오픈소스 데이터베이스 은행 서비스에 첫발을 내밀다.

DEVIEW 2017

성동찬
인프라 파트

kakaoBank

About ME

DEVIEW 2017

절대 깨지지 않는 견고한 서비스를 지향하는 국내 최초(아마도) 은행 오픈소스 데이터베이스 엔지니어

우육빛깔 까칠행원

from kakaoBank

(KT하이텔 > 티몬 > 카카오 > 한국카카오은행)

http://gywn.net
https://www.facebook.com/dongchan.sung

CONTENTS

DEVIEW 2017

INTRO

여기는 고객의 돈을 다루는 은행이다

데이터는 완벽해야 한다

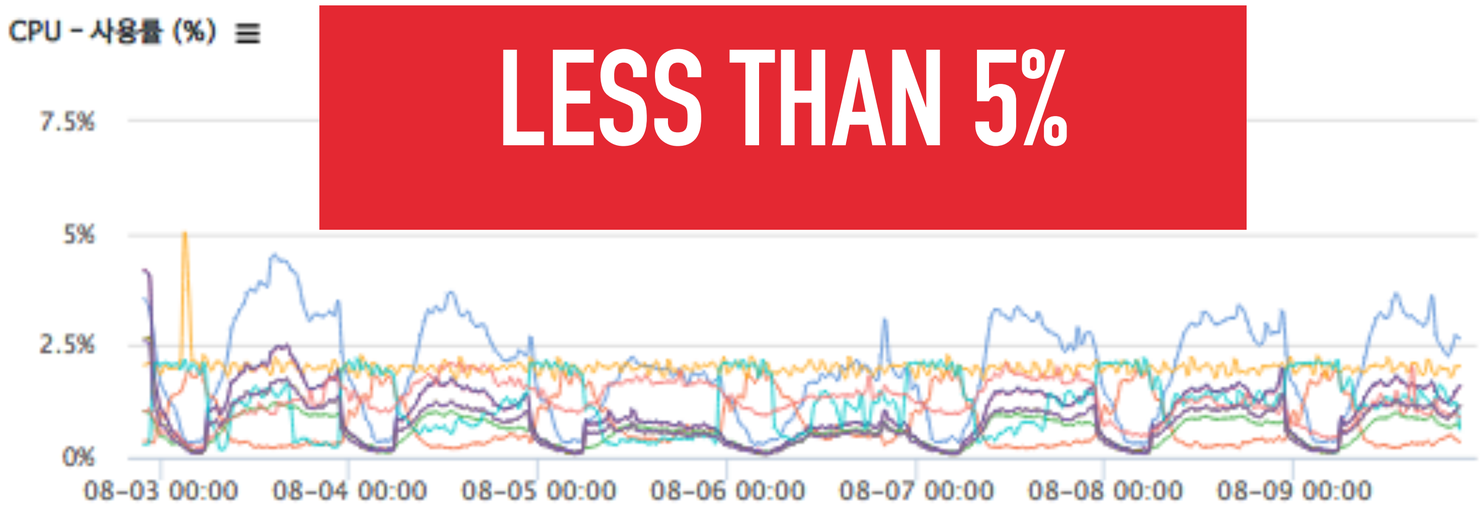
1.
MySQL in Banking Service

DEVIEW
2017

1. MySQL in Banking Service

DEVIEW 2017

Q. 금융 서비스에 오픈소스 데이터베이스를 써도 될까요?



A. 장애 없이 잘 쓰고 있습니다.

1. MySQL in Banking Service

DEVIEW 2017

Q. mysql이 어디에 쓰이나요?

1. MySQL in Banking Service

DEVIEW 2017

Q. 여기 장애나면?

1. MySQL in Banking Service

DEVIEW 2017

Q. 트래픽은 어때요?

초당 트래픽

5,000 DML, 10,000 QUERIES
10,000 DML, 20,000 QUERIES

A. 엄청 들어와요. 화들짝 놀랐어요.

1. MySQL in Banking Service

DEVIEW 2017

Q. 그래서 써도 된다고요?

금융 서비스에 잘 쓰고 있고
중요 서비스에 잘 활용되고 있고
트래픽도 엄청 많이 들어옵니다

A. 네, 쓰세요. 대신 잘 쓰셔야 해요.

(지금부터 잘 쓰기 위해 우리가 고민했던 이야기 풀어보겠습니다.)

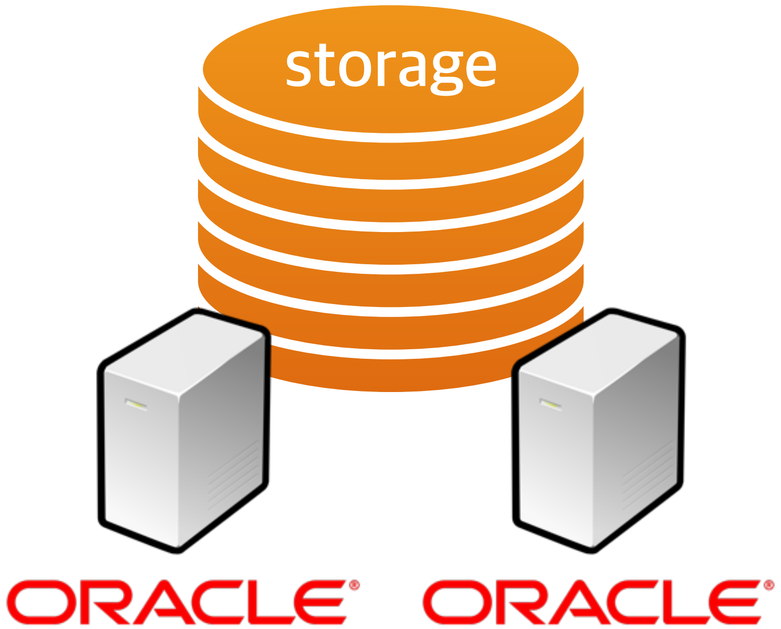
2.
Stability with MySQL

DEVIEW
2017

2. Stability with MySQL

DEVIEW 2017

Oracle RAC & MySQL Replication

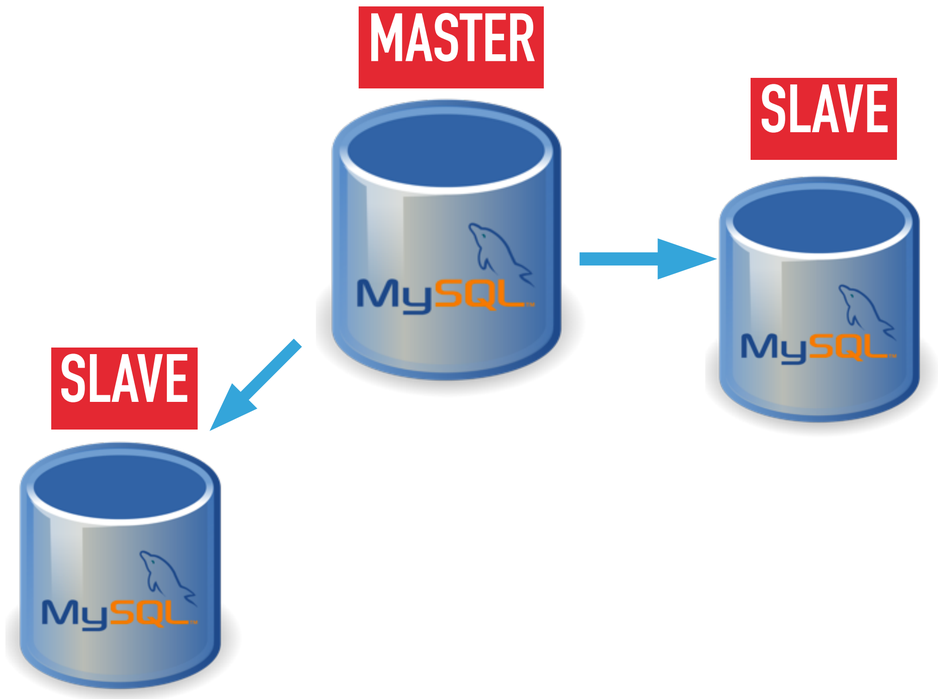


- Synchronous
- Active / Active
- Shared Storage

2. Stability with MySQL

DEVIEW 2017

Oracle RAC & **MySQL Replication**



- Asynchronous
- Active / Standby
- Shared Nothing

2. Stability with MySQL

DEVIEW 2017

에서 제공하고, 우리가 선택한 **Replication**은..

마스터에서 “커밋”된 데이터를

Async하게 **디스크에** 복제 저장한다

2. Stability with MySQL

DEVIEW 2017

마스터/슬레이브간 데이터가 일치하지 않을 수 있다.

마스터 장애시 유실이 발생할 수 있다.

노드 확장이 쉽지 않다

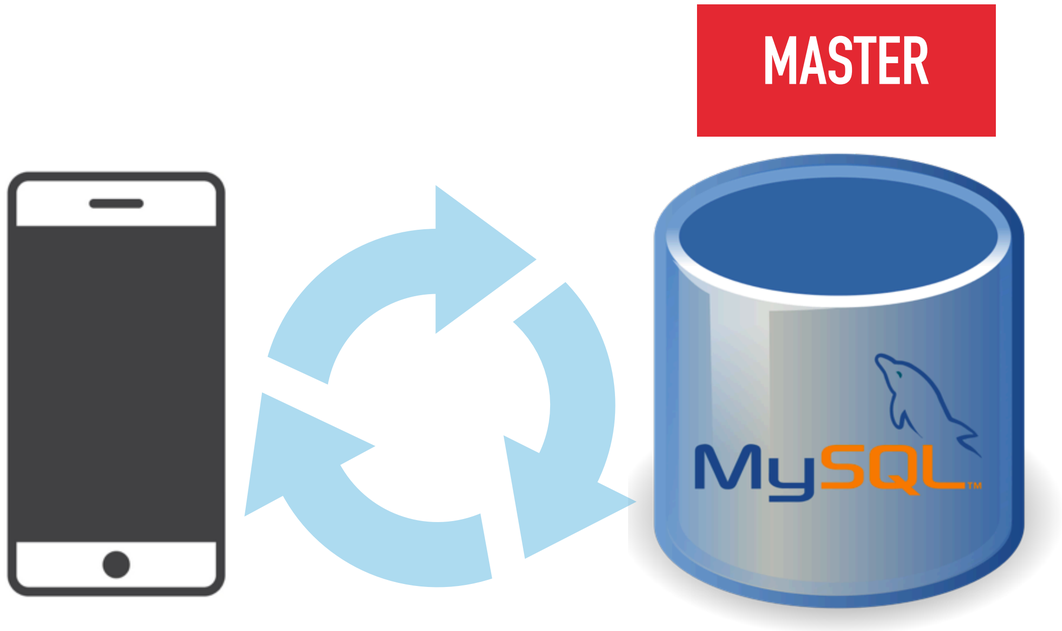
저장 공간은 유한하다.

최고 수준의 신뢰를 얻어야 한다.

2. Stability with MySQL

DEVIEW 2017

Q. 서비스 데이터 일관성 보장은?

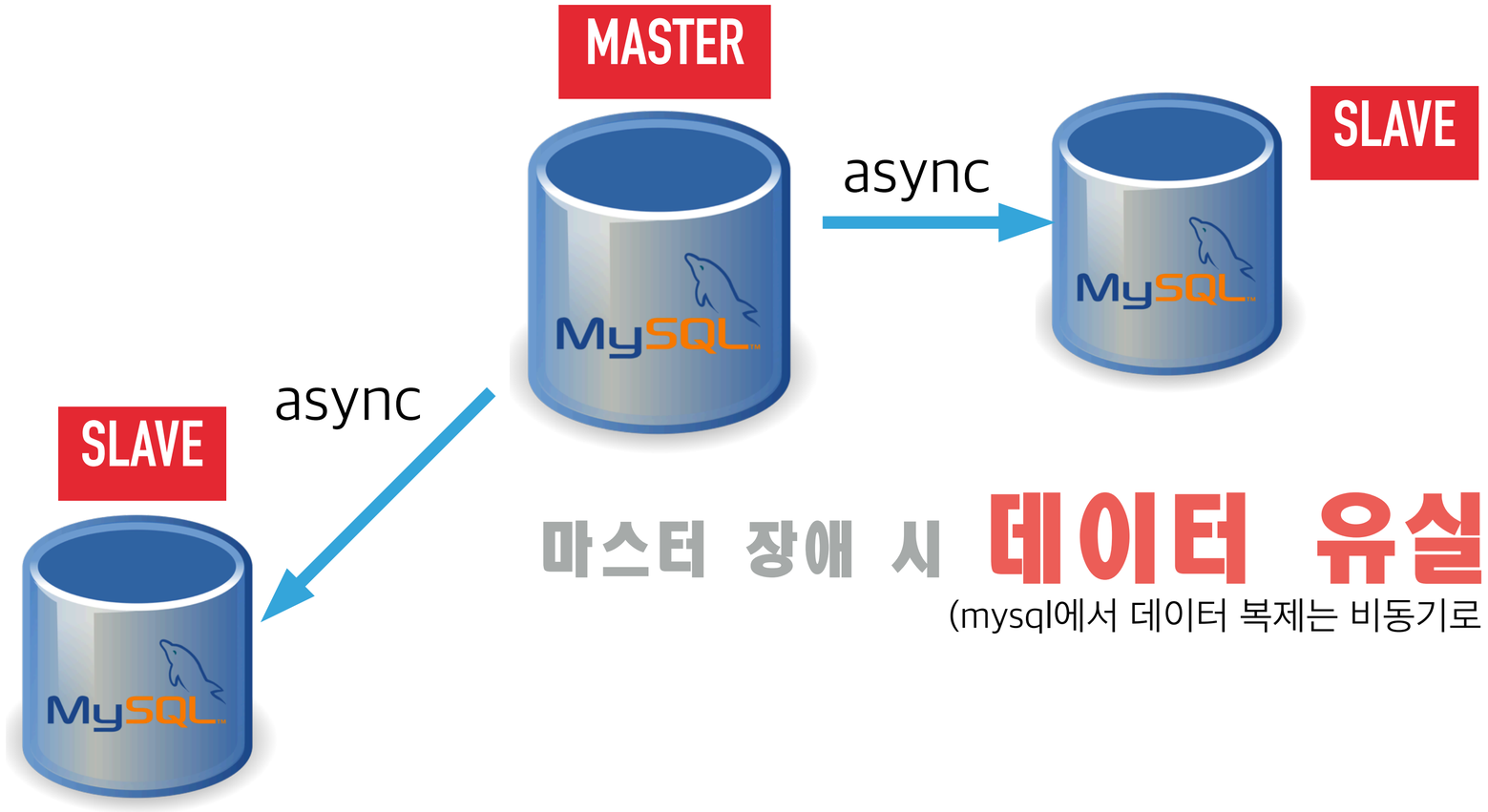


서비스는 마스터에서 노드에서!

2. Stability with MySQL

DEVIEW 2017

Q. 데이터 이중화는?



2. Stability with MySQL

DEVIEW 2017

Q. 주어진 환경과 목표

마스터에서만 서비스

비동기로 데이터 복제

공유 스토리지는 없다

데이터가 일치할 필요는 없다.

복구 시 데이터 유실만 없으면 된다.

2. Stability with MySQL

DEVIEW 2017

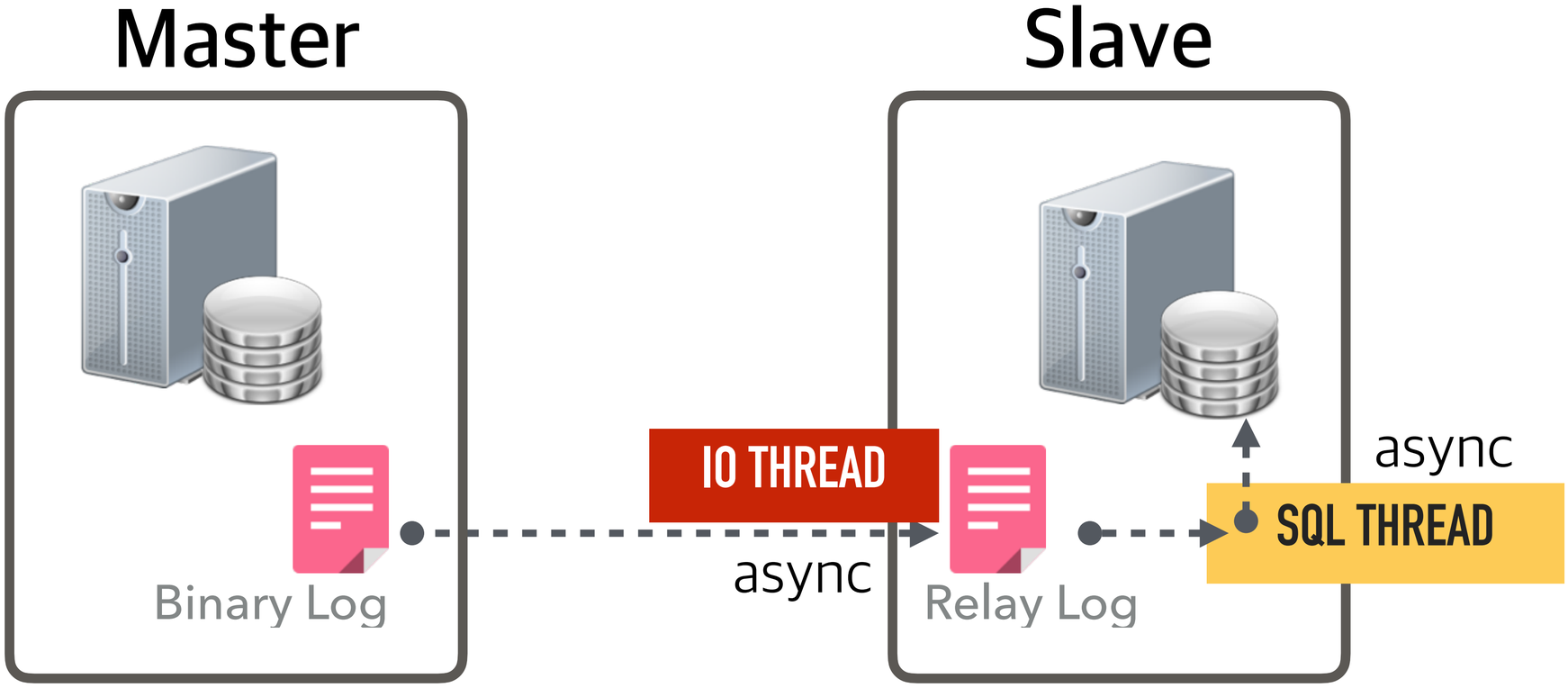
LOSSLESS REPLICATION

슬레이브 어딘가에 변경 이력이 반드시 남아있다

2. Stability with MySQL

DEVIEW 2017

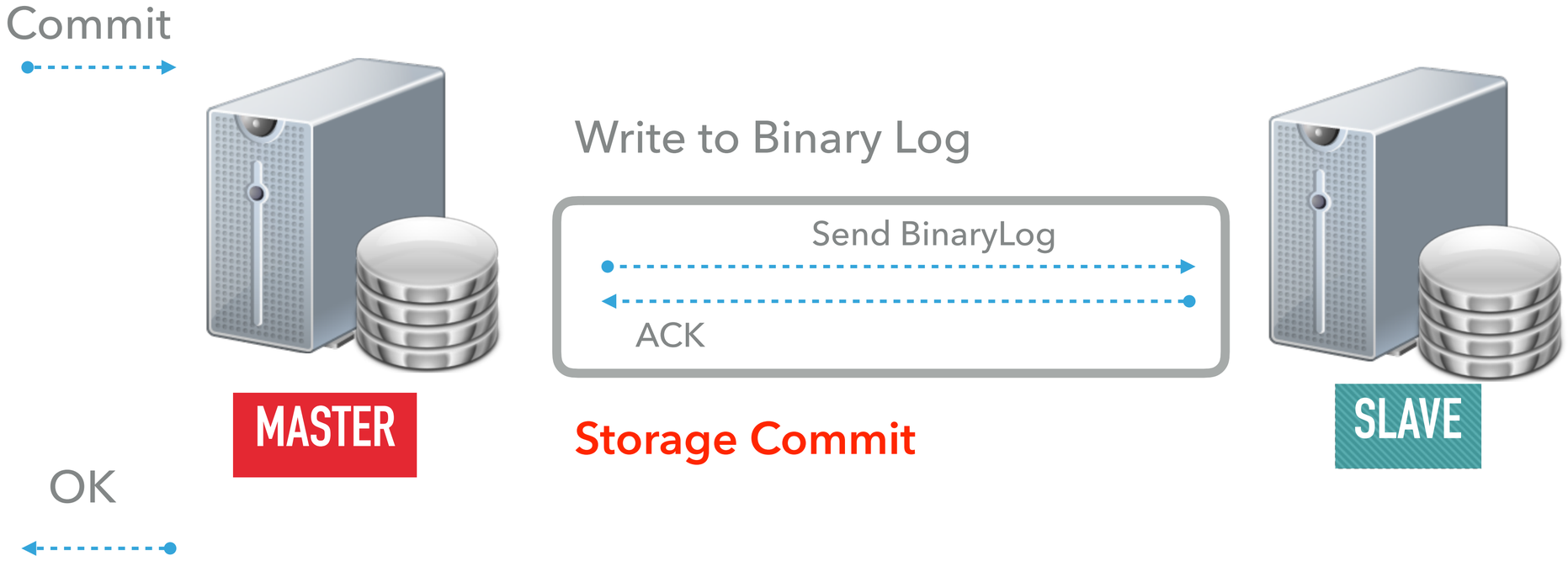
MySQL Replication Data Flow



2. Stability with MySQL

DEVIEW 2017

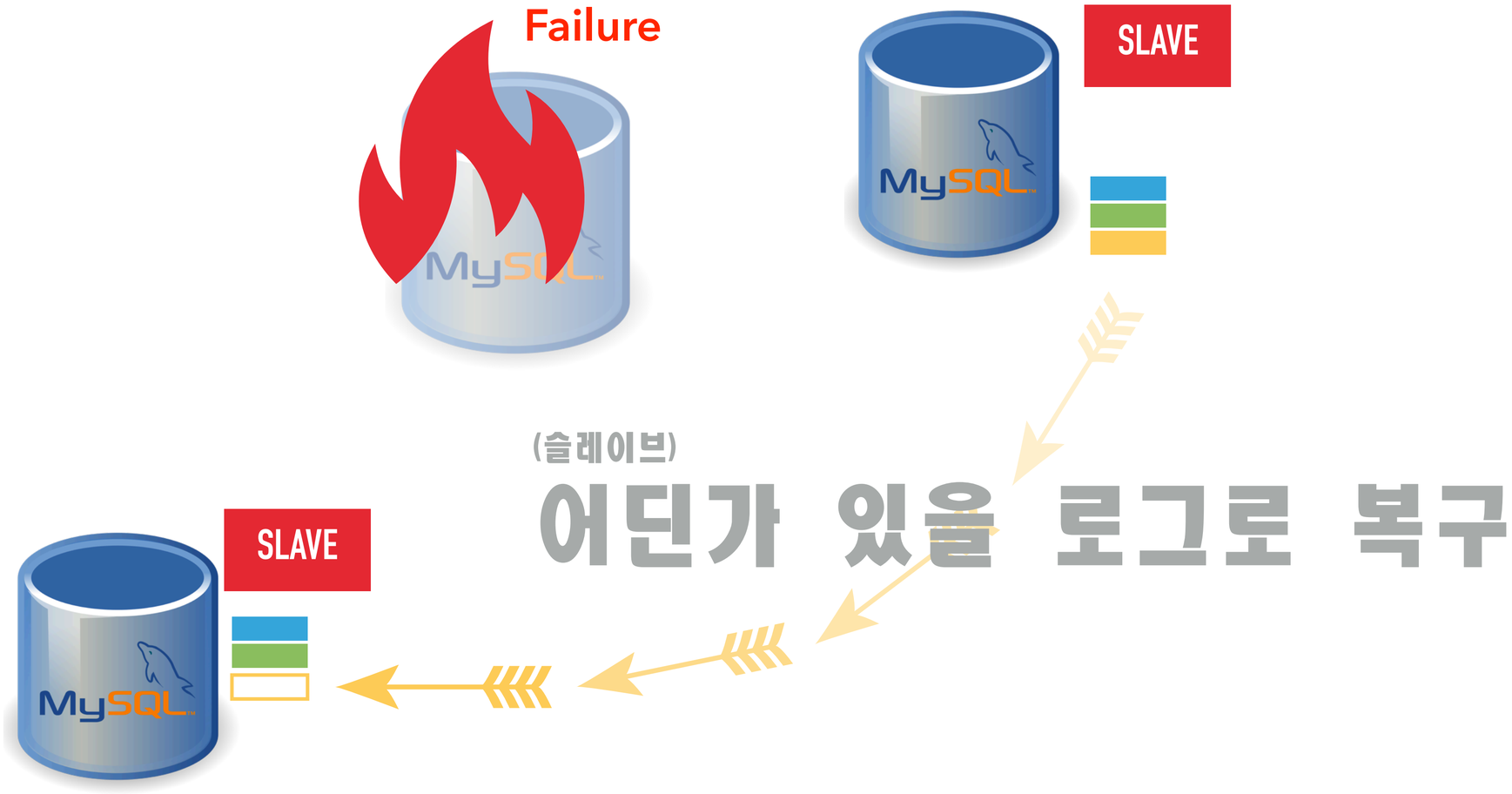
Lossless(AFTER_SYNC) Replication



스토리지 엔진 커밋 “전” 로그 전송

2. Stability with MySQL

Lossless(AFTER_SYNC) Replication



2. Stability with MySQL

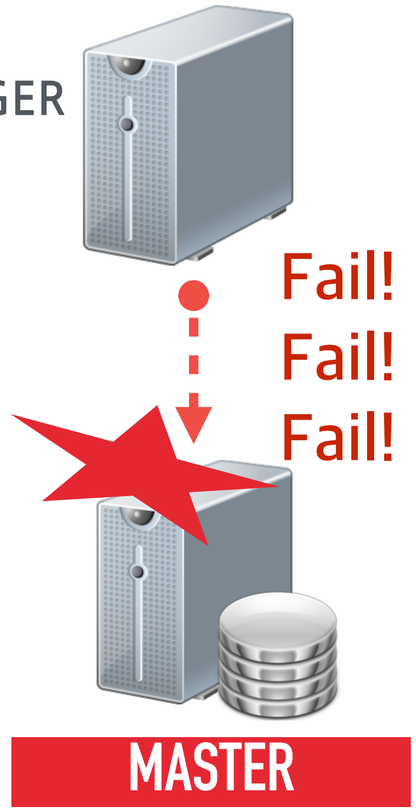
MHA

30초 이내로 데이터 유실없이 복구한다.

(**M**aster **H**igh **A**vailability)

2. Stability with MySQL

Q. MHA 헬스 체크 및 가용성 시나리오

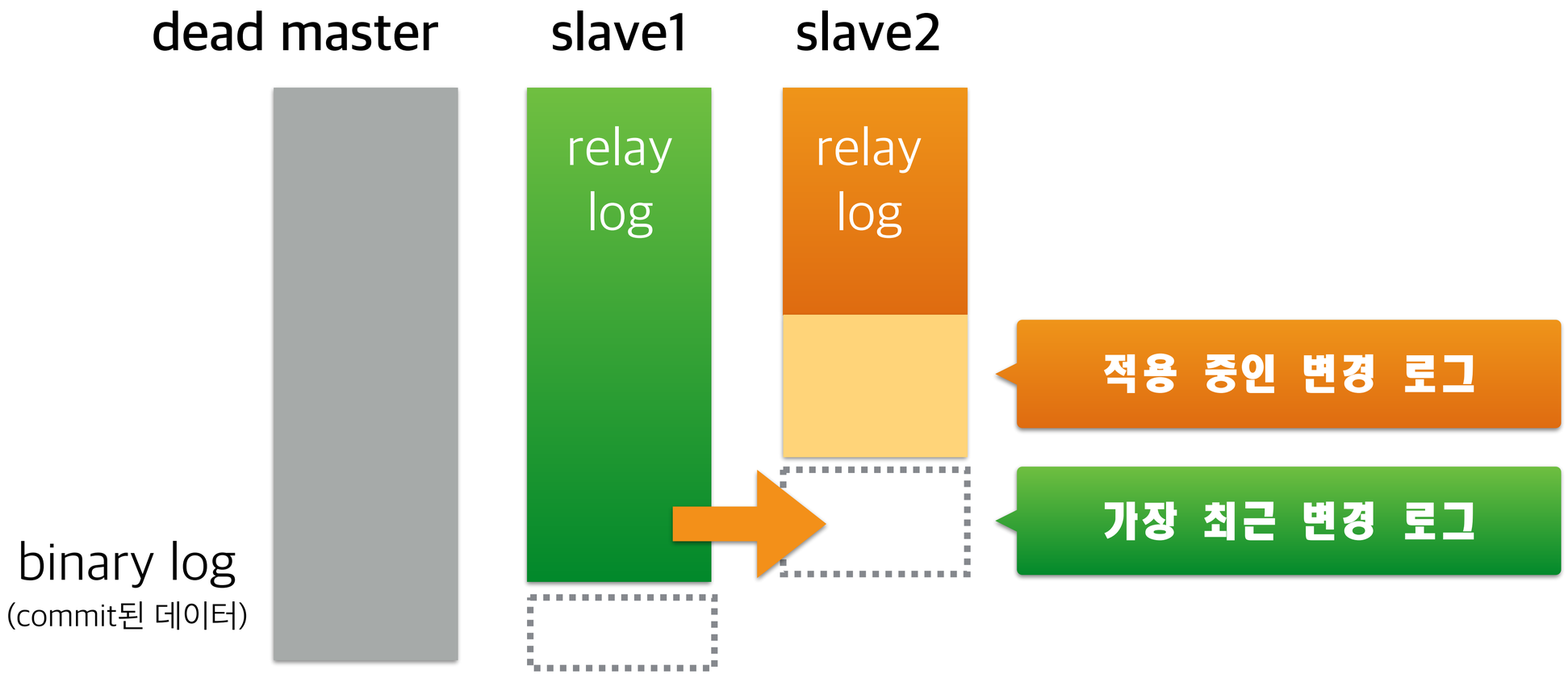


1. 3초마다 마스터 DB 헬스 체크
   CONNECT / SELECT / INSERT
2. 헬스 체크 3회 실패 시, 장애 인지
3. 페일오버 시작

2. Stability with MySQL

DEVIEW 2017

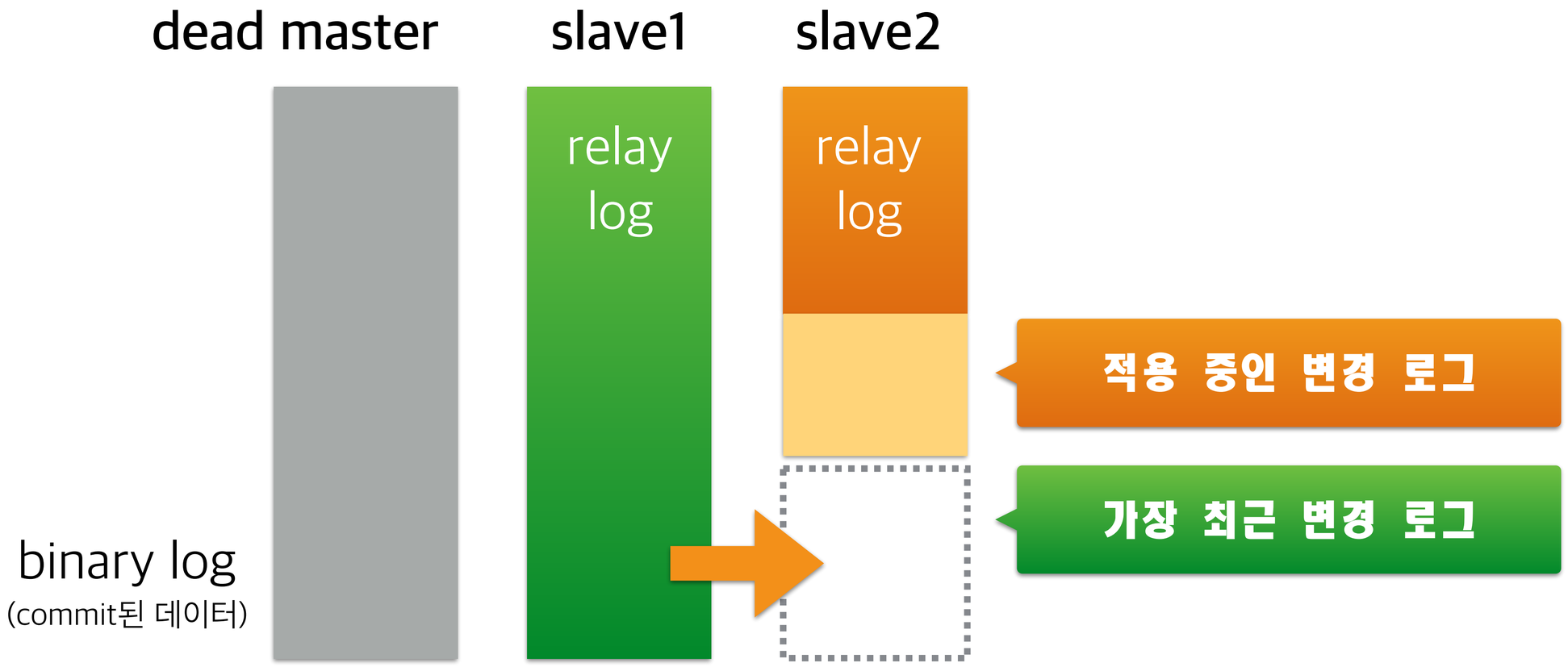
Q. MHA 헬스 체크 및 가용성 시나리오(async)



2. Stability with MySQL

DEVIEW 2017

Q. MHA 헬스 체크 및 가용성 시나리오(lossless)



2. Stability with MySQL

DEVIEW 2017

Q. 비동기 복제이지만, 유실은 없게..

lossless replication은 **relay log를 보장**하고,

MHA는 **relay log로 데이터 일관성**을 맞춘다면?

lossless replication

+

MHA

=

한달

31

내내 데이터 유실 없음

2. Stability with MySQL

DEVIEW 2017

Q. 하나 더 얻은 값진 효과!

디스크 의존성 탈피

2. Stability with MySQL

DEVIEW 2017

Q. 더욱더 견고한 서비스 구성을 위한 고민

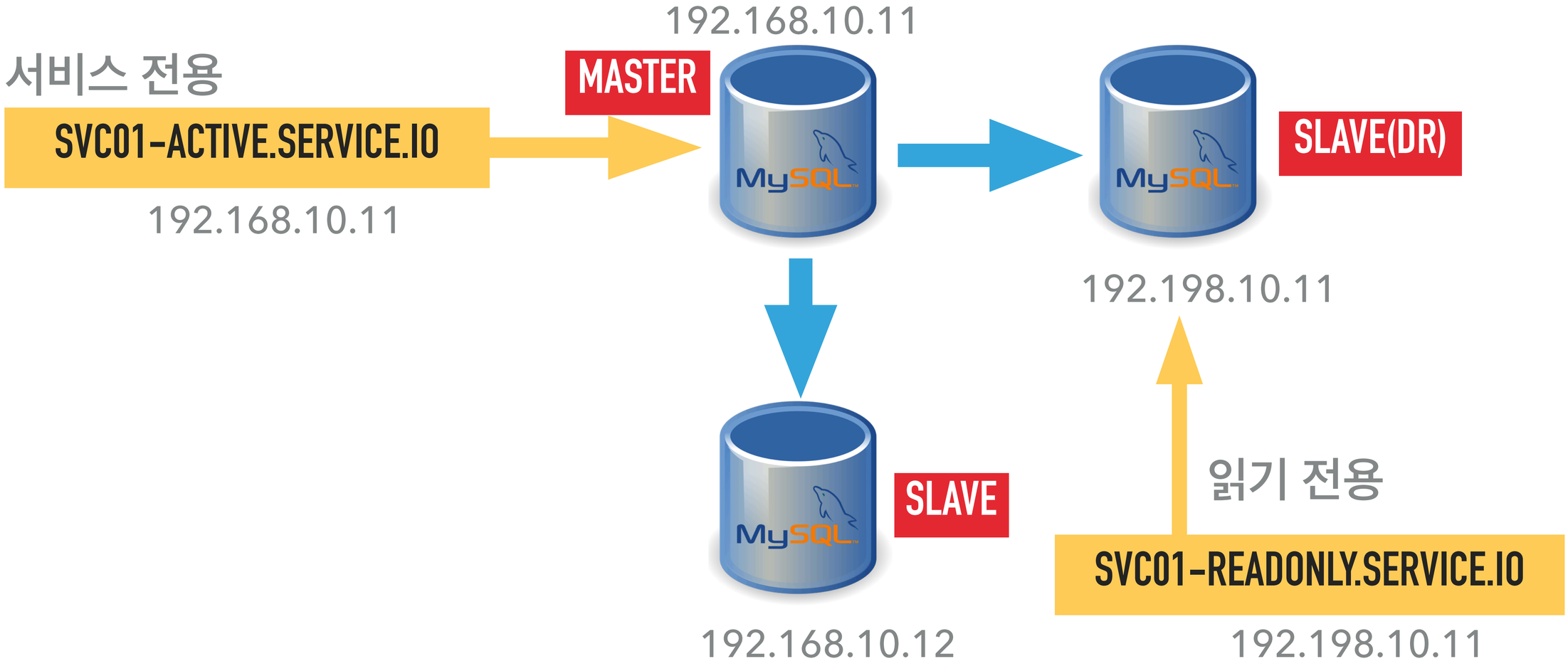
VIP Failover

✓ **Domain Failover**

2. Stability with MySQL

DEVIEW 2017

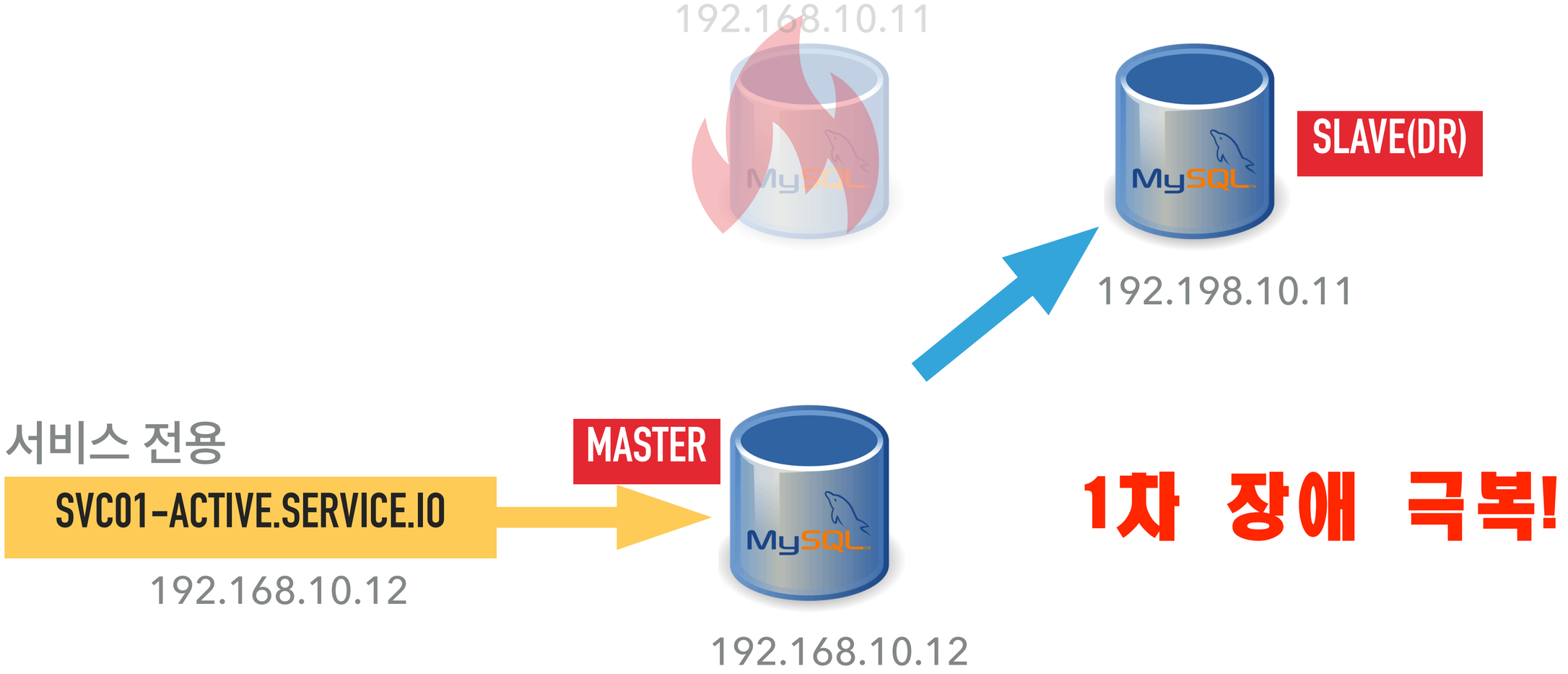
Q. 더욱더 견고한 서비스 구성을 위한 고민



2. Stability with MySQL

DEVIEW 2017

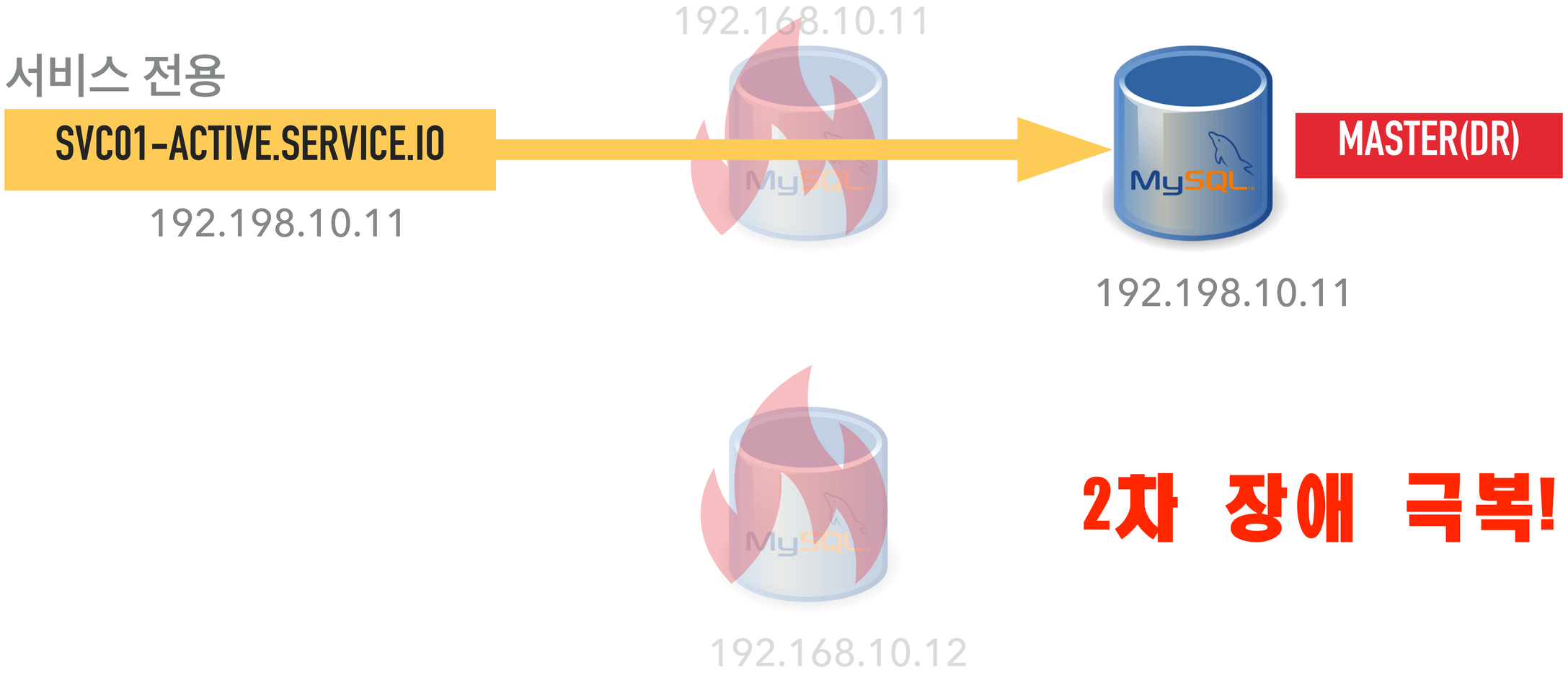
Q. 더욱더 견고한 서비스 구성을 위한 고민



2. Stability with MySQL

DEVIEW 2017

Q. 더욱더 견고한 서비스 구성을 위한 고민



2차 장애 극복!

2. Stability with MySQL

DEVIEW 2017

Q. 주의할 점!

하나. 서비스 도메인의 **TTL은 0으로** 설정합니다.

둘. **DNS 캐싱을 하지않도록**, 자바 구동 옵션을 줍니다.

-Dsun.net.inetaddr.ttl=0

셋. 서비스는 되도록이면 **커넥션 풀**로 동작해야 합니다.

http://gywn.net/2017/01/install_powerdns_with_mysql_backend/

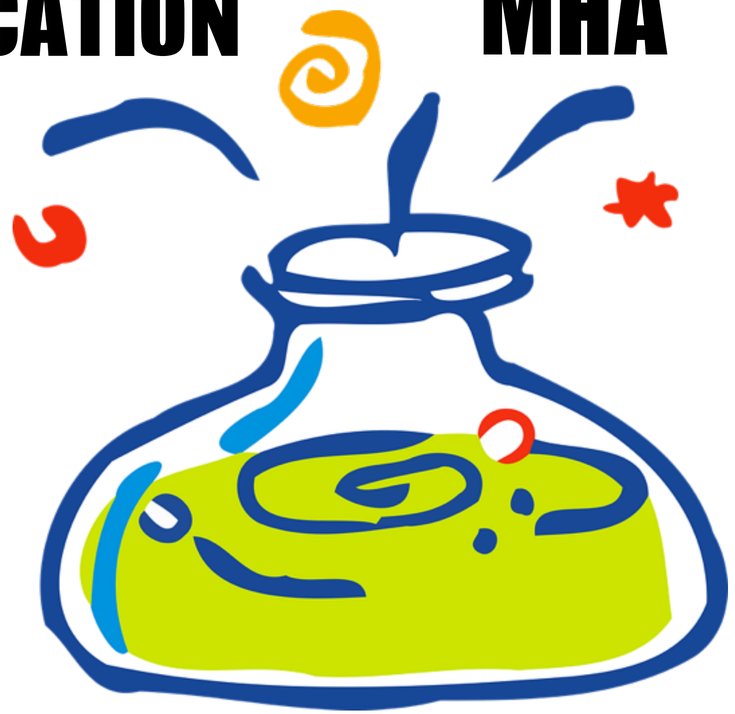
(powerDNS를 내부 정책에 따라 도입을 못했으나, 초반에 이런 식의 활용도 고민했습니다)

2. Stability with MySQL

DEVIEW 2017

(LOSSLESS)
REPLICATION

MHA



lossless replication은

relay log를 보장

MHA는 relay log을 활용하여

데이터 무손실을 보장

도메인 페일오버로 장애 대응함으로써

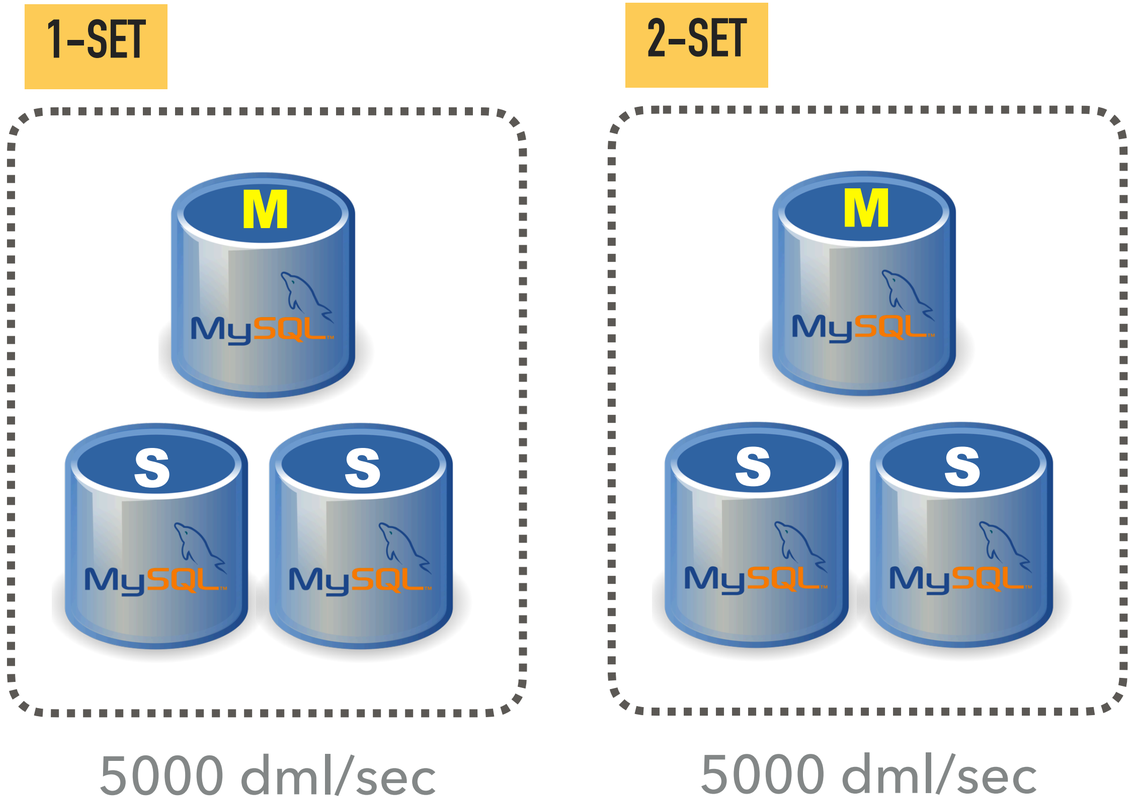
2차 장애도 유연하게 대응

3.
Scale-Out

DEVIEW
2017

3. Scale-Out

Q. 스케일아웃을 왜 해야해요?



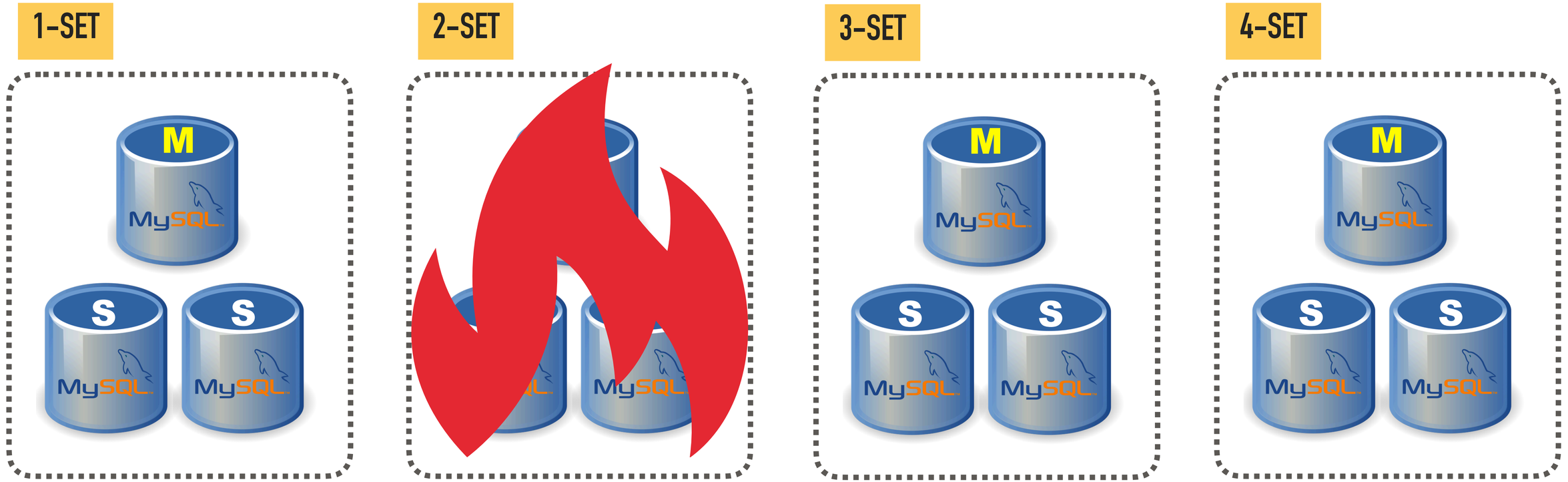
5000 dml/sec

5000 dml/sec

A. 서비스가 커지니까요.

3. Scale-Out

Q. 스케일아웃을 왜 해야해요?



A. 장애 여파도 줄일 수 있고요.

3. Scale-Out

Q. 확장의 기준은 무엇인가요?

STATIC

거의 변경 없음

DYNAMIC HISTORICAL

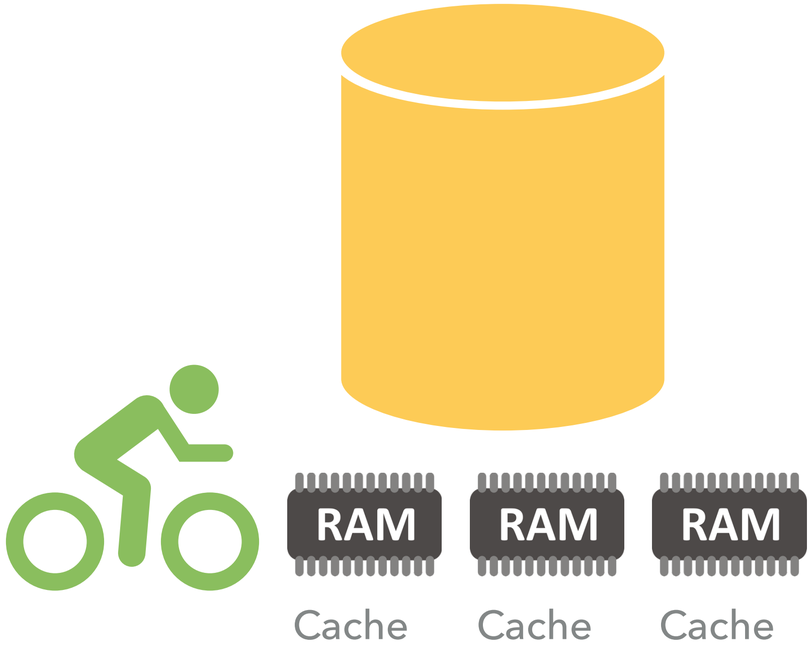
사용자가 생성하는
변경이 잦거나
로그 성격 데이터

3. Scale-Out

DEVIEW 2017

Q. 확장의 기준은 무엇인가요?

STATIC



- 캐시가 느리면, DB가 대처
- DB가 흔들리면, 캐시가 대응

DYNAMIC HISTORICAL

사용자가 생성하는
변경이 잦거나
로그 성격 데이터

3. Scale-Out

DEVIEW 2017

Q. 확장의 기준은 무엇인가요?

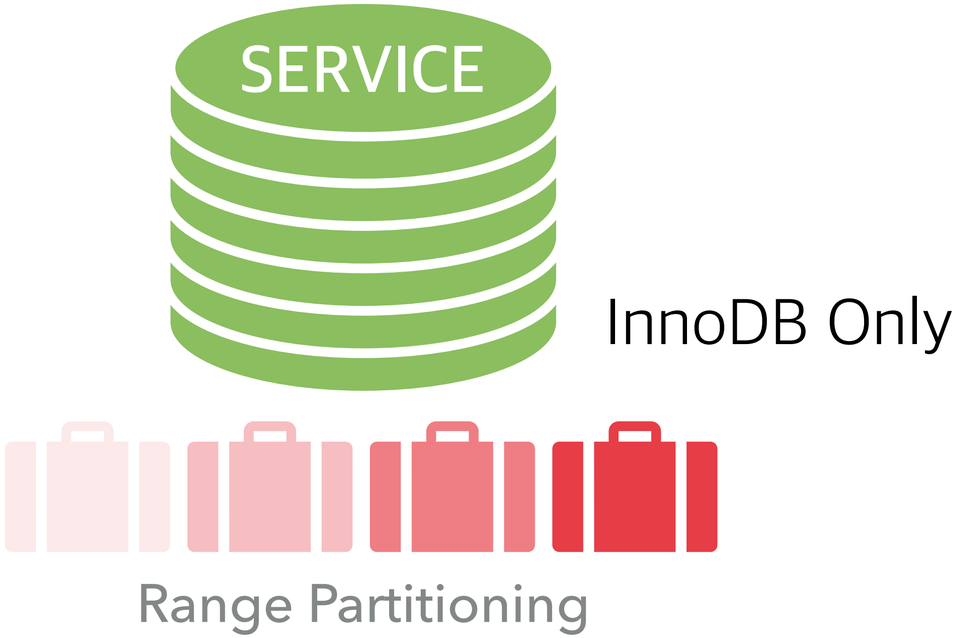
STATIC

거의 변경 없음

DYNAMIC HISTORICAL

3. Scale-Out

스토리지 엔진을 혼용하여 서버 분리에



3. Scale-Out

DEVIEW 2017

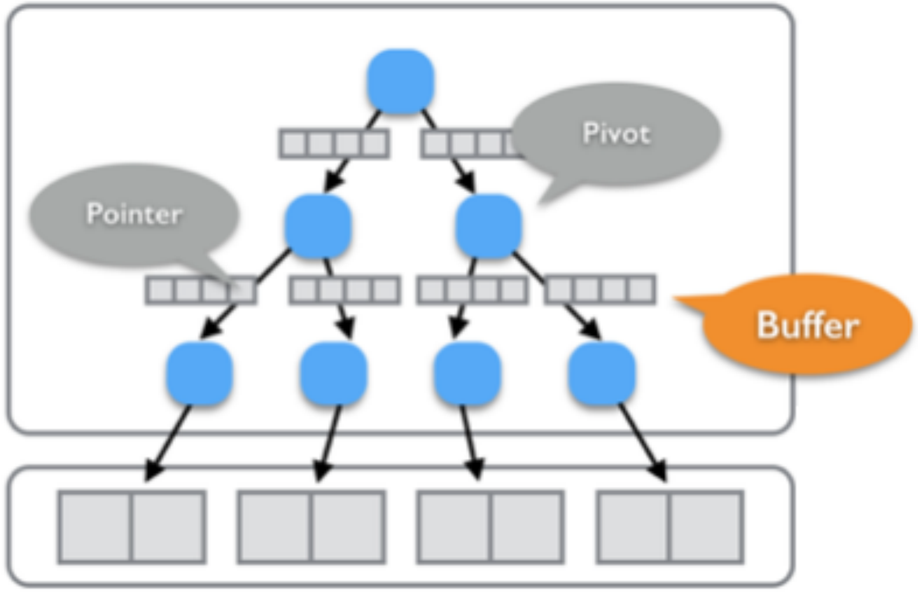
스토리지 엔진을 혼용하여 서버 분리예

3. Scale-Out

DEVIEW 2017

스토리지 엔진을 혼용하여 서버 분리에

TokuDB?



(http://gywn.net/2014/05/fractal-index-in-tokudb/)

Fractal Index

- 피벗에 버퍼를 두어서, IOPS를 줄이자.

Percona MySQL에 포함됨

기존 대비 1/8 수준으로 데이터 감소

로그 적재에 적합함

3. Scale-Out

Q. 확장의 기준은 무엇인가요?

A. 데이터 성격을 보고 결정합니다.

4.
As an open source dba

DEVIEW
2017

4. As an open source dba

DEVIEW 2017

Q. 금융에서 오픈소스로 데이터 해보니 어때요?

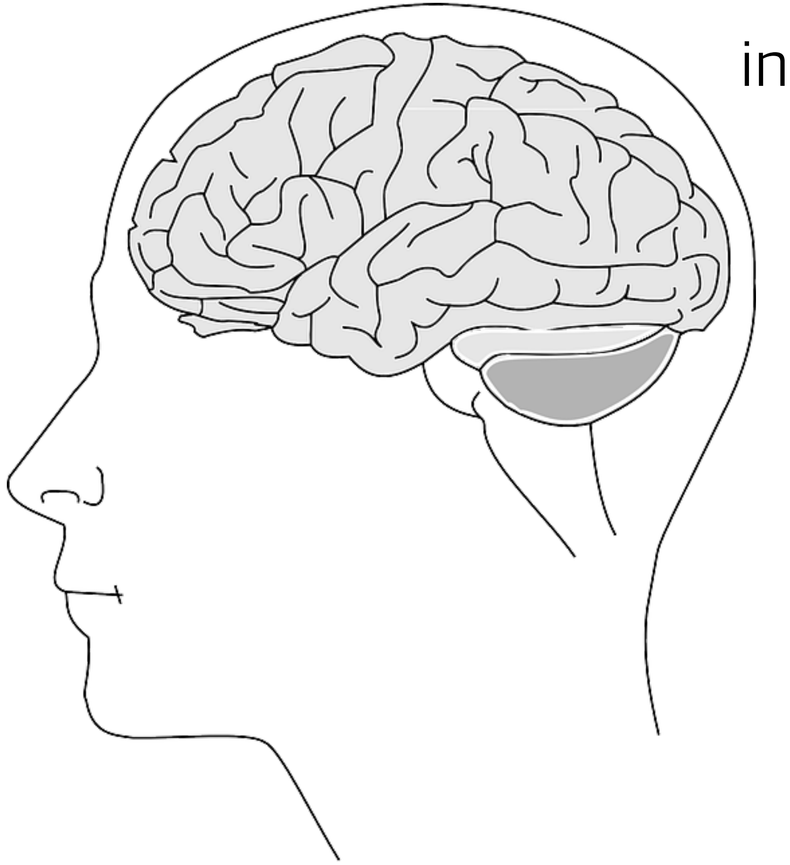
우리는 금융 어벤져스

(진심으로 섞인 것이 “기적”인 다문화 가정)

4. As an open source dba

DEVIEW 2017

Q. 금융에서 오픈소스로 데이터 해보니 어때요?



initialize

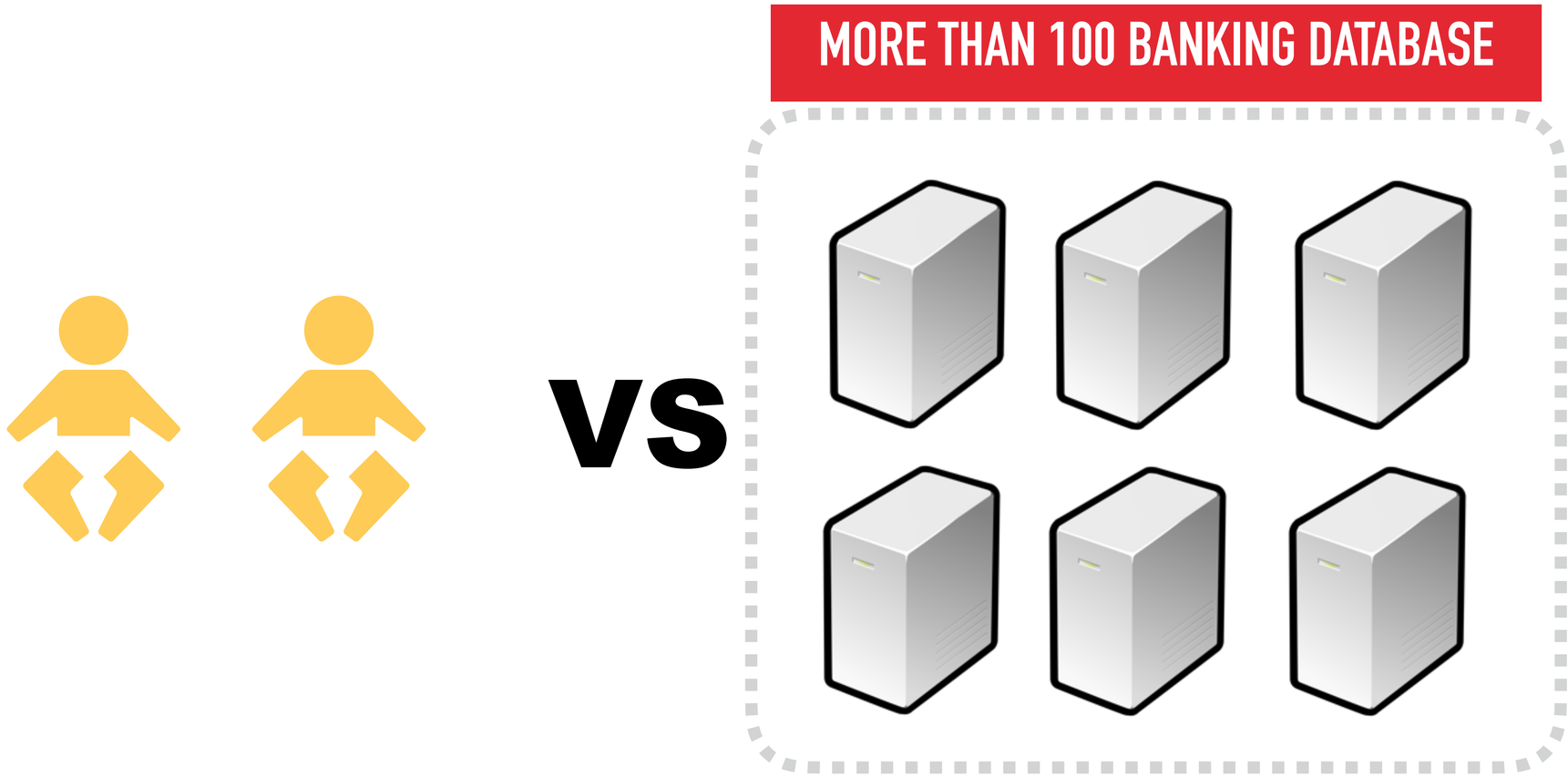
open source

기술의 어려움 보다는
금융 레퍼런스 부재로 인한 신뢰 확보와
상용 솔루션만 고집한 문화와
때로는 공감할 수 없는 환경들이 어려워요.

4. As an open source dba

DEVIEW 2017

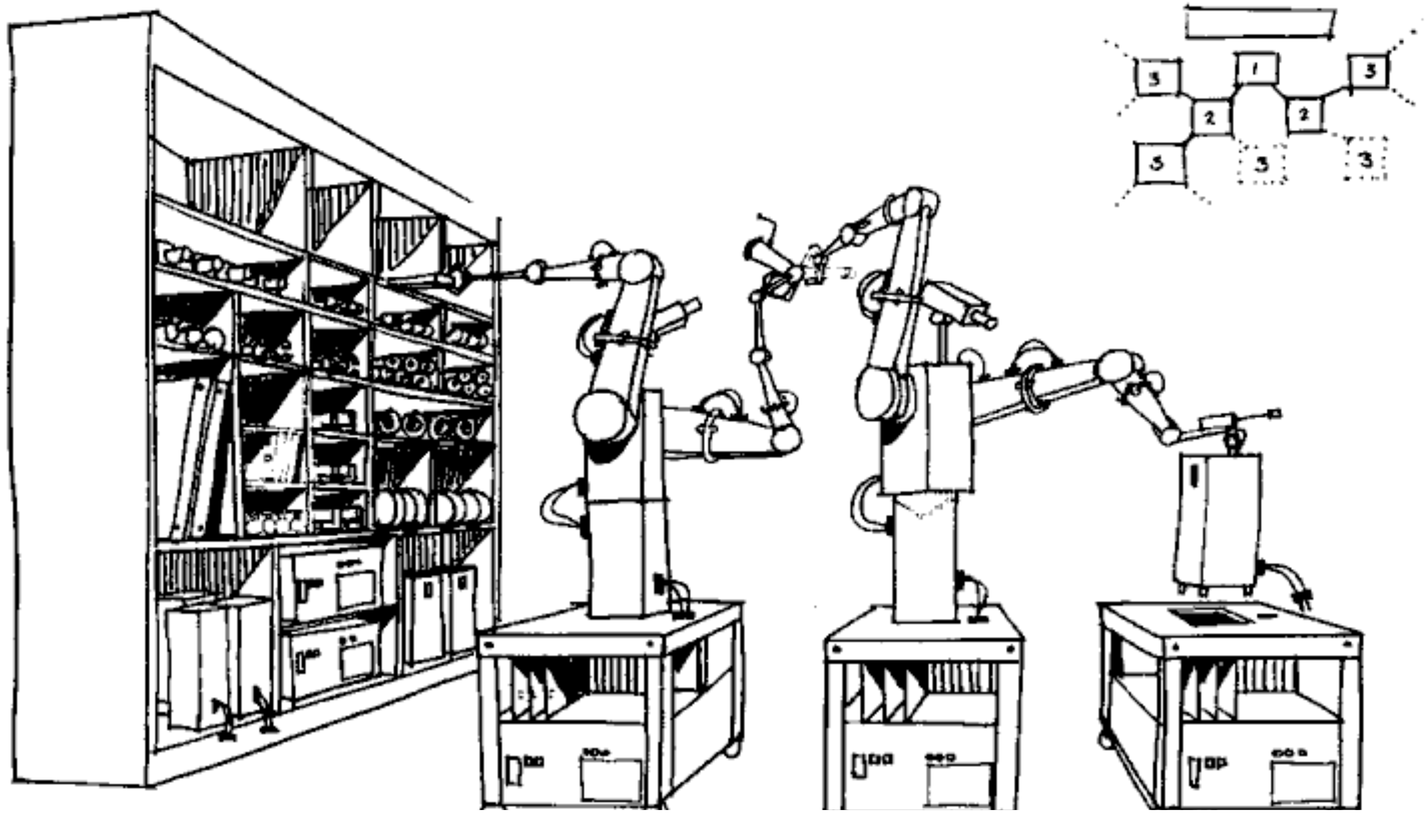
Q. 금융에서 오픈소스로 데이터 해보니 어때요?



4. As an open source dba

DEVIEW 2017

Q. 금융에서 오픈소스로 데이터 해보니 어때요?



기계에게 95% 단순 운영 업무를..

4. As an open source dba

DEVIEW 2017

Q. 금융에서 오픈소스로 데이터 해보니 어때요?

뿐만 아니라..

4. As an open source dba

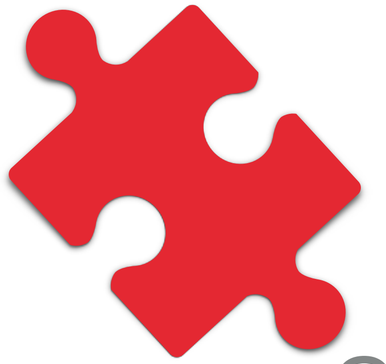
DEVIEW 2017

Q. 금융에서 오픈소스로 데이터 해보니 어때요?

4. As an open source dba

DEVIEW 2017

Q. 금융에서 오픈소스로 데이터 해보니 어때요?



“오픈된 지식”을 잘 조립해서, 최상의 서비스 완성

A. 한마디로.. 짜릿합니다.

그러다 장애나면 “여러사람 다침”

final.
정리하자면..

DEVIEW
2017

final.

DEVIEW 2017

하나. MYSQL은 은행에서 중요한 역할을 담당합니다.

둘. 안정적인 데이터 서비스를 하기 위해

- Lossless Replication
- MHA
- Domain Failover

를 잘 조합해서 씁니다.

final.

DEVIEW 2017

셋. 서비스 무한 확장을 대비해서

- Static? Dynamic? Historical?

  데이터 특성을 보고, 샤딩 구성하였으며,

넷. 제한된 인력과 예기치못한 인재 방지를 위해

- 95% 자동화 구성하고, 서비스 튜닝 포인트를 찾습니다.

final goal.

DEVIEW 2017

오픈소스는
레퍼런스가
전무하였고

(그나마 우린 자유로운 편)

금융권의
폐쇄적인
정책덕에

많은 부분을 직접 만들어야 했습니다.

이것들을 오픈해서, “레퍼런스”가 되어보자

Appendix.

DEVIEW 2017

은행 오픈하기까지 내게 있었던 일들..(외부 오픈한 것들만)

버그 픽스

https://bugs.launchpad.net/mysql-server/+bug/1660591
https://bugs.launchpad.net/percona-server/+bug/1679025
https://bugs.launchpad.net/percona-server/+bug/1657941
https://bugs.launchpad.net/percona-toolkit/+bug/1646713

기술 공유

https://brunch.co.kr/@chan/10
https://brunch.co.kr/@chan/11
http://gywn.net/2017/08/small-talk-pt-osc/
http://gywn.net/2017/08/mysql-slow-log-gather/
http://gywn.net/2017/06/mysql-os-cache-management/
http://gywn.net/2017/06/mysql-slave-addition-effect/
http://gywn.net/2017/06/mysql_57-ngram-ft-bug-and-learn/
http://gywn.net/2017/04/mysql_57-ngram-ft-se/
http://gywn.net/2017/01/automated-recovery-scenarios-for-lazy-mysql-dba/
http://gywn.net/2017/01/how_to_move_partition_data_to_another/
http://gywn.net/2017/01/install_powerdns_with_mysql_backend/

to be continued..

Q & A

(아직은 짤리면 안되니, 민감한 질문은 사양합니다.)


Thank you