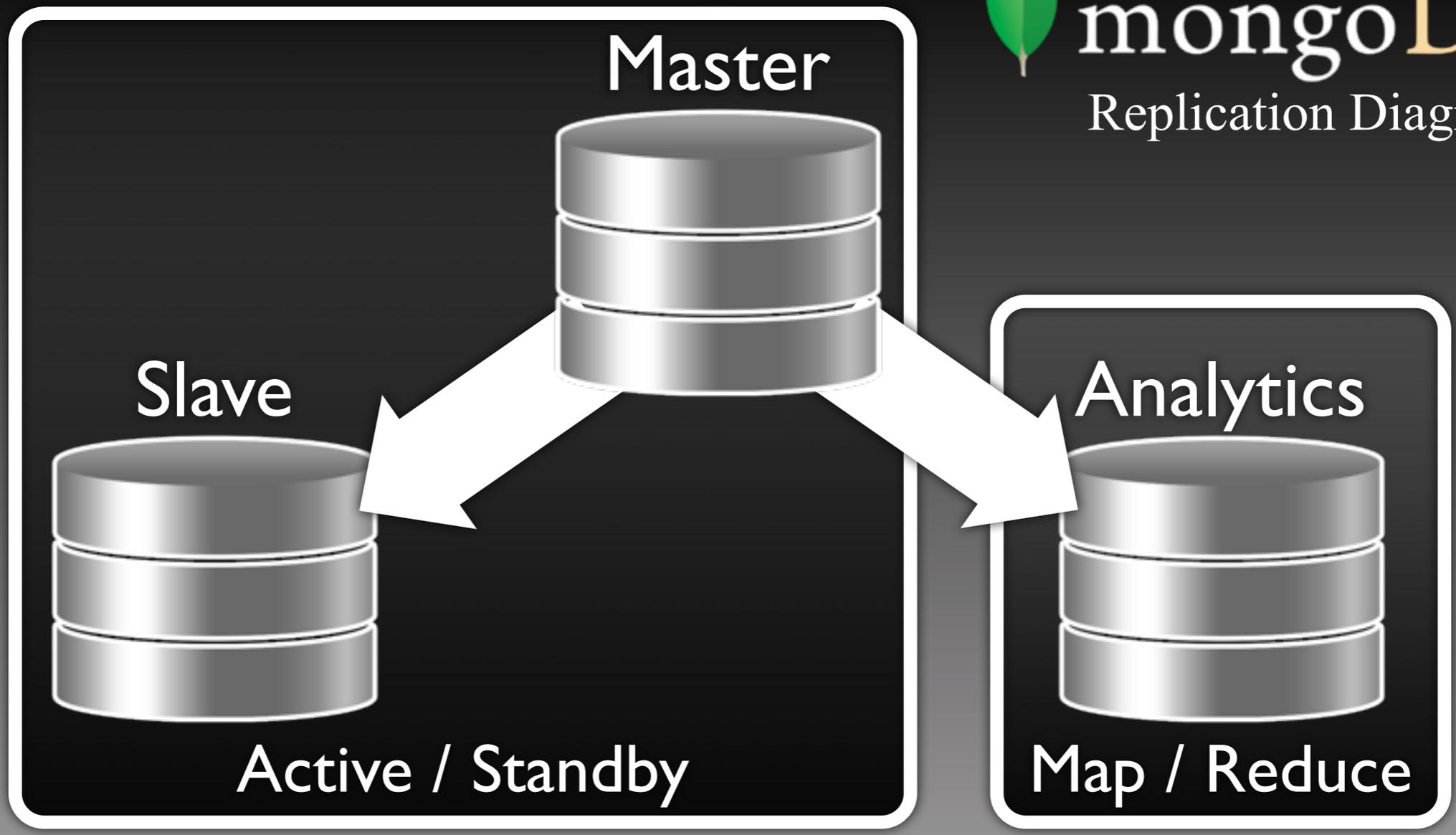
## Speaker's note

- 여러 대의 Redis 를 써야할 정도로 Redis에 부하를 주지 않아, 1대의 Redis로 옮겼고, 그 배경에는 서로 다른 Redis 에 데이터를 부어 넣음에 따른 로그의 순서 문제도 있었습니다.
- Redis에 들어가 있는 순서대로 MongoDB에 들어가게 되는데, 1분 단위로 서로 다른 타이밍에 MongoDB에 부어 넣으면서, 순차적이지 않은 문제가 있어 변경하게 되었습니다.

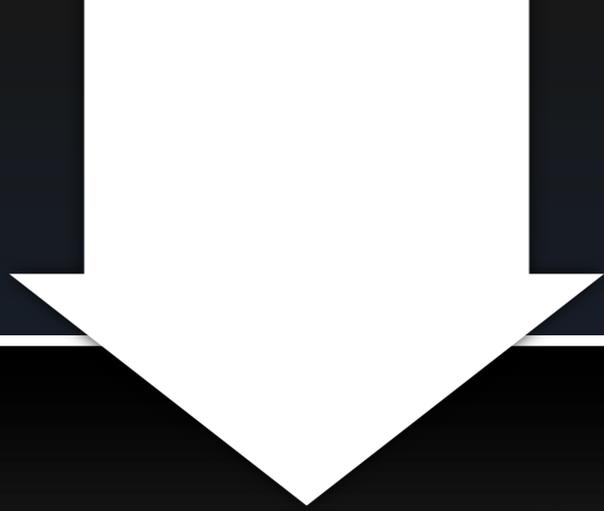


mongoDB

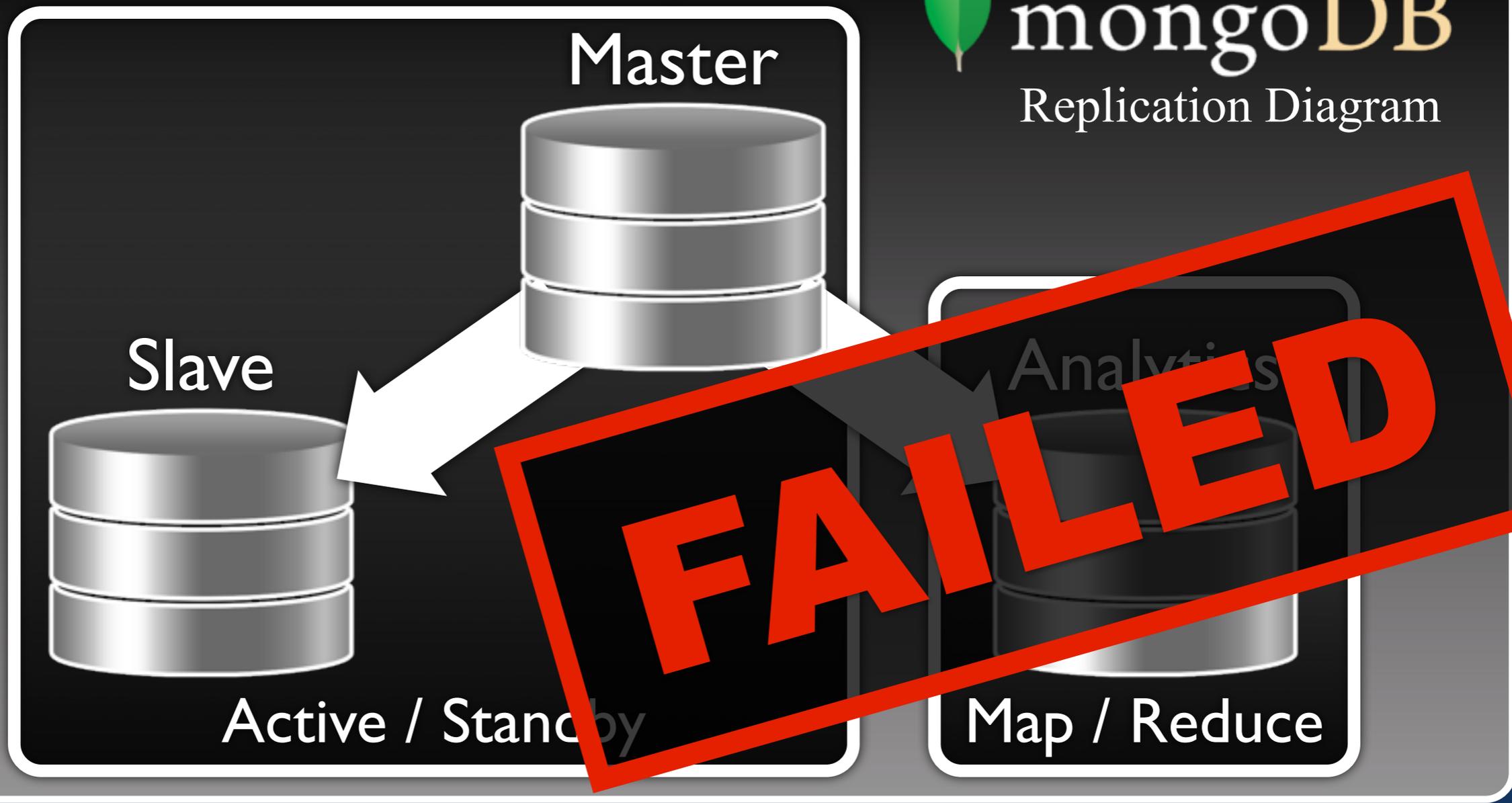
Replication Diagram



~ 10,000,000 logs/일



mongoDB  
Replication Diagram



~ 10,000,000 logs/일

# MongoDB

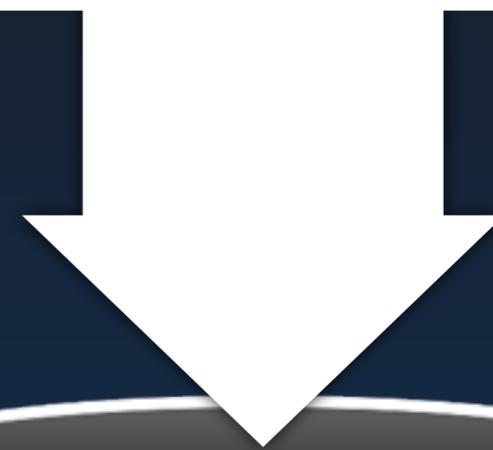
- 실 서비스는 **ReplicaSet** 추천
- 16GB RAM, SATA 2TB x 6 RAID 10 장비 사용
- 현재 약 13억 건, 1TB의 데이터를 축적
- 도저히 장비 1대로 Map / Reduce 할 수가 없음
  - Sharding 등을 포기하고 다른 방법을 시도



Active / Standby



Map / Reduce

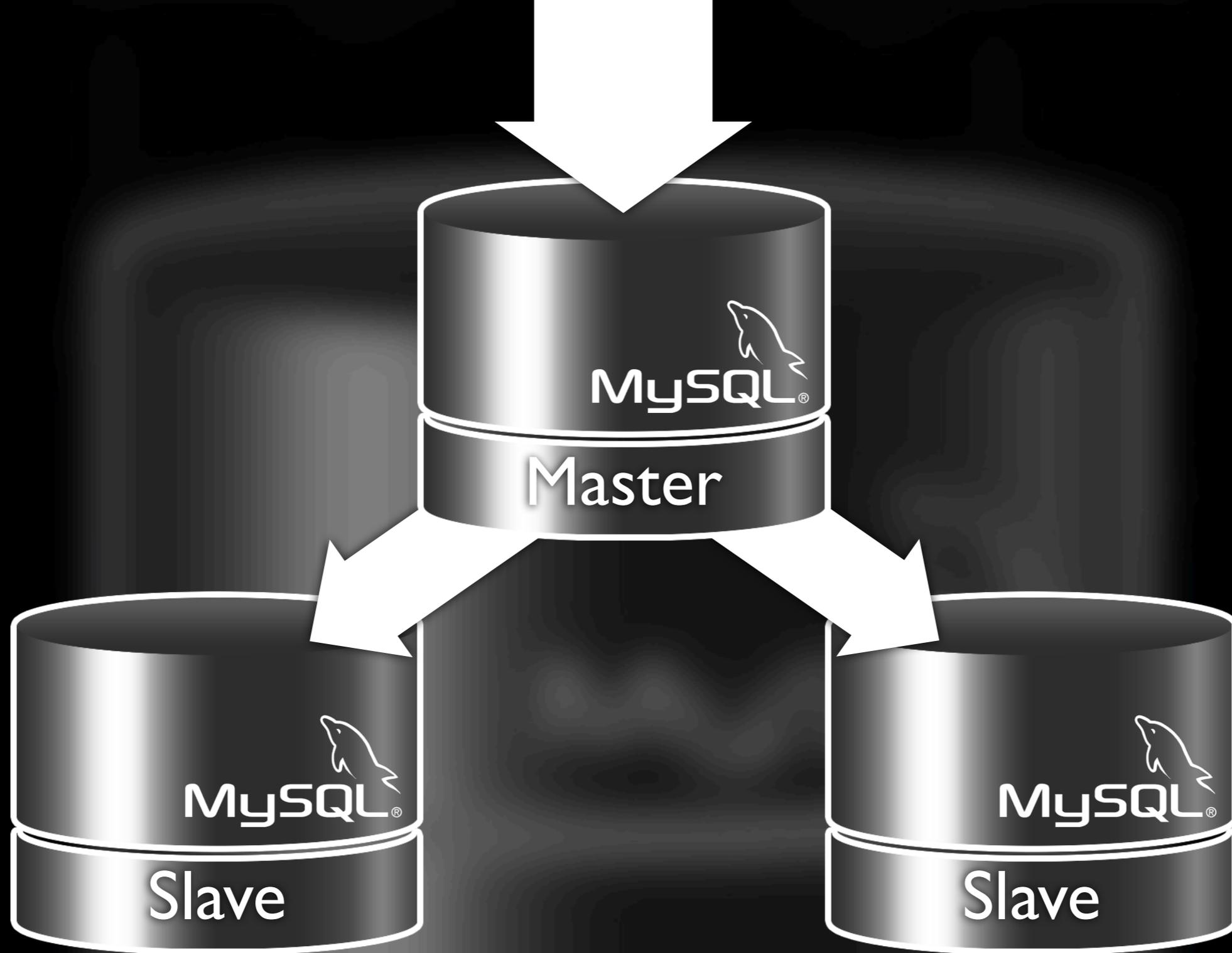




# MySQL / MyISAM

- SQL에 친숙하다
- 운용 경험이 있다
- 계속 쌓기만 하면 빠른 편





- 16GB RAM, SATA 2TB x 4 RAID 10 장비 사용

# Speaker's note

- MongoDB와 마찬가지로 3대의 MySQL 을 구동하고 있으며,
- 2+1대를 운용하는 것은, 한 대의 장비에 장애가 생겼을 경우, 2대로 구성된 M/S 구조라면 HA가 바로 깨어지는 문제가 있을 뿐 아니라, 성능 개선이나 확장을 위해 Slave 장비를 교체하는 경우에도 마찬가지입니다.
- 현재 하나의 Slave는 통계 전용으로 쓰고, 하나는 서비스에서 읽기 전용 데이터를 읽는데 사용해 낭비를 최소화 하고 있습니다.

# SQL to MongoDB

<http://www.mongodb.org/display/DOCS/SQL+to+Mongo+Mapping+Chart>

# SQL to MongoDB

## SQL

```
SELECT * FROM users
```

## MongoDB

```
db.users.find()
```



# SQL to MongoDB

## SQL

```
SELECT DISTINCT last_name FROM users
```

## MongoDB

```
db.users.distinct('last_name')
```

**FAKE**

Speaker's note : distinct 로 10,000개 이상을 처리할 수 없습니다.

# SQL to MongoDB

## SQL

```
SELECT team, SUM(score) AS total, COUNT(*)  
FROM scoreboard  
GROUP BY team  
ORDER BY total DESC
```

# SQL to MongoDB

## MongoDB

```
db.scoreboard.group({
  key: { team: true },
  initial: { total: 0, count: 0 },
  reduce: function(obj, prev) {
    prev.total += obj.score;
    prev.count += 1;
  }
})
```

**OH MY GOD**

**NO SORT**

# SQL to MongoDB

## SQL

```
SELECT msg FROM feed
WHERE user_id IN
  ( SELECT id_to FROM friend
    WHERE id_from = x )
ORDER BY written_dt
DESC LIMIT 10
```

# SQL to MongoDB

## MongoDB

```
db.feed.find(  
  { user_id: { $in: db.friend.group(  
    cond: { id_from: x }, initial:  
    { friends: [] }  
    reduce: function(obj, prev) {  
      prev.friends.push(obj.id_to);  
    }  
  )['friends'] }, { msg:1, _id:0 } )  
  .sort( { written_dt: -1 } )  
  .limit( 10 );
```

**HARDCORE**



Active / Standby



Map / Reduce

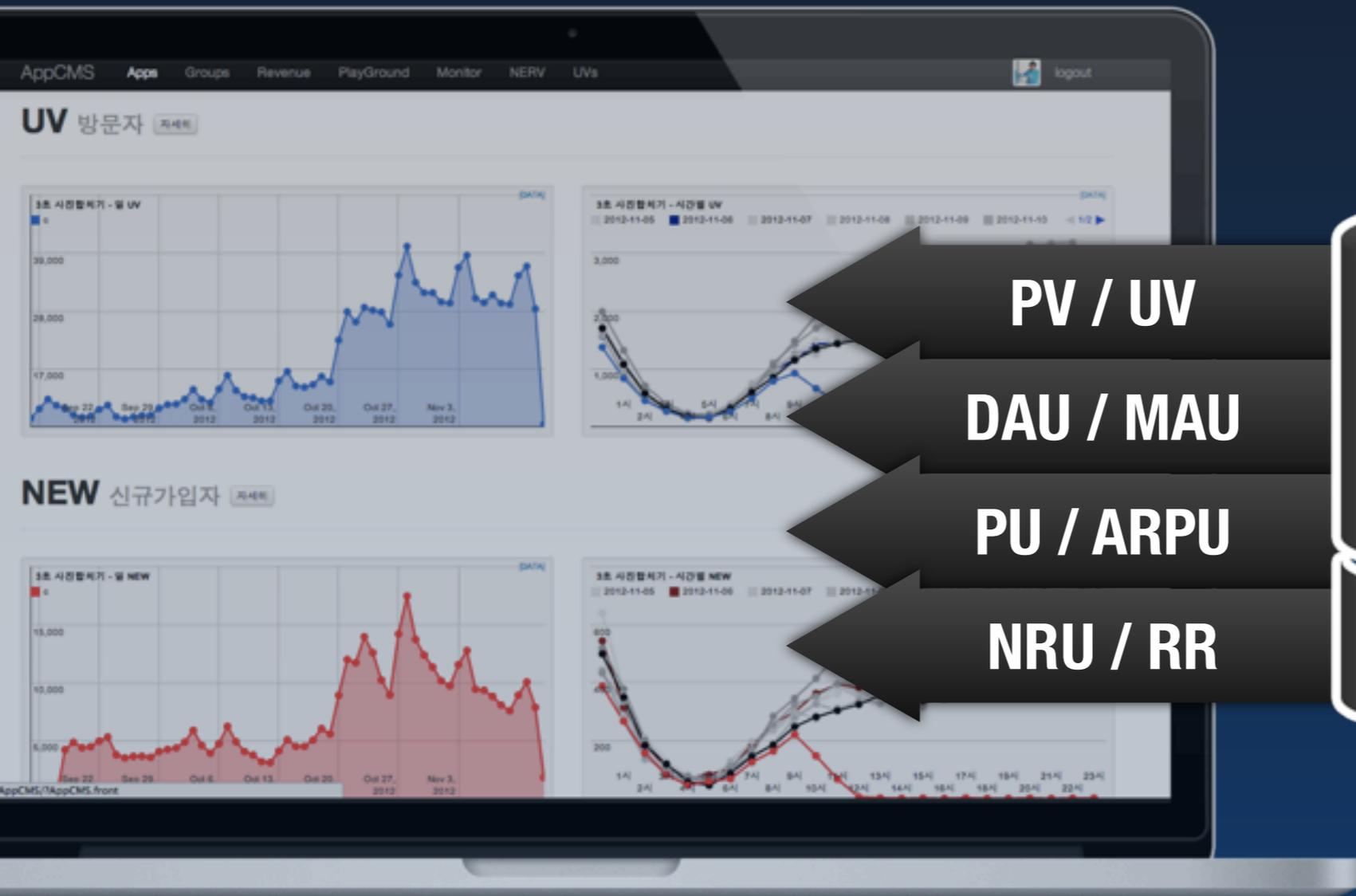


PV / UV

DAU / MAU

PU / ARPU

NRU / RR



# MongoDB to MySQL

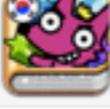
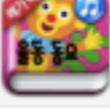
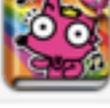
- 1분 단위로 전송
  - `_id` 값을 이용해 중복 피함
- String을 Integer로 변환 > GROUP 용이
- 익숙한 SQL을 통해 원하는 데이터 추출
- 추출된 데이터를 다시 MySQL에 저장

# Speaker's note

- 로그를 생성한 시점의 디바이스 시간이 JSON에 포함되어 있고, Redis가 받은 시간과 MongoDB로 넣은 시간까지 기록하므로, 여러 로그를 한 번에 수집하고, 1분 단위로 벌크 인서트를 수행하더라도, 로그들 사이의 상대적인 시간 관계를 확인할 수 있습니다.
- 개인 식별자, 기기 종류, 앱 ID, 이벤트 ID, 시간, 타임존, 언어 설정 등을 수집합니다.
- String에서 Integer로 변환하기 위한 Lookup 테이블을 생성하고, 이를 지속적으로 업데이트 합니다.

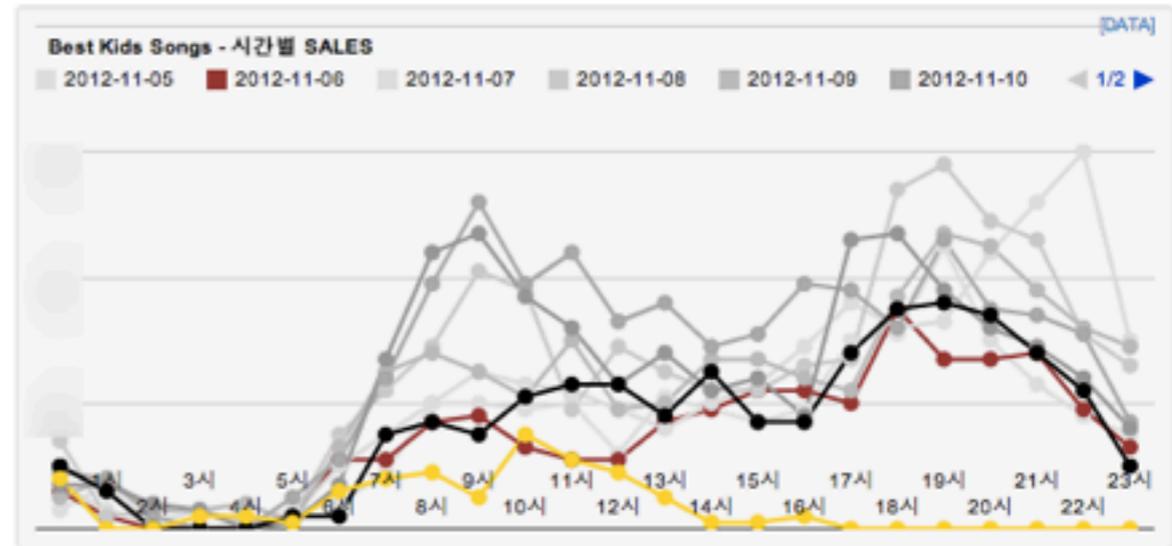
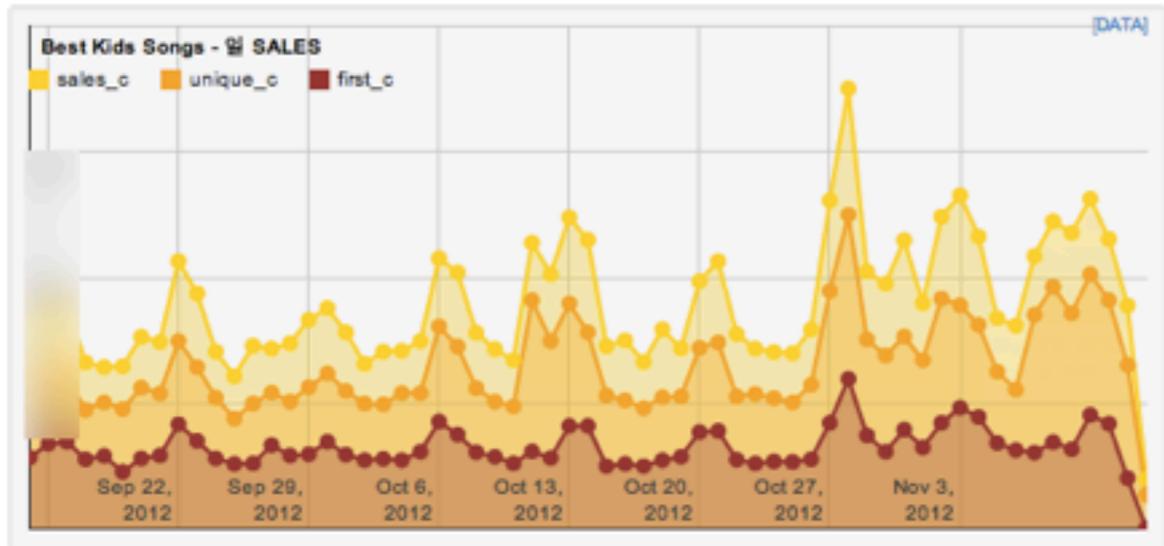
# Visualization

- 쌓여 있는 데이터를 읽어 JSON 생성
- JSON을 Google Chart API에 던짐
- Twitter Bootstrap 사용

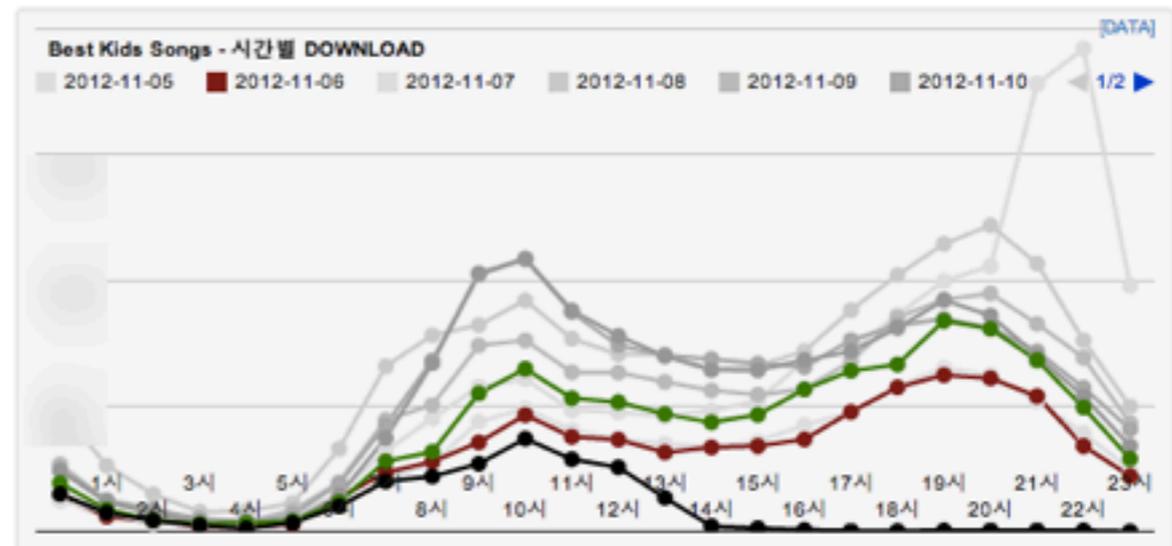
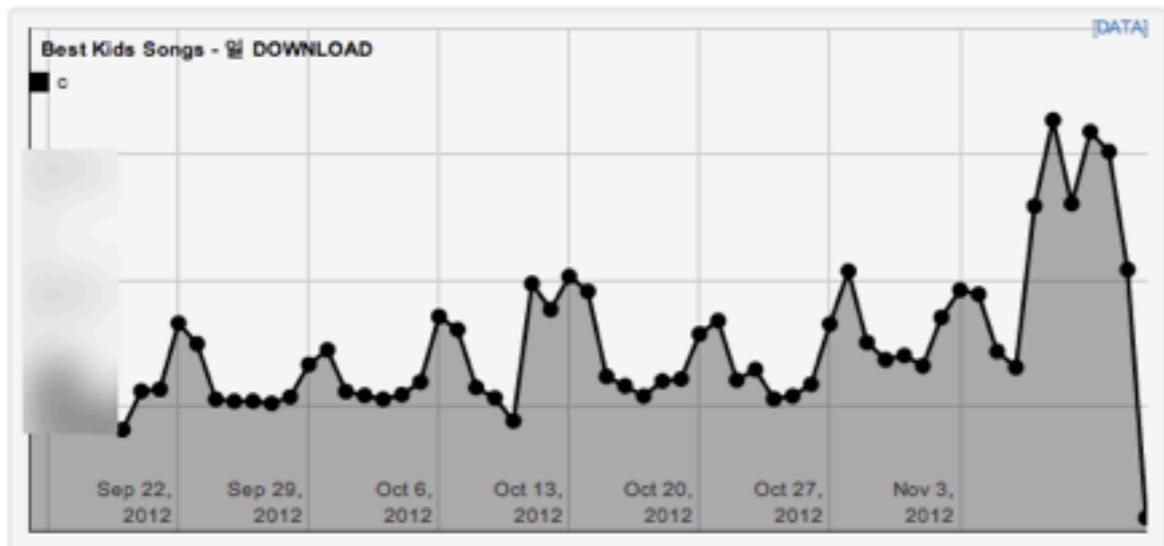
- |   |  |
|---|--|
|  6시 김성모 [7개] <span>Modify Title</span> <span>Add App</span> <span>X</span>             |  6시 무협 [3개] <span>Modify Title</span> <span>Add App</span> <span>X</span>                 |
|  6시 아트림 [2개] <span>Modify Title</span> <span>Add App</span> <span>X</span>             |  MP3 단품 (OZSTORE) [7개] <span>Modify Title</span> <span>Add App</span> <span>X</span>      |
|  MP3 단품 (PLAYSTORE) [9개] <span>Modify Title</span> <span>Add App</span> <span>X</span> |  MP3 단품 (TSTORE ONLY) [18개] <span>Modify Title</span> <span>Add App</span> <span>X</span> |
|  MP3단품 [38개] <span>Modify Title</span> <span>Add App</span> <span>X</span>             |  김성모 색칠공부 [3개] <span>Modify Title</span> <span>Add App</span> <span>X</span>              |
|  만화류 [6개] <span>Modify Title</span> <span>Add App</span> <span>X</span>              |  매거진류 [3개] <span>Modify Title</span> <span>Add App</span> <span>X</span>                |
|  미끼류 [34개] <span>Modify Title</span> <span>Add App</span> <span>X</span>             |  미끼류 (주요앱) [5개] <span>Modify Title</span> <span>Add App</span> <span>X</span>           |
|  베비풍 [6개] <span>Modify Title</span> <span>Add App</span> <span>X</span>              |  보들북 단품 [41개] <span>Modify Title</span> <span>Add App</span> <span>X</span>             |
|  보들북 인앱 [16개] <span>Modify Title</span> <span>Add App</span> <span>X</span>          |  보들북+MP3 영어권 [9개] <span>Modify Title</span> <span>Add App</span> <span>X</span>         |
|  보들북+MP3 일어권 [10개] <span>Modify Title</span> <span>Add App</span> <span>X</span>     |  색칠놀이 [4개] <span>Modify Title</span> <span>Add App</span> <span>X</span>                |



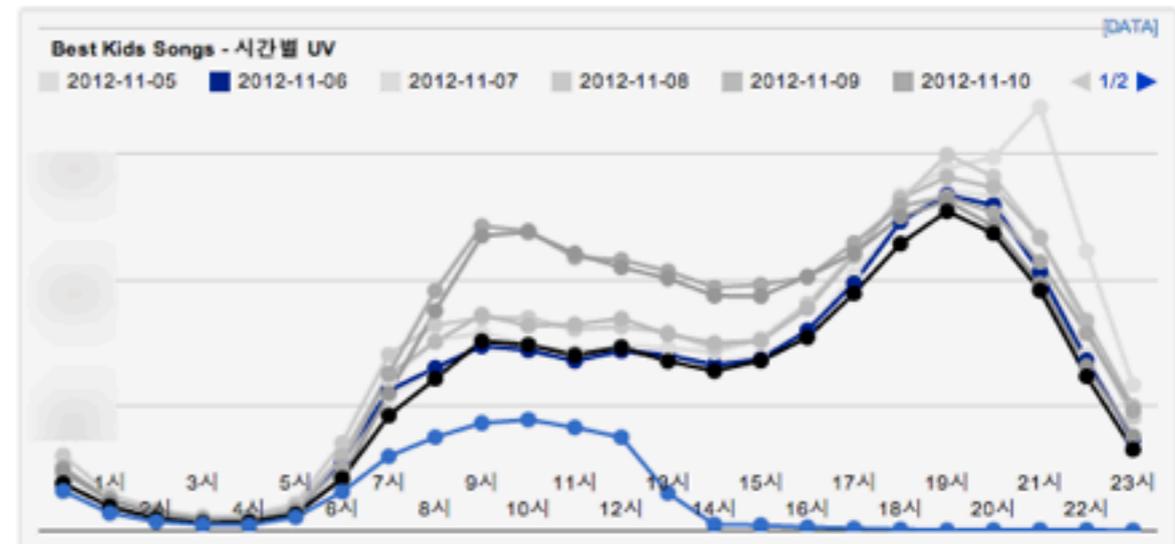
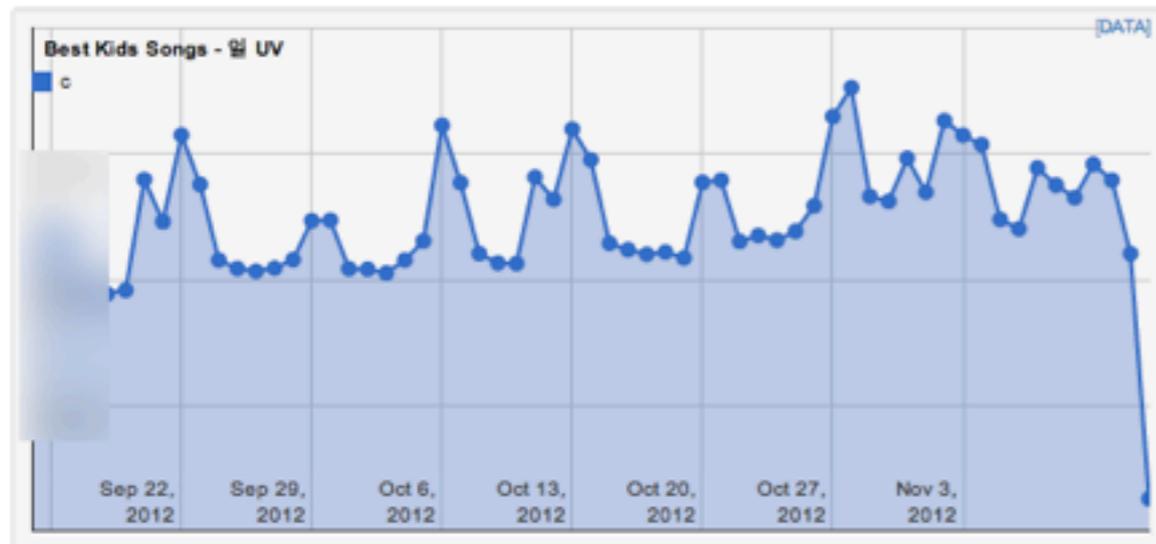
# SALES 판매량 [자세히](#)



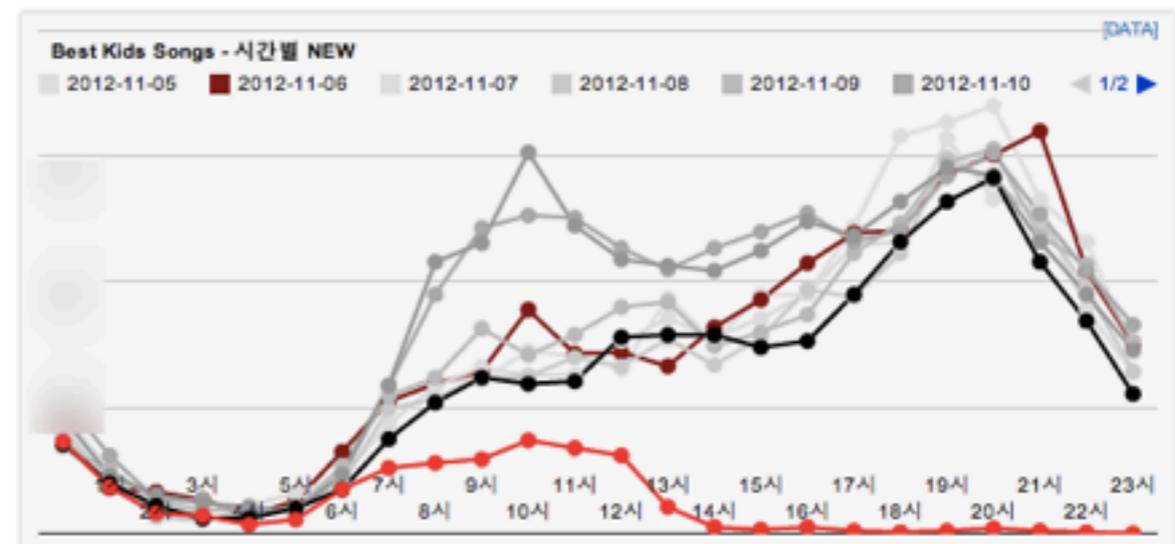
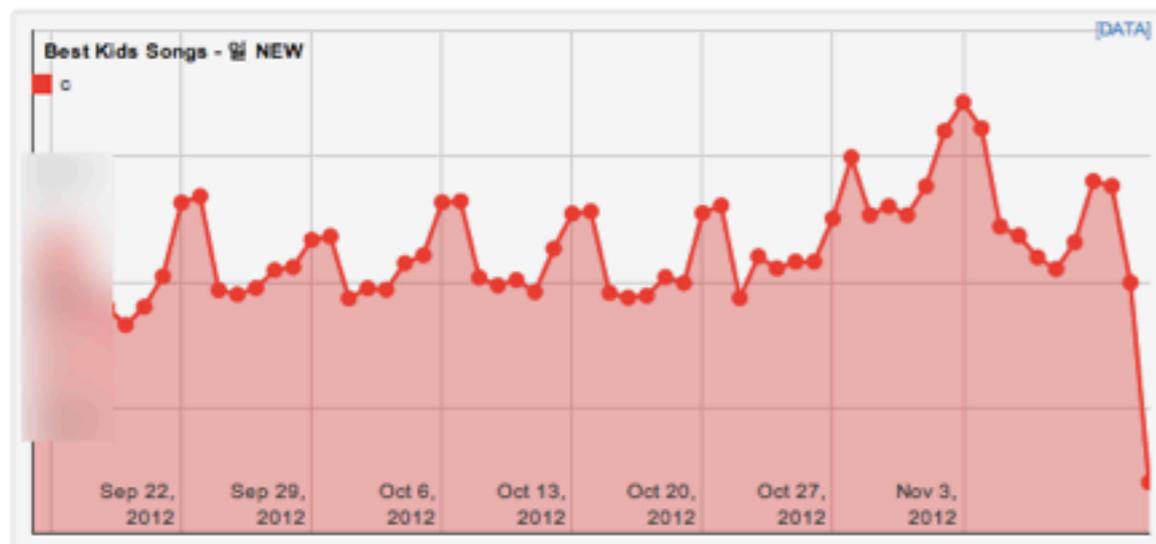
# DOWNLOAD 다운로드 [자세히](#)



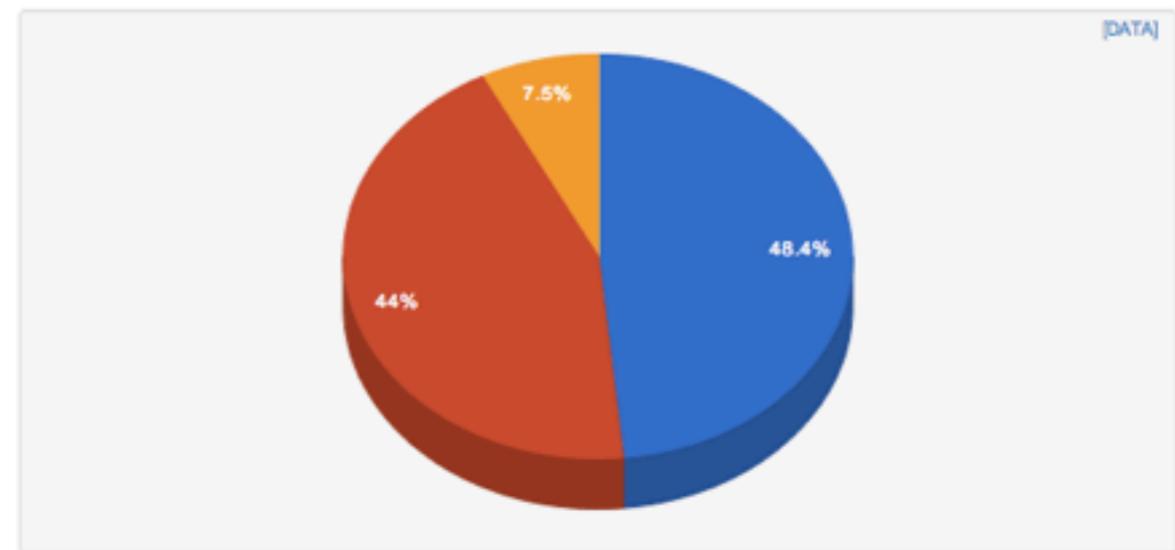
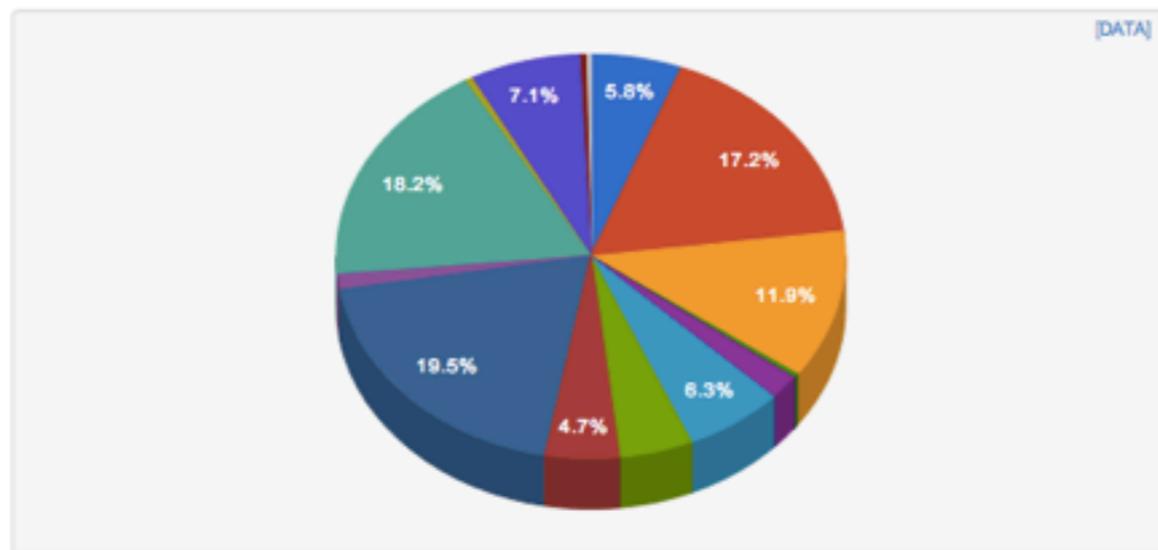
## UV 방문자 [자세히](#)



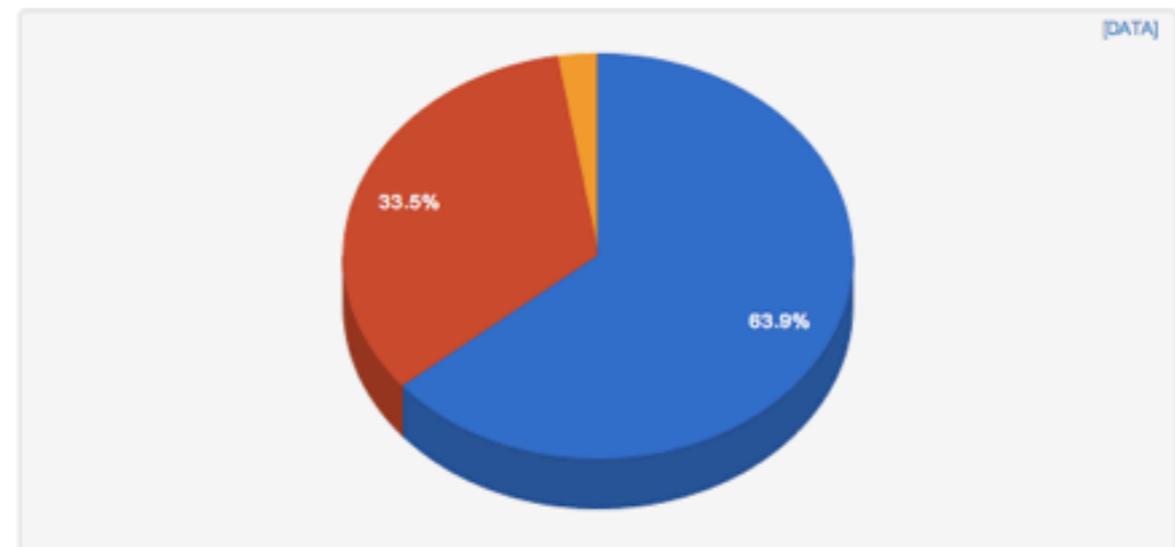
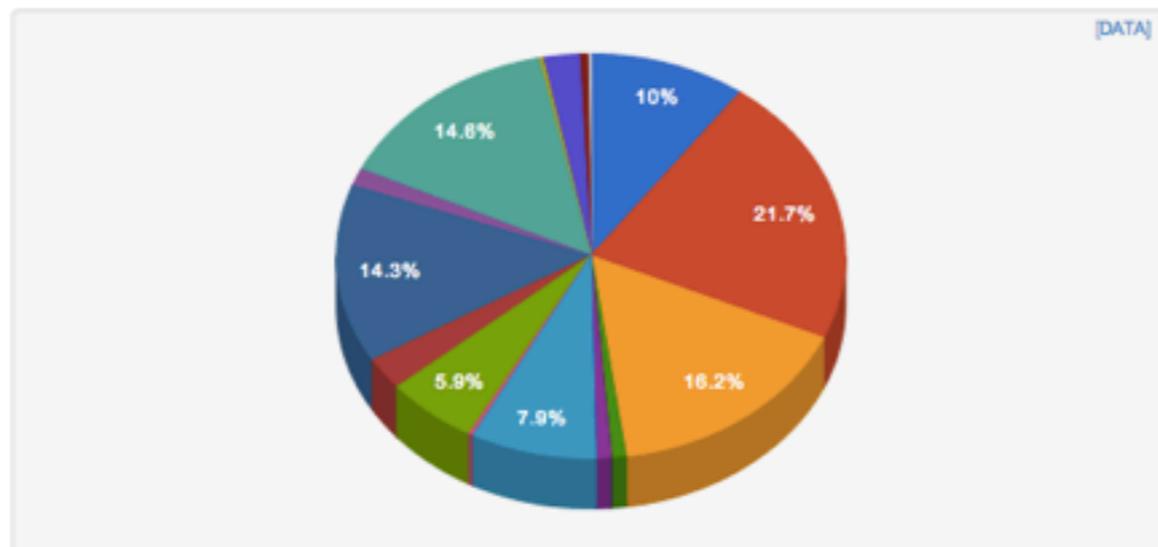
## NEW 신규가입자 [자세히](#)



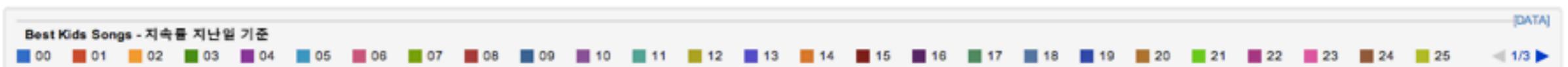
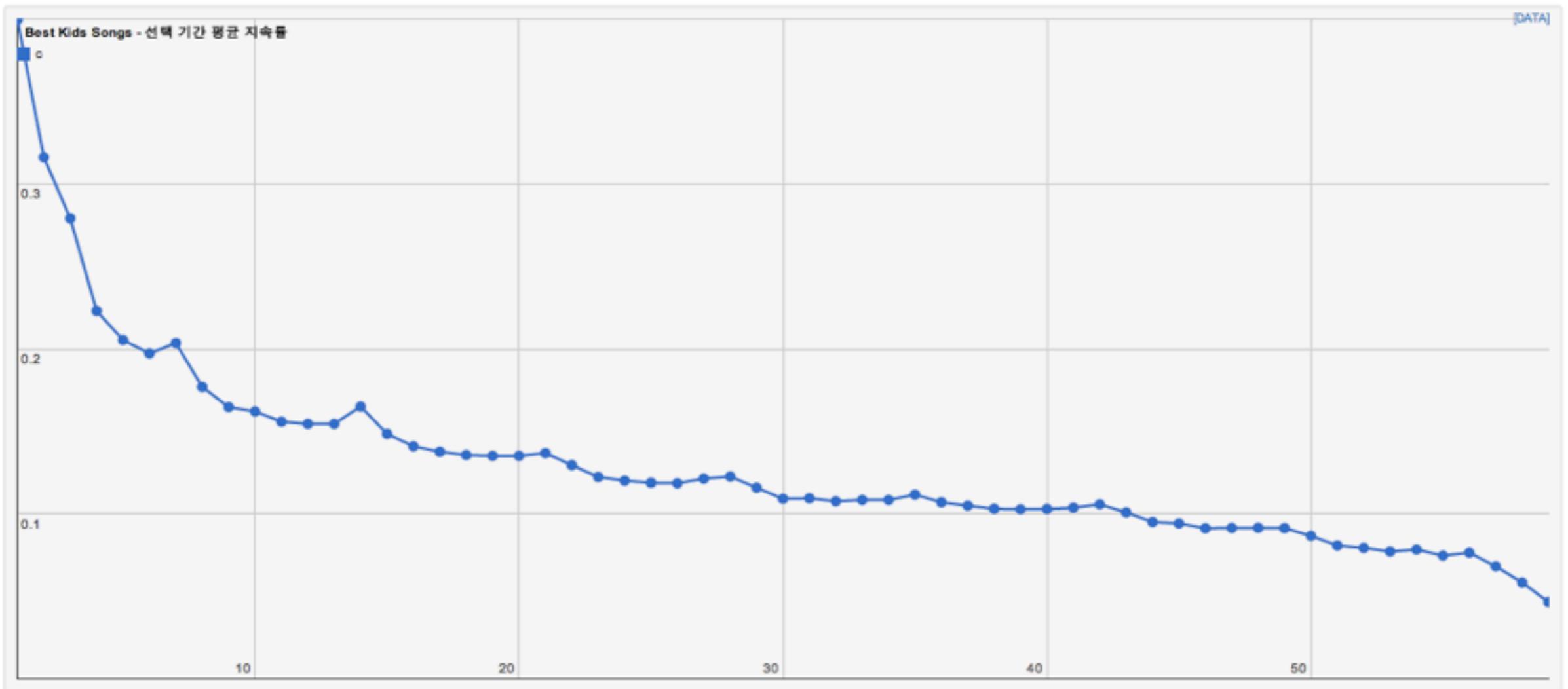
## Members Group by Device 기기별 가입자 구분



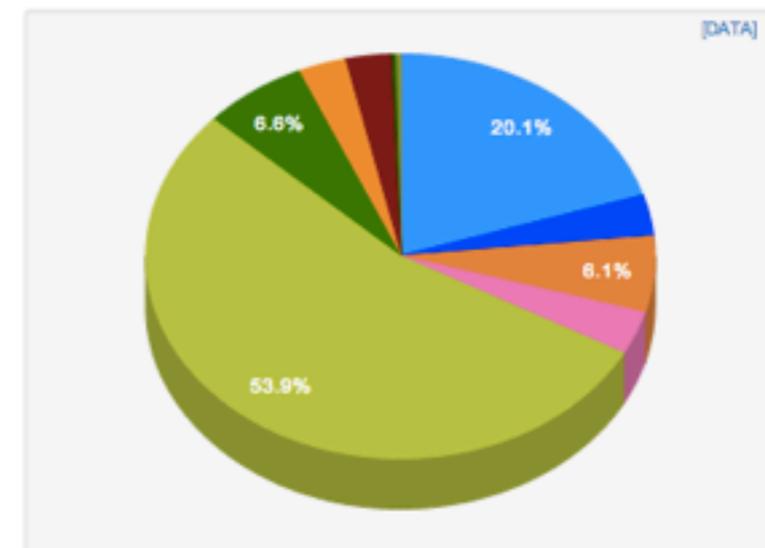
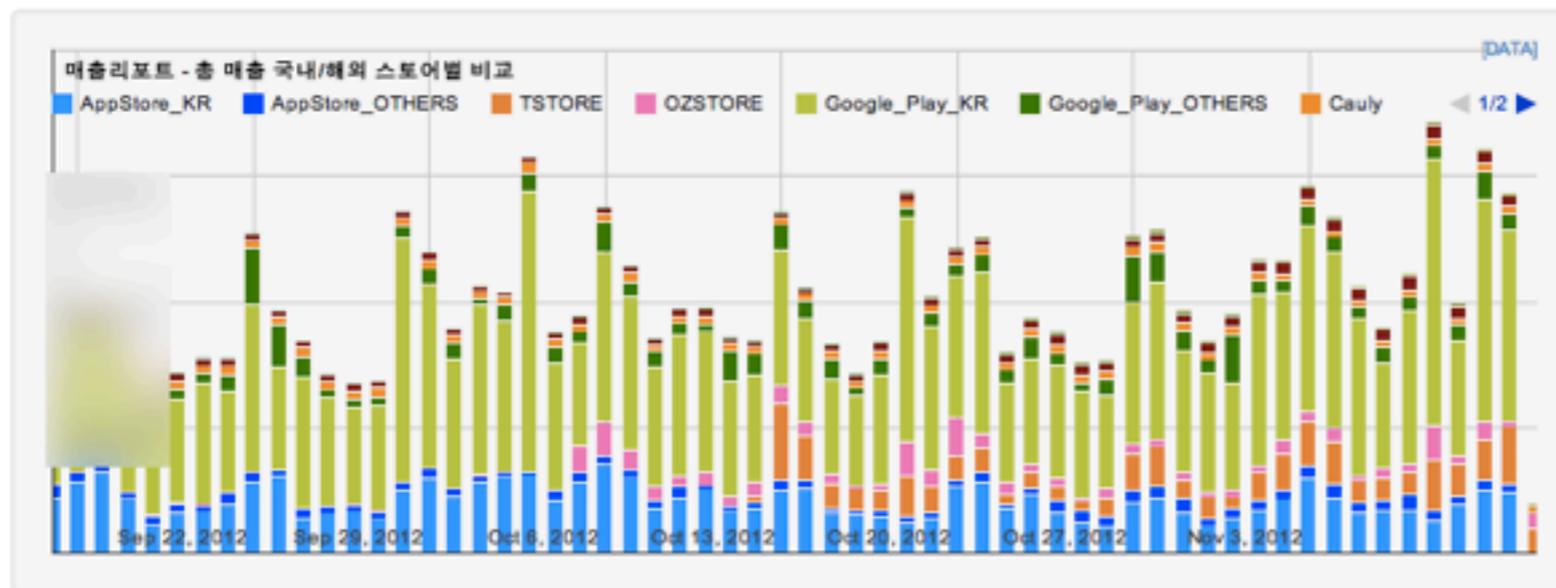
## Sales Group by Device 기기별 구매자 구분



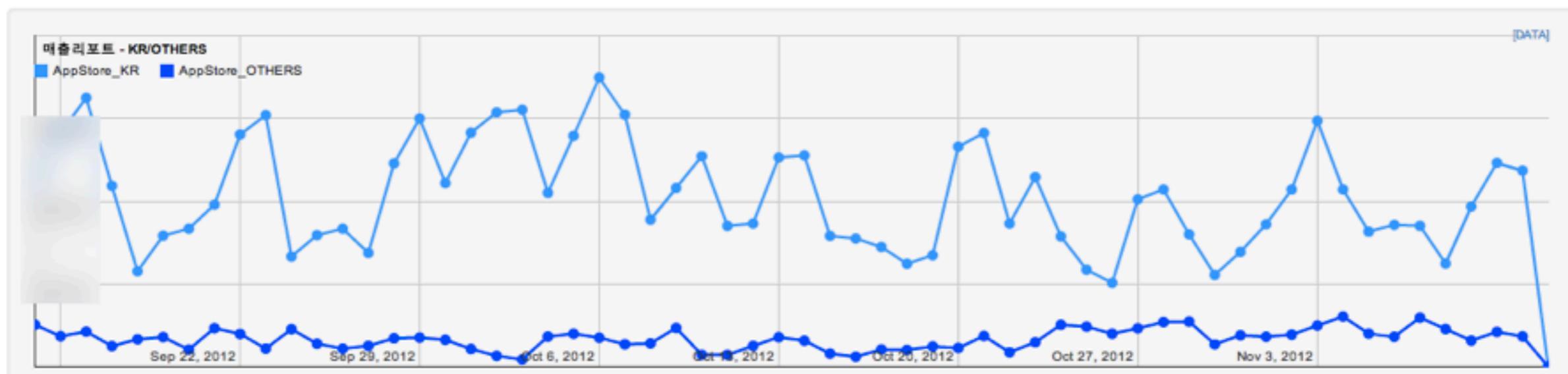
## RR 지속률 자세히



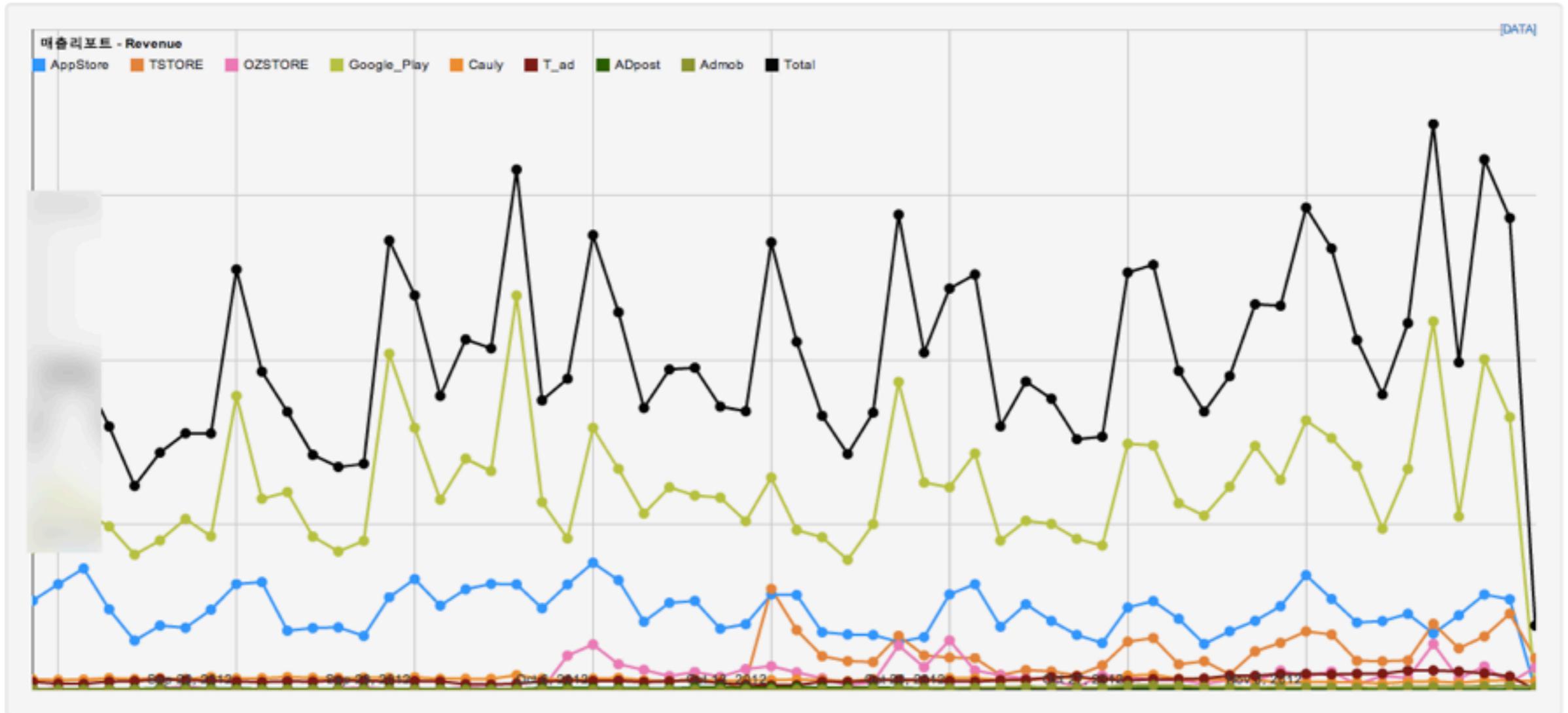
## Stores 총 매출 국내/해외 스토어별 비교



## AppStore 애플 앱스토어

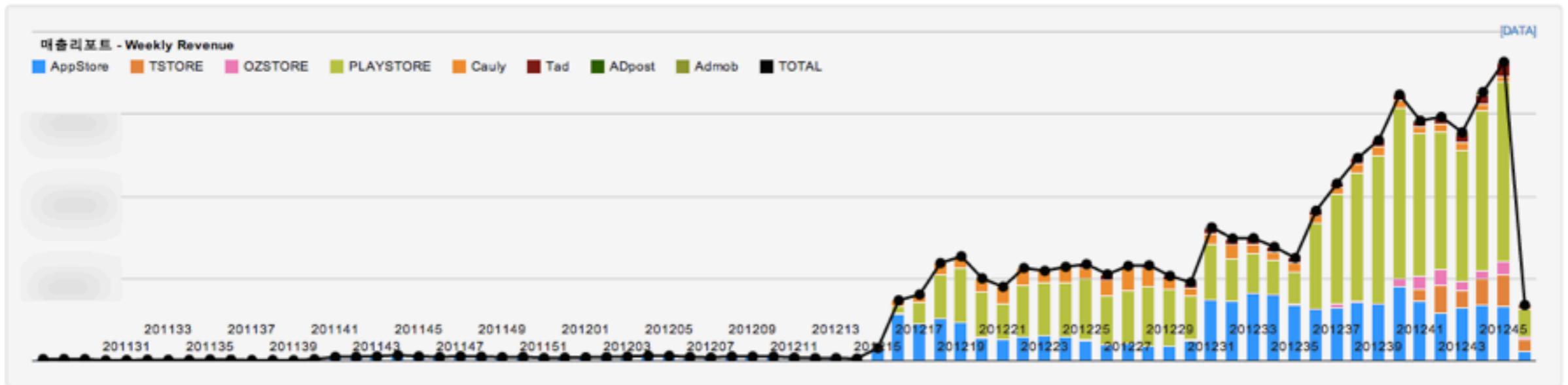


# Daily 일매출

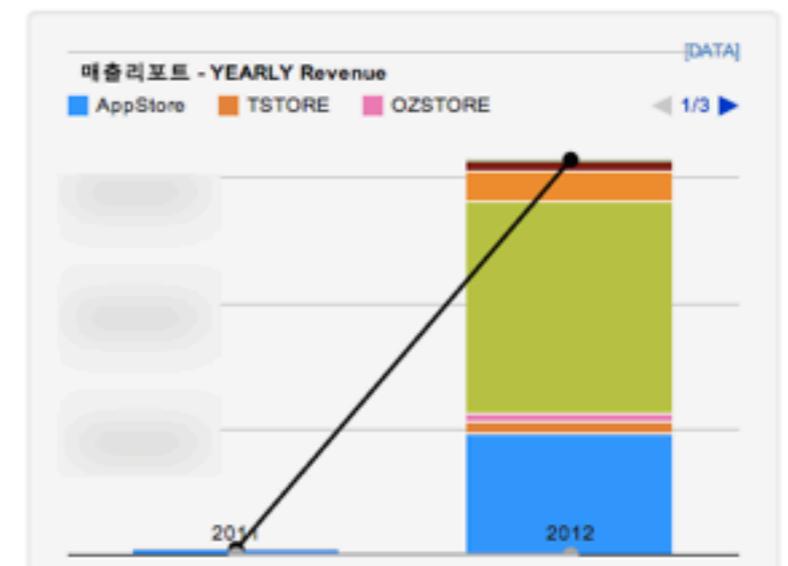
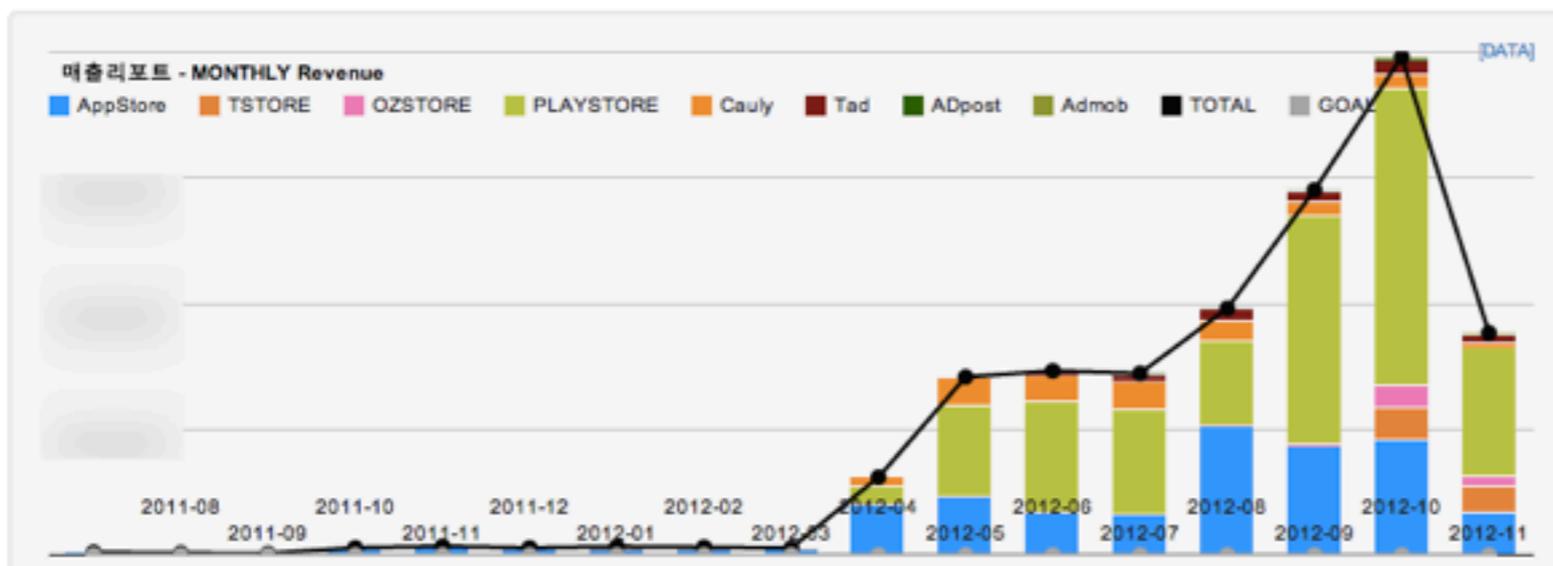


# Weekly 주매출

## Weekly 주매출



## Monthly/Yearly 월/년매출





과연 잘 하고 있는 걸까?

# 바퀴의 재발명

<b>Fluentd</b>	Treasure Data	Ruby	Lightweight
<b>Flume</b>	Apache	Java	Simple, Cloudera support
<b>Chukwa</b>	Apache Hadoop	Java	Hadoop based, Rich metadata
<b>Scribe</b>	Facebook	C++	Complex, but fast

# 바퀴의 재발명

<b>Fluentd</b>	<a href="http://fluentd.org/">http://fluentd.org/</a>
<b>Flume</b>	<a href="http://flume.apache.org/">http://flume.apache.org/</a>
<b>Chukwa</b>	<a href="http://incubator.apache.org/chukwa/">http://incubator.apache.org/chukwa/</a>
<b>Scribe</b>	<a href="https://github.com/facebook/scribe">https://github.com/facebook/scribe</a>

# 바퀴의 재발명



# 바퀴의 재발명

<b>baas.io</b>	<a href="http://baas.io/">http://baas.io/</a>
<b>Google Analytics</b>	<a href="http://google.com/analytics">http://google.com/analytics</a>
<b>Flurry</b>	<a href="http://www.flurry.com/">http://www.flurry.com/</a>

# Speaker's note

- 이미 다양한 로그 수집 애플리케이션이 존재하므로, 알아보지 않고 새로운 것을 만드는 것은 낭비입니다.
- 수집하고자 하는 내용과 분량에 따라 적합한 솔루션을 선택해 기회 비용을 낭비하지 않을 수 있습니다.
- 로그 수집뿐 아니라 분석과 통계까지 제공하는 다양한 솔루션을 활용할 수도 있습니다.

# Conclusion

- 달라진 환경에 대응하는 수집 전략
- 꼭 한가지 기술만 쓸 필요 없음
- 이미 알고 있던 기술을 최대한 활용
- 역할에 맞는 일을 맡겨, 장점을 최대화



감사합니다