



공개 SW의 이해와 트랜드))

시작, 트랜드 그리고 나아갈 길.



1. 오픈소스의 시작점..



3명의 거장.



Richard Stallman



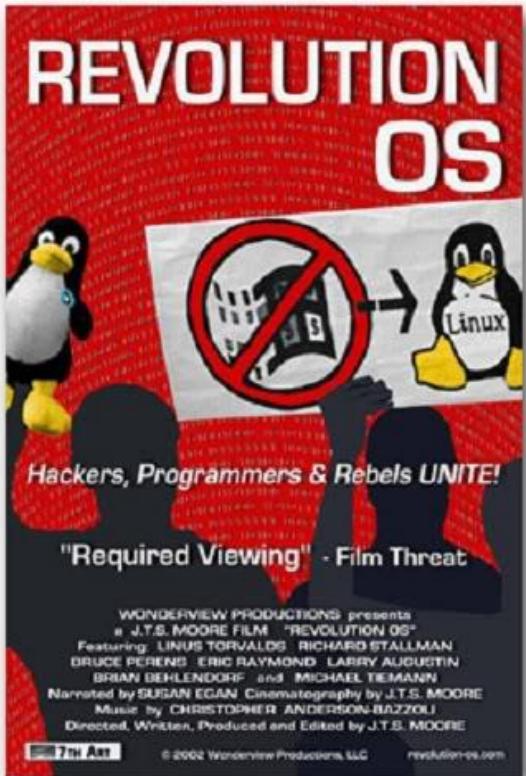
Linus Torvalds



Eric S. Raymond



이 3명과 오픈소스를 잘 이해할수 있는 영화



영문 자막

https://youtu.be/GsHh2wfy_-4

한글 자막 - <http://bit.ly/YDb5TN>



오픈소스

- 오픈소스는 ‘소스코드’를 공개하여 누구나 수정 · 재배포할 수 있는 소프트웨어



H 9003 DYNAMIC TEST SYSTEM

1960 ~ 1970



- 1960 ~ 1970
- 하드웨어 판매가 핵심
- 소프트웨어의 코드는 공유하는 문화
- FORTRAN, COBOL, Algol



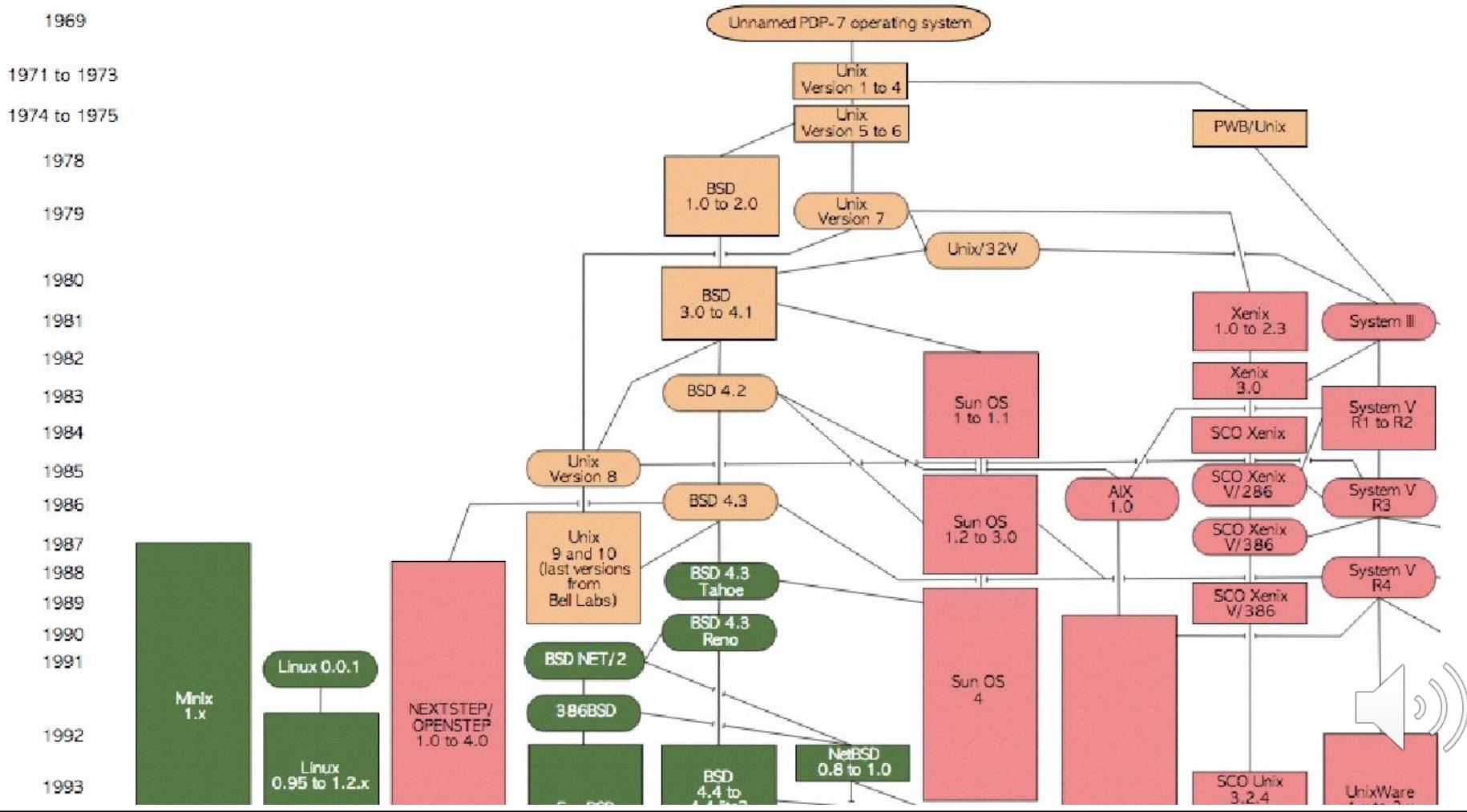


1970 ~ 1980
유닉스(UNIX) 탄생



- 1970 ~ 1980
- 70년대 초반 벨 연구소 직원인 켄 톰슨, 데니스 리치, 더글러스 매클로리 등이 처음 개발
- AT&T를 비롯한 여러 회사들과 버클리대학(UC Berkeley) 등 비영리 단체들이 개발한 다양한 버전





1980 ~ 1990 상용 소프트웨어



- 1980 ~ 1990

- 소프트웨어 저작권에 대한 인지가
시작된 상업용 소프트웨어 시대



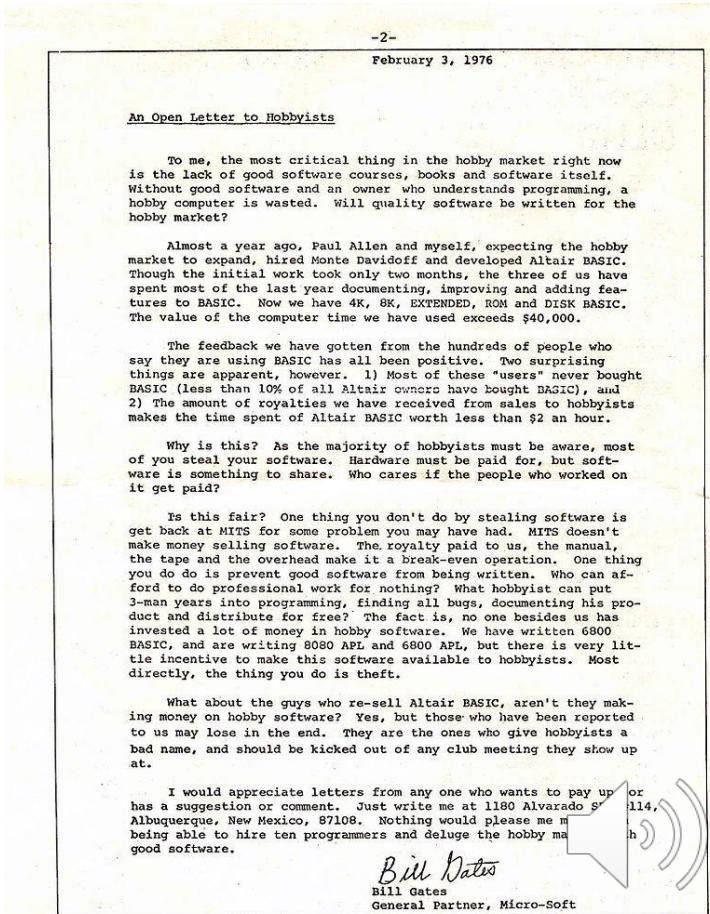
• 1980 ~ 1990

빌 게이츠

컴퓨터 애호가들에게 보내는 공개 편지 “Open Letter to Hobbyists”

저작권의 개념이 대두됨

<http://bit.ly/1DoDJ0o>





1984 ~ 2000

자유소프트웨어의 성행
리차드 스톤먼, GNU 선언



- 1985 - GNU 선언문을 발표
- 유닉스에 대항한 자유로운 대안을 만들기 위한 의지를 말한 것
- Free Software 운동 후
자유 소프트웨어 재단(FSF) 설립



- Free = 무료

- 사용

- 수정

- 재배포

- 에 자유로운 것!





리눅스(Linux) 탄생



- 리눅스(Linux) 1991~
- 리눅스는 핀란드 헬싱키 대학의 대학원생인, 리누스 토발즈가 취미로 개발한 커널
- 현재 200여 종류가 넘는 배포판이 존재





Eric S. Raymond OSI 설립



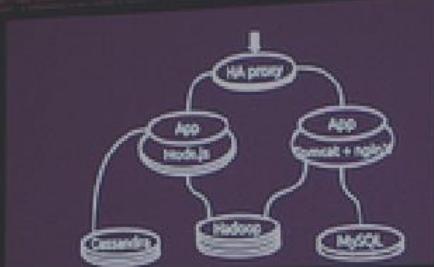
1998년 - OSI(Open Source Initiative)

- 오픈소스 문화를 활성화해 기업들의 참여를 이끄는 목적.
- GPL은 파생물 역시 항상 동일한 라이센스 GPL을 준수해야 하므로, 즉 소스가 외부에 공개되어야 하므로, 기업의 참여가 제한됨
- 오픈소스를 변경하더라도, 다른 라이센스로 또는 상업용으로 쓰는 것을 가능케 함.





오픈소스가 주류로!



현재는 대세!!

- OS (리눅스, OSX- freebsd 기반)
- 브라우저 (크롬, 파이어폭스, 사파리 - Webkit)
- 모바일 OS (안드로이드, 타이젠 등)
- 백엔드 서버 소프트웨어 (웹서버, 데이터베이스등)



2. 라이센스 이야기.





원저작자

사용자

누구에게 더 많은 권한을 줘야할까?



원 저작자의 권한을..

GPL / LGPL ..



고객 (사용자)의 권한을..

MIT / BSD / Apache..



주요 License 비교

	GPL	LGPL	MIT	BSD	Apache
저작권 보호 기능	O	O	O	O	O
상용 SW 사용 가능	O	O	O	O	O
기능 확장 공개 의무	O	O	X	X	X
타인에게 특허권 제공	X	X	X	X	O
독점 프로그램에서 사용 가능 여부	X	O	O	O	O
라이선스 전파 여부	O	O	X	X	

주요 License 비교

	GPL	LGPL	MIT	BSD	Apache
저작권 보호 기능	O	O	O	O	O
상용 SW 사용 가능	O	O	O	O	O
기능 확장 공개 의무	O	O	X	X	X
타인에게 특허권 제공	X	X	X	X	O
독점 프로그램에서 사용 가능 여부	X	O	O	O	O
라이선스 전파 여부	O	O	X	X	

주요 License 특허권 제공여부

	GPL	LGPL	MIT	BSD	Apache
저작권 보호 기능	O	O	O	O	O
상용 SW 사용 가능	O	O	O	O	O
기능 확장 공개 의무	O	O	X	X	X
타인에게 특허권 제공	X	X	X	X	O
독점 프로그램에서 사용 가능 여부	X	O	O	O	O
라이선스 전파 여부	O	O	X	X	

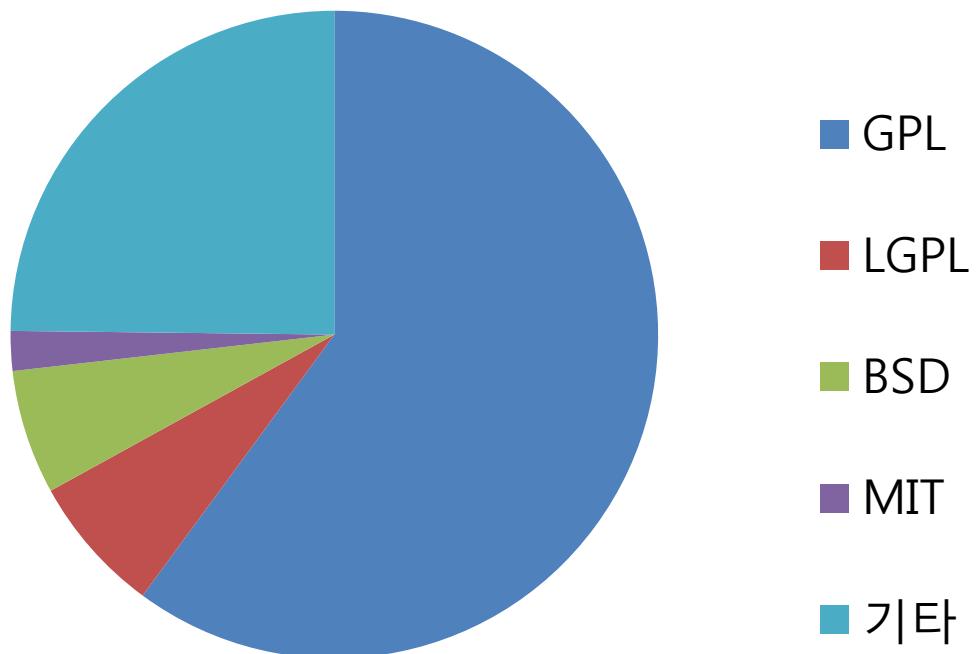
2008년 Open Source License 통계

GPL (60%)

MIT (2%)

BSD (6%)

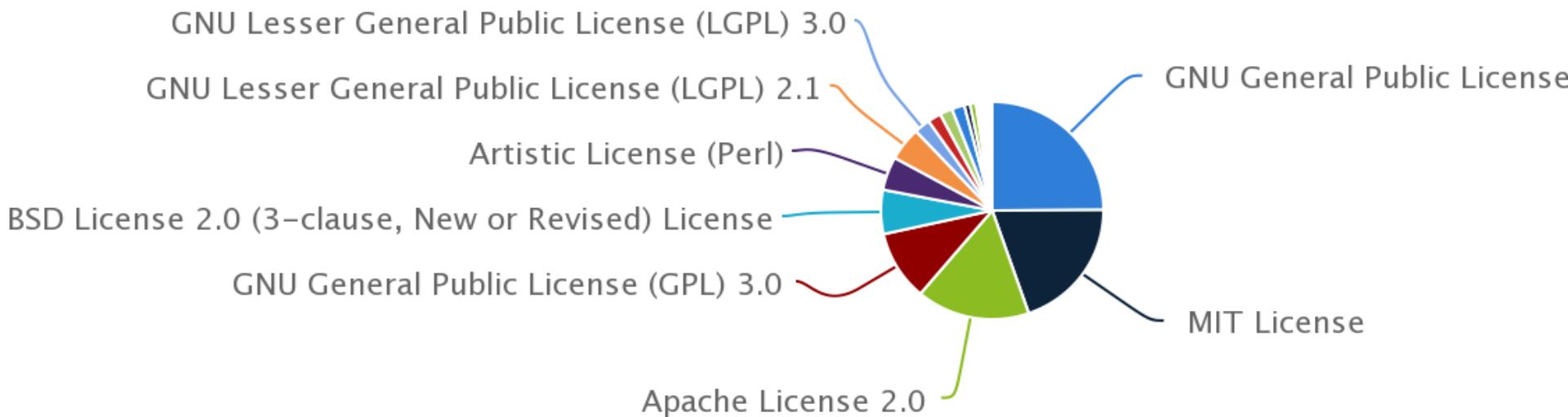
LGPL (7%)



출처 <http://bit.ly/JqfTFz> , 2008년



2015년 Open Source License 통계



왜 이렇게 BSD/MIT가 증가했나?

The screenshot shows the homepage of Code for America. At the top left is the logo 'CODE for AMERICA'. To its right is the tagline 'A NEW KIND OF PUBLIC SERVICE' and a subtitle 'Helping governments work better for everyone with the power of the web.' Below the tagline are social media icons for RSS, Facebook, YouTube, Twitter, and Email. A navigation bar below the tagline includes links for HOME, ABOUT, FELLOWS, PARTNERS, ISSUES, PROJECTS, BLOG, and DONATE. A large green button in the center foreground contains the Korean text '미 정부의 끊임없는 노력!' (The continuous effort of the US government!). The background features a grid of images showing various projects across different cities like Boston, Philadelphia, and Seattle.

A NEW KIND OF PUBLIC SERVICE

Helping governments work better for everyone
with the power of the web.

HOME ■ ABOUT ■ FELLOWS ■ PARTNERS ■ ISSUES ■ PROJECTS ■ BLOG ■ DONATE

BY THE NUMBERS

BOSTON

PHILADELPHIA

SEATTLE

The Boston Project:
Improving High School Education by
Connecting the Community

미 정부의 끊임없는 노력!



플랫폼 싸움!



오픈 소스 라이선스 위반

"오픈소스 라이선스 위반도 저작권 침해"

5일 한국저작권위원회에 따르면, 지난 2월 오픈소스SW 라이선스 위반이 저작권 침해에 해당하는가를 묻는 '제이콥슨 대 카처 사건'의 화해안이 미국 법원에 의해 결정됐다. 법원은 오픈소스SW 라이선스의 의무규정...

[네이버 ▶](#) | 관련기사 보기

오픈소스SW 라이선스 소송 늘듯

특히 미국과 유럽 법조계는 현지 법원이 오픈소스SW 라이선스 위반이 저작권 침해에 해당한다는 취지의 판결을 내리면서부터 이 문제에 본격적인 관심을 갖기 시작한 것으로 알려졌다. 대표적인 사례가 지난해 초

[네이버 ▶](#) | 관련기사 보기

"아이폰·안드로이드의 오픈소스 라이선스 위반"

오픈로직의 조사에 따르면, 이를 앱은 기본적인 라이선스 조건 위반하고 있다. GPL/LGPL 라이선스는 개발자가 소스 코드를 이용할 수 있도록 해야 한다는 조건이 붙어 있으나, 애플과 구글의 라이선스는 기속 여부를...

[관련기사 보기](#)

오픈 소스 라이선스 분쟁



오픈 소스 라이선스 위반 사례

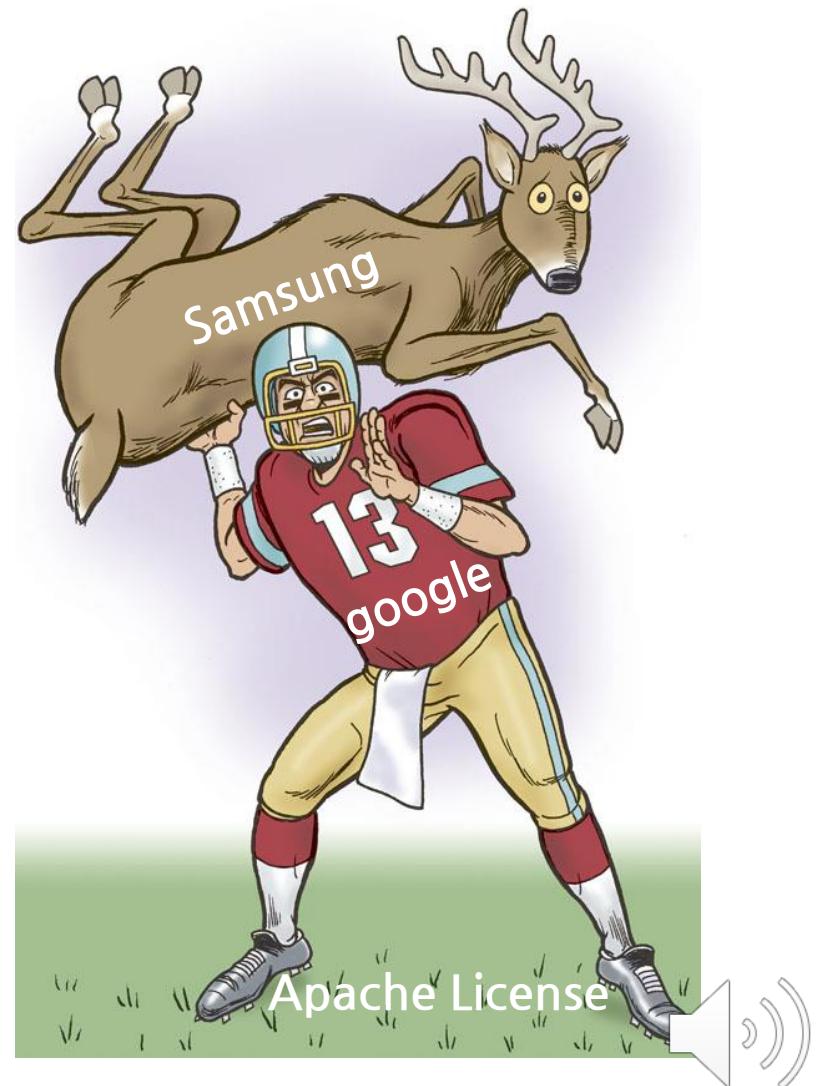
“스카이프, GPL 위반했다”... 노동조합원



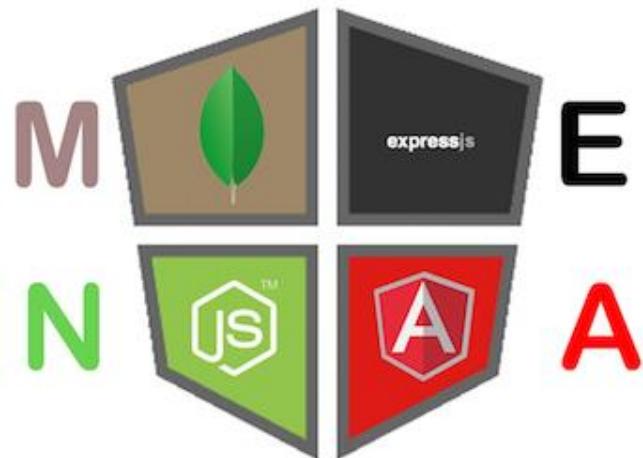
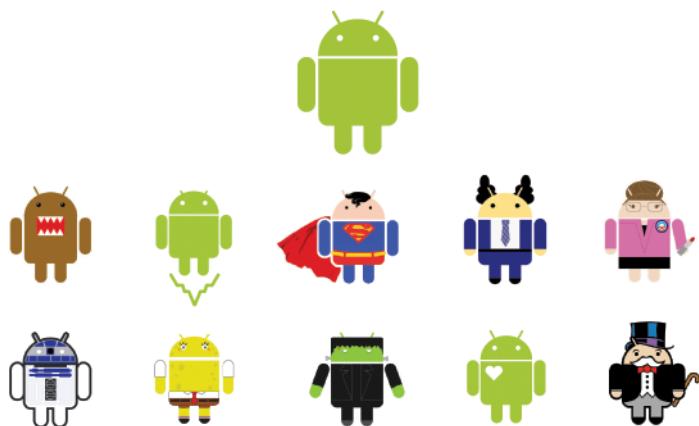
해당 제품이 GPL 기반 제품이라는 것을 명시해야 하는데, 스카이프는 모두 지키지 않았다는 것.



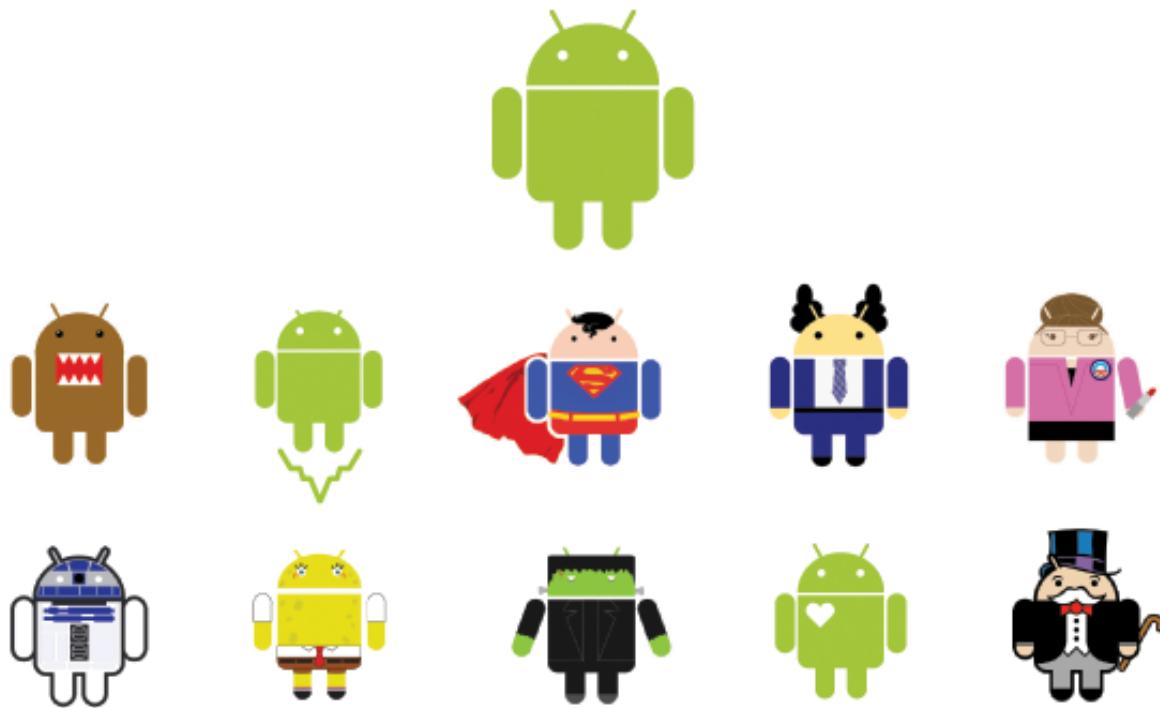
특허 침해 관련



3. 모바일이 가져온 오픈소스 생태계의 변화..



Android 의 문제..



배경.. 극심한 파편화. 엄밀히 말하면..

A	B	C	D	E	F	G	H	I	J	K	???	L
												
Cupcake	Donut	Eclair	Froyo	Gingerbread								
Android 1.5	Android 1.6	Android 2.0/2.1	Android 2.2/2.2.3	Android 2.3/2.3.7								
												
Honeycomb	Ice Cream Sandwich	Jelly Bean	KitKat	Lollipop								
Android 3.0/3.2	Android 4.0/4.0.4	Android 4.1/4.3	Android 4.4	Android 5.0								



엄밀히 말하면.. 3개의 전혀 다른 OS..

A B C D E F G H I J K ??? L



Cupcake

Android 1.5



Donut

Android 1.6



Eclair

Android 2.0/2.1



Froyo

Android 2.2/2.2.3



Gingerbread

Android 2.3/2.3.7



Honeycomb

Android 3.0/3.2



Ice Cream Sandwich

Android 4.0/4.0.4



Jelly Bean

Android 4.1/4.3



KitKat

Android 4.4



Lollipop

Android 5.0

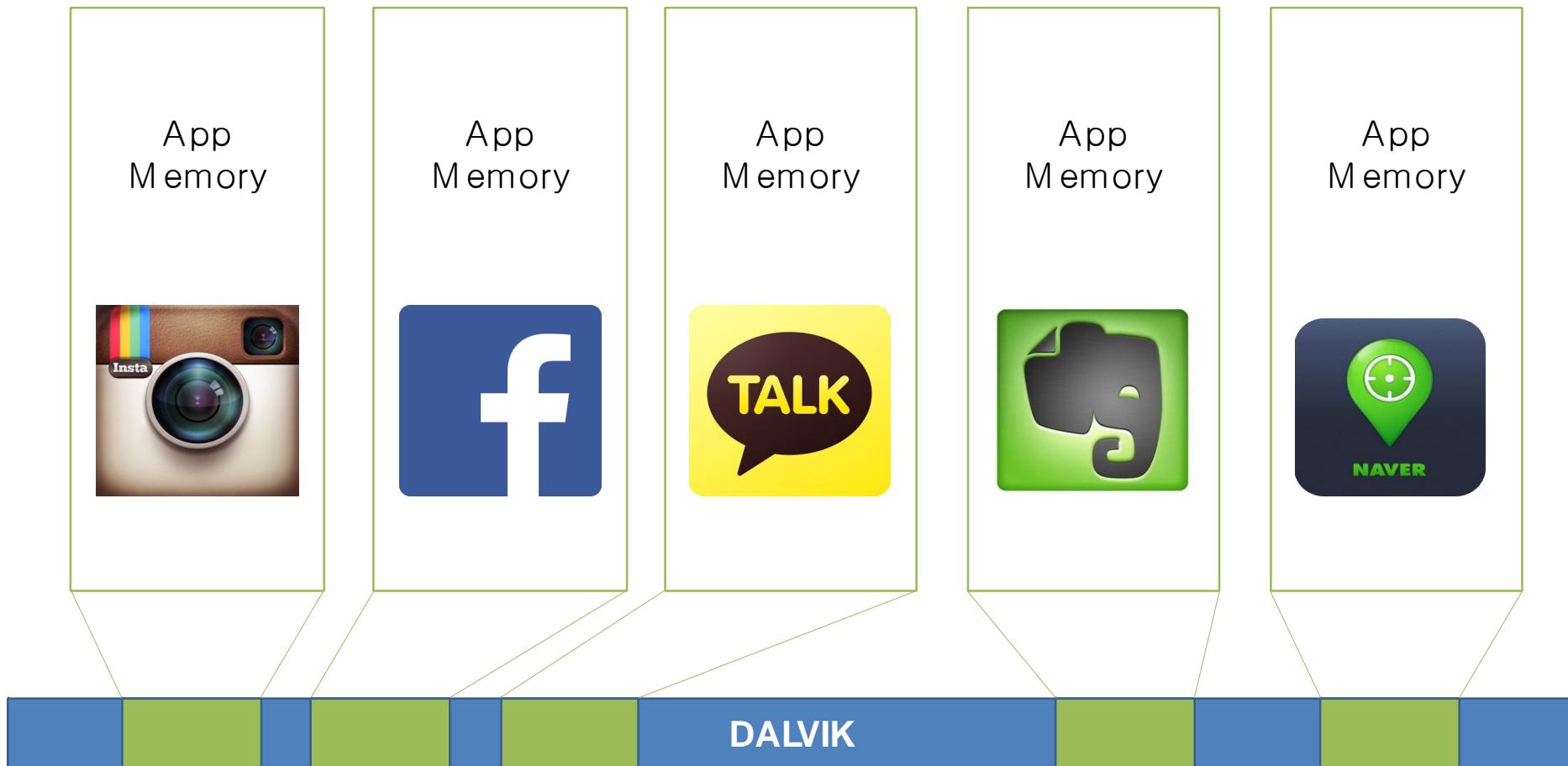


3.0 기준으로 다룬 OS

	3.0 이전	3.0 이후
저장 위치	Dalvik외부 native Heap	Dalvik 내부
GC 동작	전체 App이 멈춤	부분적으로 진행(partially)
Bitmap의 GC	GC의 대상이 아님	GC의 대상
다른 App의 영향	메모리 누수가 발생하면 다른 App도 함께 죽음	다른 앱에 영향이 없음

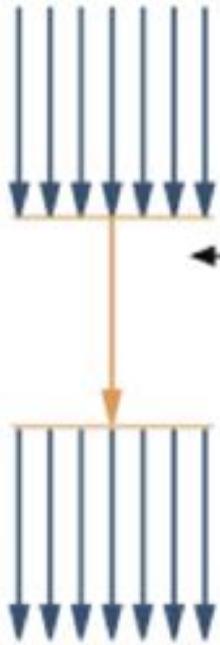


Dalvik is Isolation mode

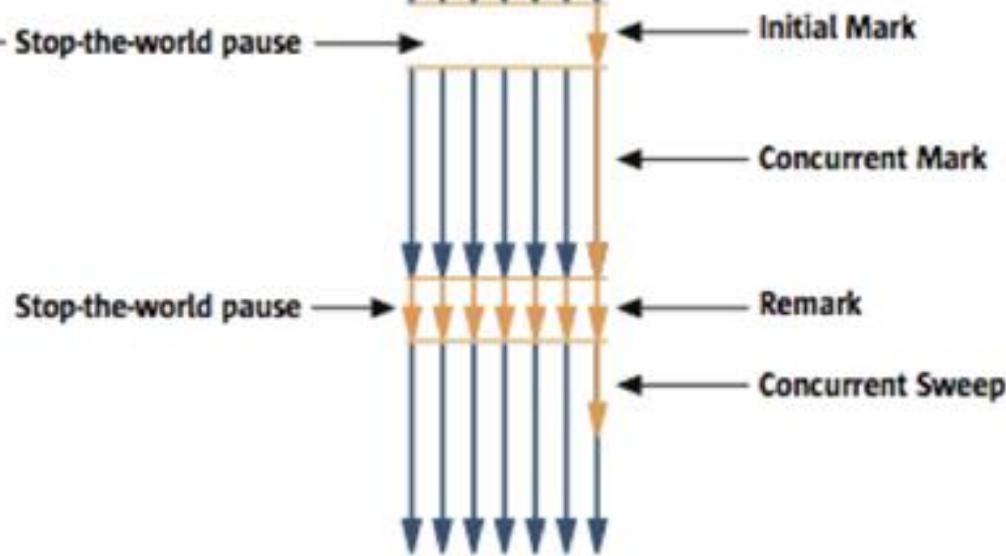


변경된 GC 메커니즘

Serial Mark-Sweep-Compact
Collector



Concurrent Mark-Sweep
Collector



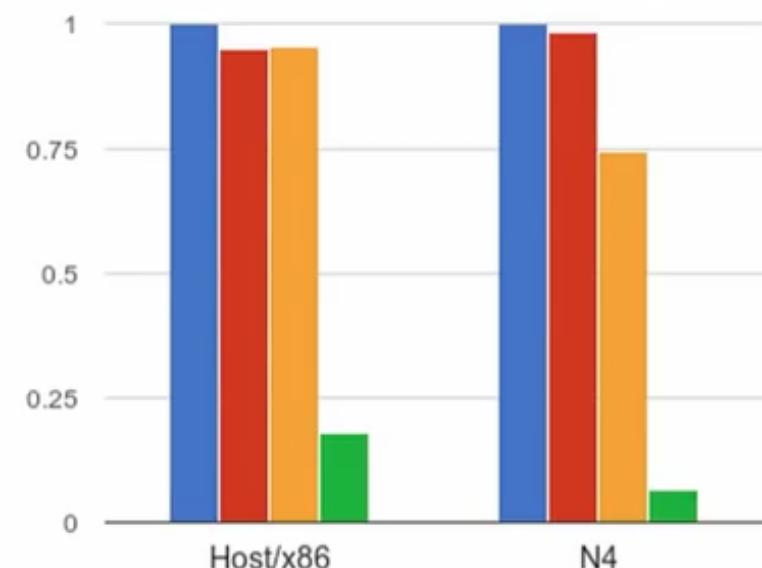
3.0 미만

3.0 이후

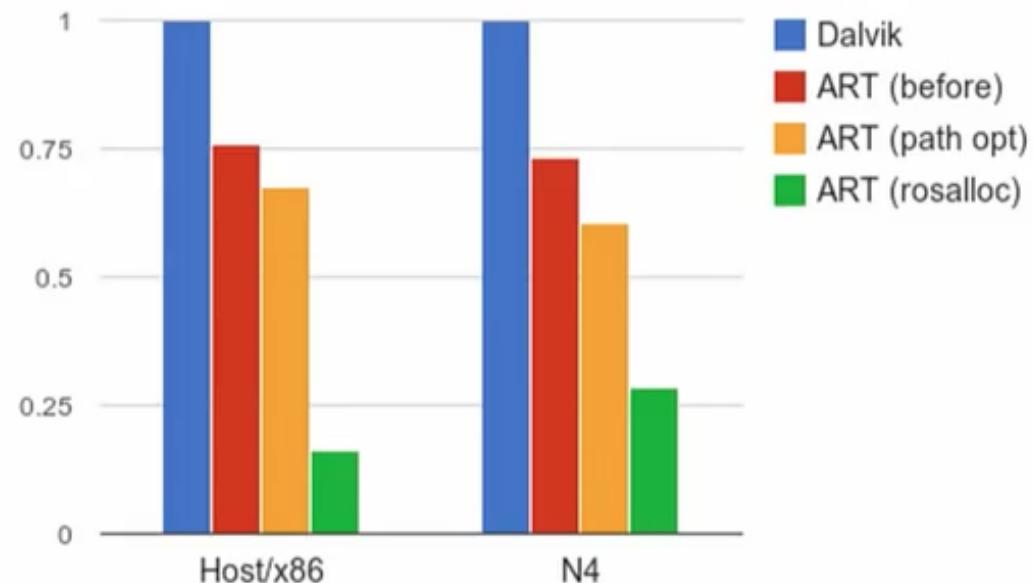


GC의 성능 향상을 위해 새로운 메모리 할당 메커니즘 (Rosalloc)

MemAllocTest (4 Threads)



Sheets MemAllocTest



Rosalloc 이란

Runs fo Slots Allocator

작은 객체를 할당하는 영역과

큰 객체를 할당하는 영역을 구분하자

하나의 영역에 객체를 할당하니,
GC가 빈번하게 발생해서,
역할을 나누자!

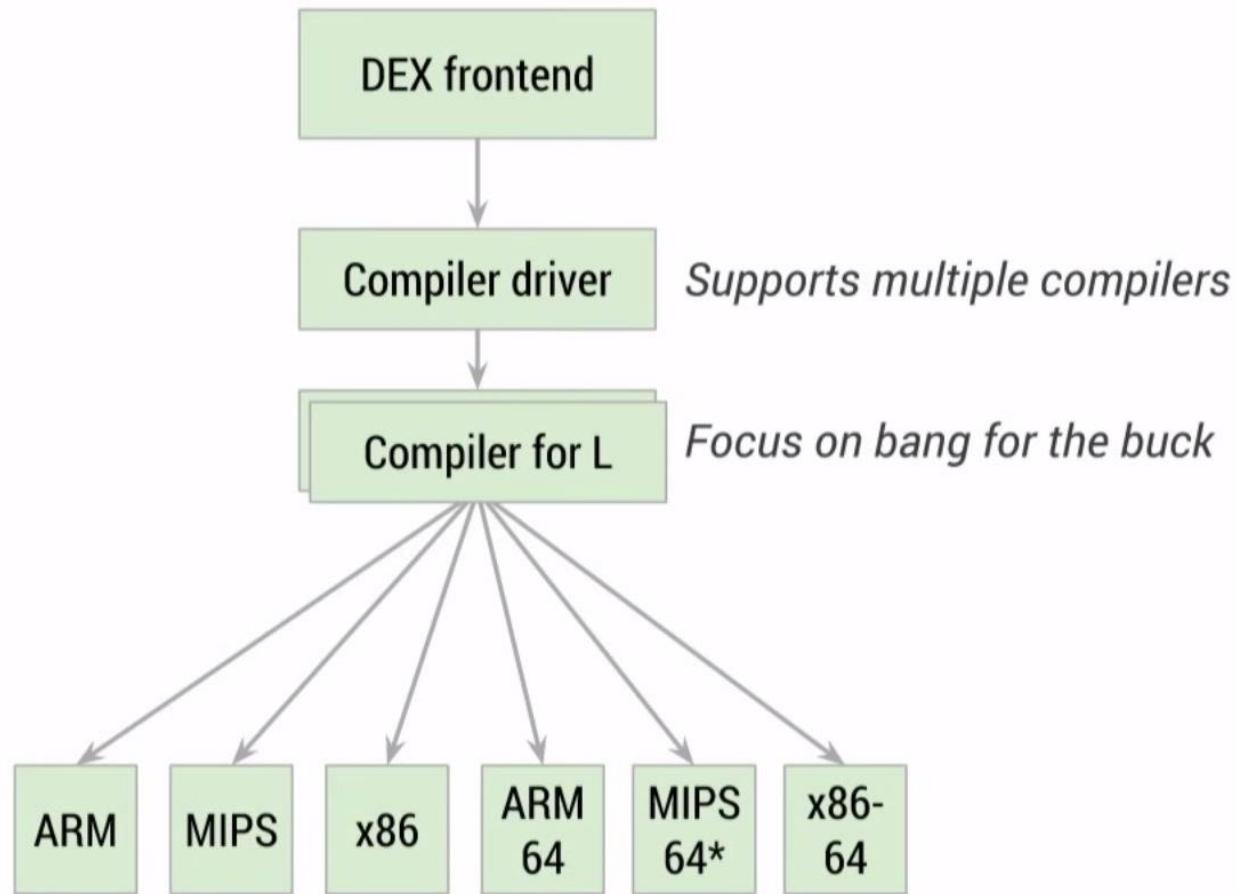
```
// Represents a run of memory slots of the same size
//
// A run's memory layout:
//
// +-----+
// | magic_num      |
// +-----+
// | size_bracket_idx |
// +-----+
// | is_thread_local   |
// +-----+
// | to_be_bulk_freed  |
// +-----+
// | top_slot_idx     |
// +-----+
// |
// | alloc bit map    |
// |
// +-----+
// |
// | bulk free bit map |
// |
// +-----+
// |
// | thread-local free  |
// | bit map            |
// |
// +-----+
// | padding due to alignment |
// | slot 0             |
// +-----+
// | slot 1             |
// +-----+
// | slot 2             |
// +-----+
// ...
// +-----+
// | last slot          |
// +-----+
//
```



Android 5.0 이후 변화된 메모리 구조..



Android ART Runtime (5.0 이후 도입)



피해야할 저주들..

- Gingerbread (2.3)의 저주 - apk 사이즈 제한
- 2.x 이전 / 3.0 이후 메모리 핸들링 기법 다름
- Apache 네트워크 라이브러리의 저주
- 삼성폰의 저주
- 홍대향의 저주



화면사이즈의 Pixel을 얻어오는 함수 API 패턴화...

```
public static Point getSize(Display display) {  
    Point result = new Point();  
    if (Build.VERSION.SDK_INT >= 17) {  
        DisplayMetrics metrics = new DisplayMetrics();  
        display.getRealMetrics(metrics);  
        result.x = metrics.widthPixels;  
        result.y = metrics.heightPixels;  
    } else if (Build.VERSION.SDK_INT >= 14) {  
        result = getRealSize(display);  
    } else if (Build.VERSION.SDK_INT >= 13) {  
        display.getSize(result);  
    } else {  
        result.x = display.getWidth();  
        result.y = display.getHeight();  
    }  
    return result;  
}
```



구글이 만든 이슈는.. 여기서..

<https://code.google.com/p/android/issues/list>



The screenshot shows the 'Issues' tab of the Android Open Source Project's Issue Tracker. The page includes a search bar with fields for 'for' and 'Search', and links for 'Advanced search', 'Search tips', and 'Subscriptions'. Below the search bar is a table listing 29 issues. The columns are: ID, Type, Status, Owner, and Summary + Labels. The issues listed are:

ID	Type	Status	Owner	Summary + Labels
185323	Defect	New	---	MySwitchpreference ,extends SwitchPreference,do not play switch animations
185319	---	New	to...@android.com	Status bar in layout preview does not respect colorPrimaryDark settings
185318	Defect	New	---	Android studio The Win7 location of AndroidStudioProjects and the platform-tools
185317	Defect	New	to...@android.com	TextInputLayout hint already shown before pressing for focus
185316	Enhancement	New	to...@android.com	Request: make importing of Github better, by letting us choose to use latest versions and auto-fix issues
185315	Defect	New	---	Android Wear app 1.3.0 notifications have no content for RemoteViews
185314	---	New	to...@android.com	Error:Could not run build action using Gradle installation 'D:\!STVCap\STVProject\AS\gradle\gradle-2.4'.
185313	Defect	Assigned	alr...@google.com	Android Studio should not create IDE SDKs for Add-ons
185312	---	New	to...@android.com	Clicking "Clear logcat" button frequently hangs
185311	Defect	New	---	Start Failed.... using windows 10
185310	Defect	New	---	Android 5.1.1.1 update crashing.. :(
185309	Defect	New	---	Internal error when starting Android Studio
185308	Defect	New	----	7838082253, +88 Fuck Escort Service In Noida sector 53, 54, 55, 56, Call Girls In Noida Sec 57,
185307	Defect	New	----	7838082253, +89 Fuck Escort Service In Noida sector 53, 54, 55, 56, Call Girls In Noida Sec 57,
185306	Defect	New	----	7838082253, XXX Fuck Escort Service In Noida sector 53, 54, 55, 56, Call Girls In Noida Sec 57,
185305	Defect	New	----	7838082253, GooD Fuck Escort Service In Noida sector 53, 54, 55, 56, Call Girls In Noida Sec 57,
185304	Defect	New	----	7838082253, Red Fuck Escort Service In Noida sector 53, 54, 55, 56, Call Girls In Noida Sec 57,
185303	Defect	New	----	7838082253, Gud Nyt Escort Service In Noida sector 53, 54, 55, 56, Call Girls In Noida Sec 57,
185302	Defect	New	----	7838082253, Blue Nyt Escort Service In Noida sector 53, 54, 55, 56, Call Girls In Noida Sec 57,
185301	Defect	New	to...@android.com	Prevent Studio from auto-switching project window during build
185300	---	New	to...@android.com	Bug: cannot view bitmap of config 565 while debugging
185299	Defect	New	----	Start Failed
185298	---	New	to...@android.com	Application isn't working, showing error

제조사의 파편화

삼성갤럭시 S3의 일부 버전, S4 키캣의 경우 ClipboardManager로 복사붙이기를 시도하면 리부팅됩니다.

저에게 S3는 없지만 S4 키캣은 2번째 시도 때 언제나 일어나는군요.

깜짝 놀랐지만 꽤 유명한 버그로 보입니다.

1. <http://developer.samsung.com/forum/board/thread/view.do...>
2. <http://code.google.com/p/android/issues/detail?id=35732>

이것을 해결하려면? 다음을 추가하면 ㅇㅋ...^^)b

```
if(device != samsung_s3_ics && device != samsung_s4_kitkat)) { ... }
```

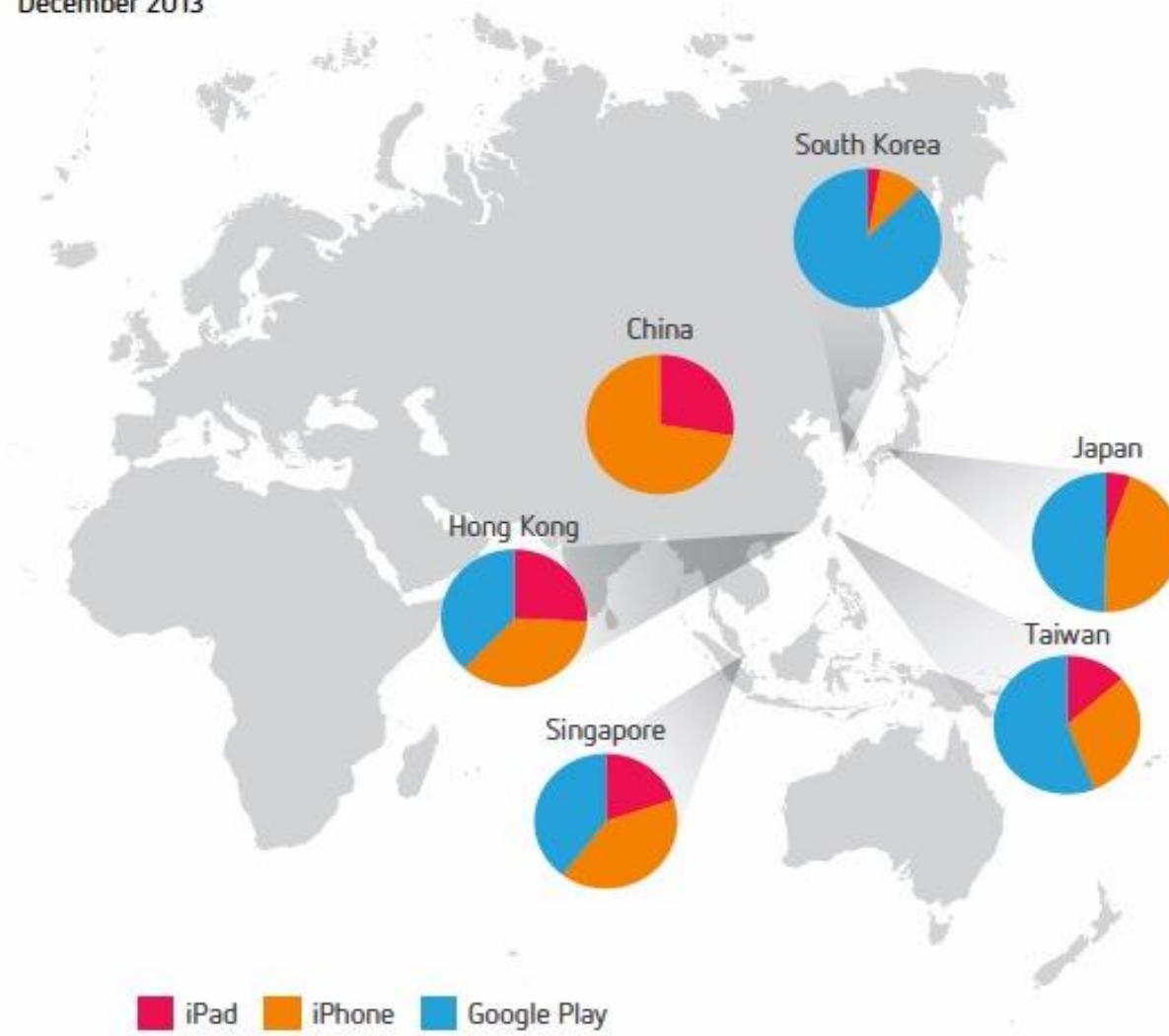
Issue 35732 - android - Clipboard
Crash - Android Open Source
Project - Issue Tracker - Google...

CODE.GOOGLE.COM



지표에 빠져 있는 나라..

Top 6 Countries in Asia split by Apple App Store and Google Play Revenue
December 2013



중국은 구글 Play 사용 못함

Google Play is NOT the place to be in China, ‘App in China’ connects you to Top 20 Chinese Android App Stores

Vallabh Rao | February 14, 2014 at 10:48 am



China is the biggest mobile market in the world in terms of subscribers which no mobile app developer with global ambitions can choose to ignore. [Statistics from November 2013](#) put the total number of mobile subscribers in China at 1.23 billion. Of these, around 403 million mobile users subscribe to 3G mobile technology.

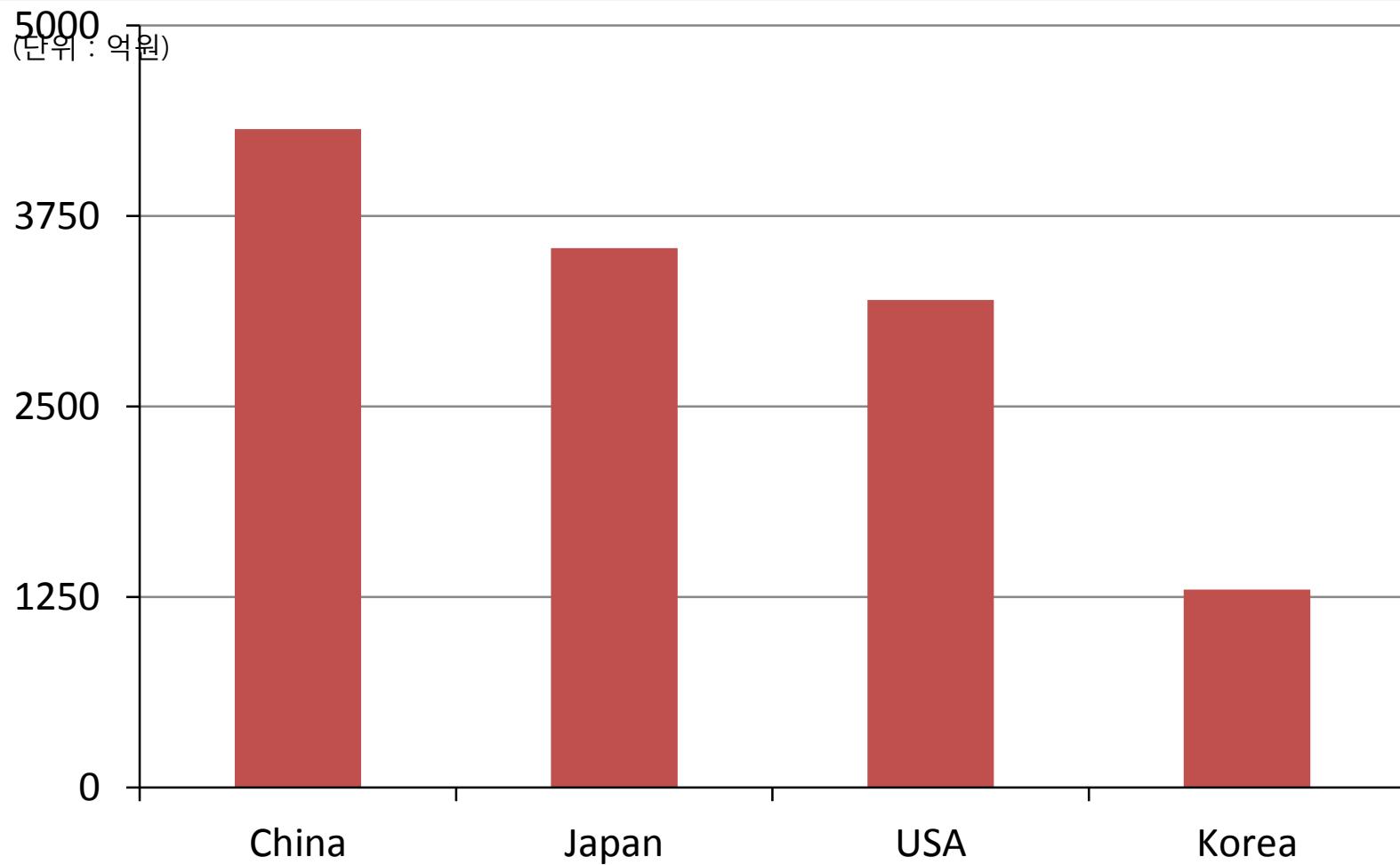
Android market share in China reached 78.6% in the fourth quarter of 2013 according to a [Kantar World Panel Report](#).

But one major difference between Android users in China and elsewhere is how they download and buy apps on their phones. While Google Play, Google's marketplace for Android phones is the primary source for Android apps in most markets, Chinese mobile users download and purchase apps from numerous third party app stores whose shared penetration reduce Google Play store to a very small player in the Chinese app market.



2013 중국 앱 수입 (이머징 마켓)

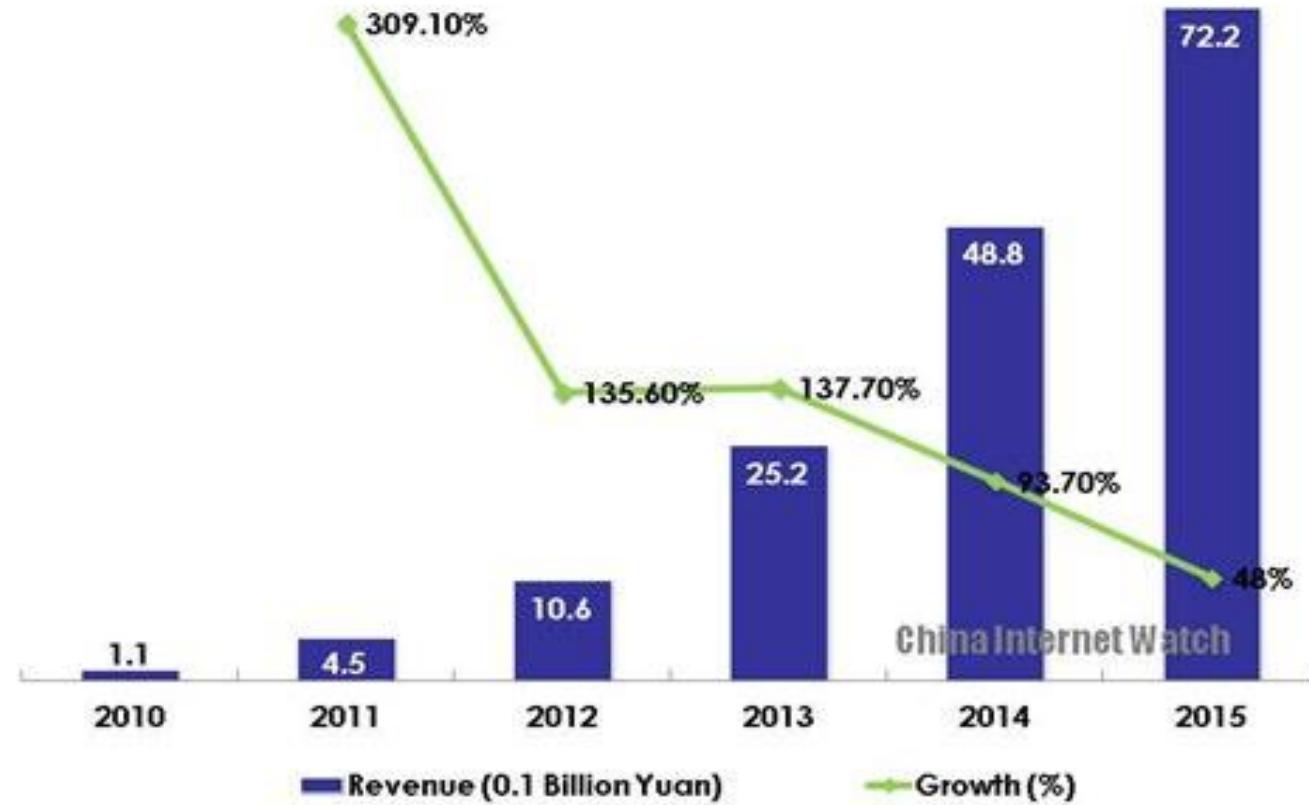
2013 Revenue



China Market

광고 수익은 전세계 1위
앞으로 성장 가능성 3배 이상 2년안에 원화 약 1조 3000억원

Revenues of China Mobile App Ad Platform From 2010 to 2015



중국에서만 볼수 있는 믿지 못할 에러들..

Introduction

Mid
Status

Current
Status

Competition

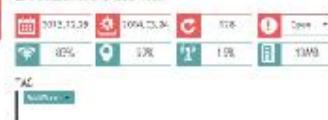
Technical
Issue

Plan

CN Error

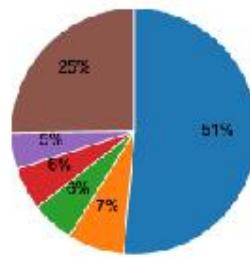


an error occurred by NetFwInException NoActivity found to handle listen failed and failed to handle PDU data on port 179 in session 1200 media 10000000000000000000000000000000



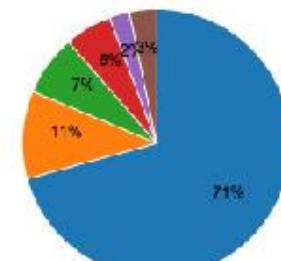
Device

● Readboy_G60 ● G1-i9100 ● C1801_Eyes ● NX503A
● LG-F320K ● Others



Country

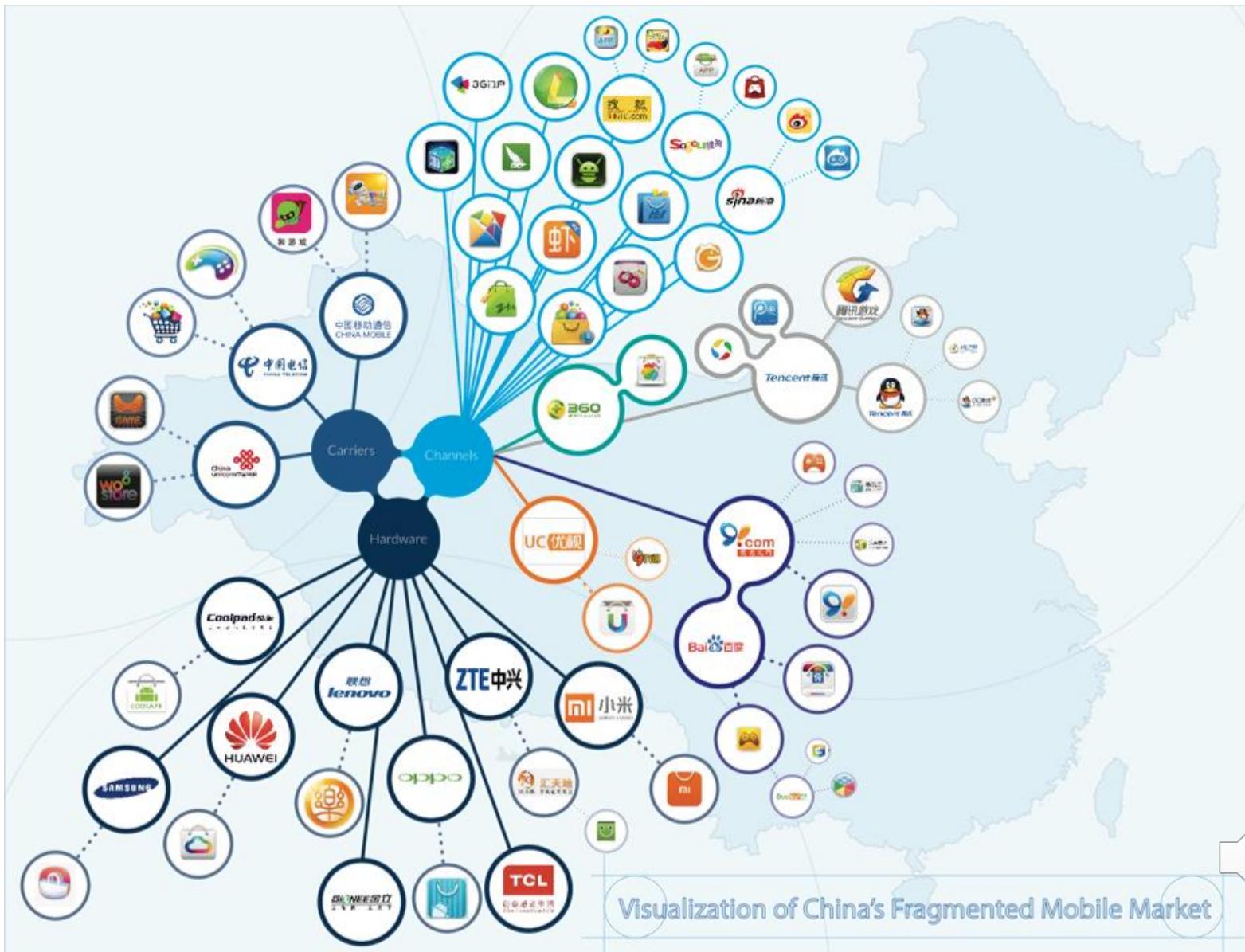
● CN ● US ● KR ● JP ● DE ● Others



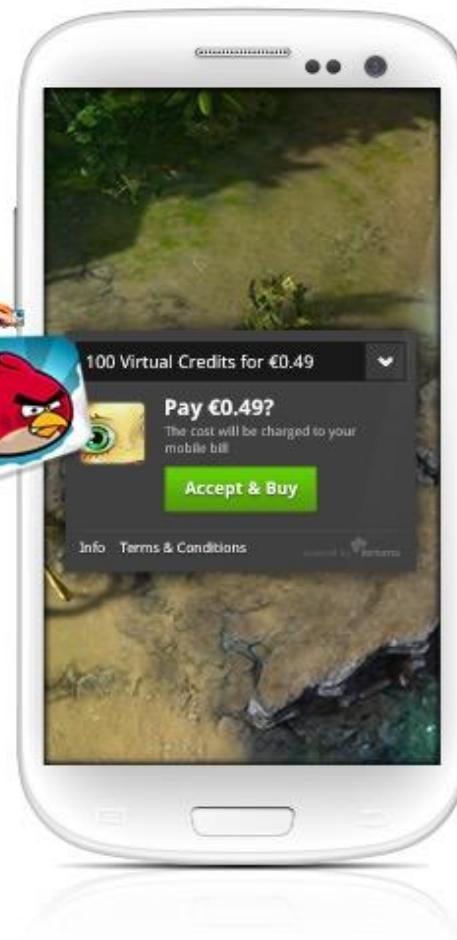
안드로이드 갤러리를 단순히 띄우는 기능



믿지못할 마켓 단편화..



Android In-App Billing Tracking



Chinese App Stores:



[More info >](#)

OEM App Stores:



[More info >](#)

3rd Party App Stores:



nook

Fortumo is the official In-App purchasing partner for NOOK Apps.
[Read more >](#)

Carrier App Stores:

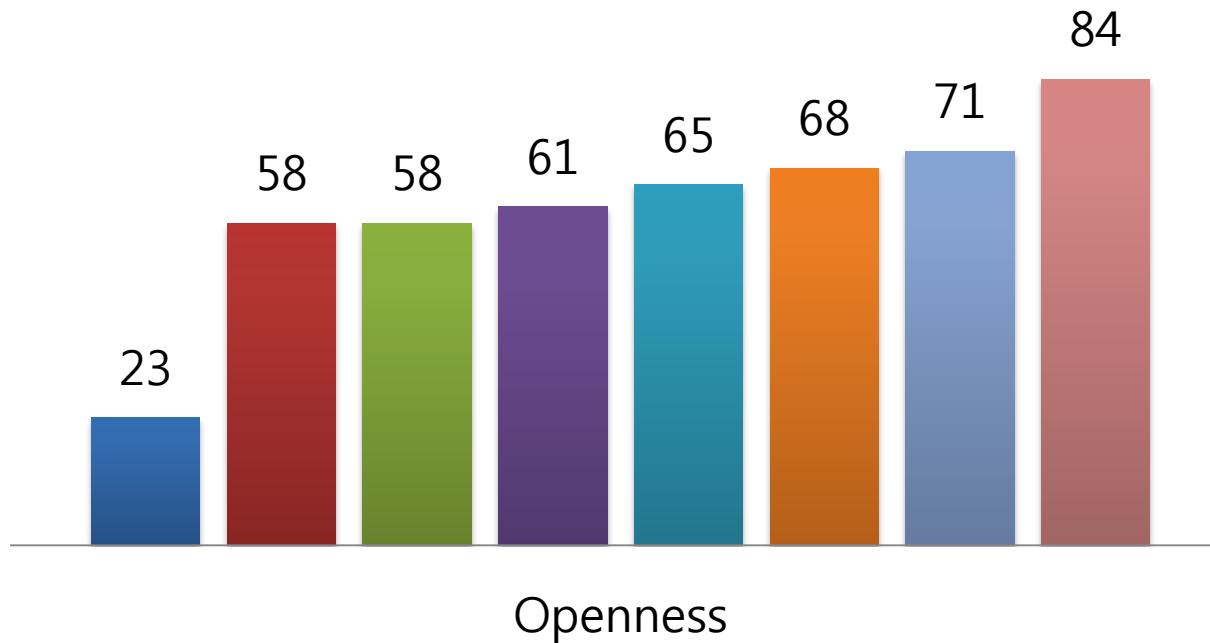


[Read more >](#)



안드로이드의 개방성

■ Android ■ Qt ■ Symbian ■ Meego
■ Mozilla ■ Webkit ■ Linux ■ Eclipse



<http://bit.ly/oTmEmF>

오픈 소스 이긴 한데.. 구글러가 아니면 할수 있는게 없다...

Best practices

Ease of source-code access via the Apache License

Ease of access to mailing lists, very good developer tools and forums

Simple code-contributions process for developers to follow

Clever targeting of developers via the Android Challenge, Summer of Code, etc.

Practices to avoid

Unilateral Android project decision-making processes, as Google determines the roadmap, feature-set and releases of Android

Closed code committer process, i.e., committers are exclusively Google personnel

Closed contributions process model

Opaque decision-making and control process around the Android Compliance Program

No project metrics around contributions, commits, contributors, top participants and bugs

No public information provided regarding meeting minutes or decisions.

No intention to move towards a more open governance model



정리하면..

버그만 잡아줘..



구글의 폐쇄성으로 안드로이드에

좋은 기능을 외부에서는 추가할수 없다.

오직 구글 직원만 가능..



안드로이드 오픈소스(AOSP)를 이용한 반구글 전략은 중국의 스마트폰 업체인 샤오미, 화웨이, ZTE 등도 추진

Advanced OS (Millions)	4Q 2013	Share %	4Q 2012	Y on Y Growth	3Q 2013	Share %	Q on Q Growth
Android & AOSP	221.5	77%	146.7	51%	184.5	80%	20%
Android	150.4	52%	116.7	29%	134.5	58%	12%
AOSP	71.1	25%	30.0	137%	50.0	22%	42%
Apple iOS	51.0	18%	47.8	7%	33.8	15%	51%
MS Windows Phone	10.9	4%	5.3	104%	9.1	4%	19%
BlackBerry OS	3.2	1%	6.9	-54%	2.3	1%	42%
BlackBerry 10	1.1	0%	NA		1.4	1%	-23%
Firefox	.1	0%	NA		.0	0%	1714%
MS Windows Mobile		0%	.1	-100%	.0	0%	100%
Symbian		0%	2.1	-100%		0%	
Grand Total	287.8	100%	208.9	38%	231.1	100%	25%

Sources: ABI Research



늘어나는 안드로이드 점유율

Worldwide Smartphone Sales to End Users by Operating System in 2Q16 (Thousands of Units)

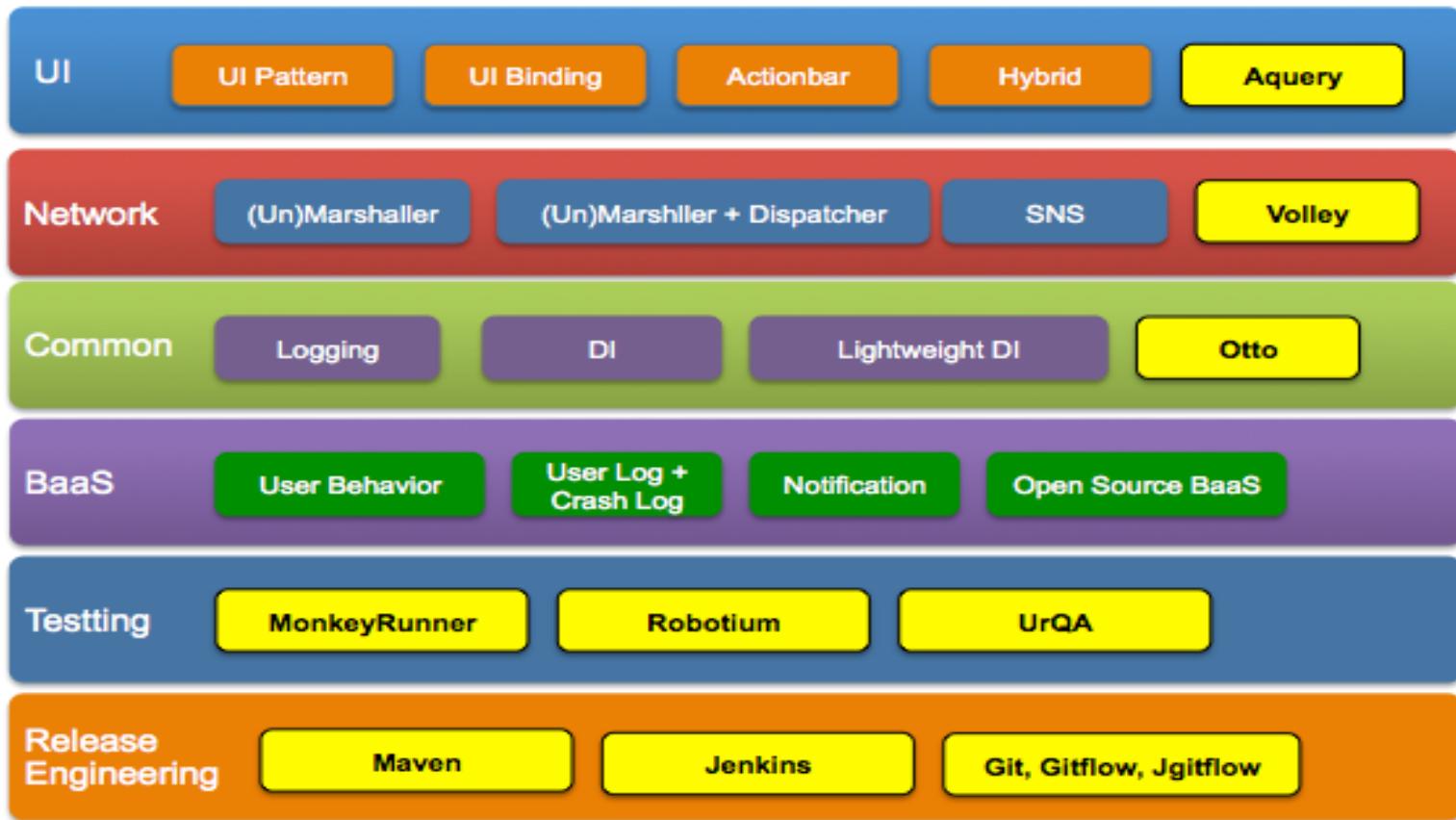
Operating System	2Q16 Units	2Q16 Market Share (%)	2Q15 Units	2Q15 Market Share (%)
Android	296,912.8	86.2	271,647.0	82.2
iOS	44,395.0	12.9	48,085.5	14.6
Windows	1,971.0	0.6	8,198.2	2.5
Blackberry	400.4	0.1	1,153.2	0.3
Others	680.6	0.2	1,229.0	0.4
Total	344,359.7	100.0	330,312.9	100.0

Source: Gartner (August 2016)

Company	2Q16 Units	2Q16 Market Share (%)	2Q15 Units	2Q15 Market Share (%)
Samsung	76,743.5	22.3	72,072.5	21.8
Apple	44,395.0	12.9	48,085.5	14.6
Huawei	30,670.7	8.9	26,454.4	8.0
Oppo	18,489.6	5.4	8,073.8	2.4
Xiaomi	15,530.7	4.5	15,464.5	4.7
Others	158,530.3	46.0	160,162.1	48.5
Total	344,359.7	100.0	330,312.9	100.0



안드로이드에 포함되지 못한 외부 오픈소스...



랭킹까지 매길 정도..

<http://www.appbrain.com/stats/libraries/dev>

AppBrain | Stats

Last updated: July 19, 2015

Android Statistics > Android libraries > Development tools

Google Play stats

- Number of available apps
- Free versus paid apps
- Apps by rating
- Apps by downloads
- Top categories
- In-app billing apps
- Monthly Top
- CPI per country

Android device stats

- Top Android phones
- Top manufacturers
- Top Android SDK versions
- Devices by country

AppBrain site stats

- Top search terms on AppBrain

Android library stats

- Android ad network stats
- Social SDK stats

Development tool stats

Promote your Android app with AppBrain

Monetize your Android app with AppBrain

Other tools for developers by AppBrain

Sort by: Installs, Apps Marketshare in: Overall, New, Top 500

Library	Description	Marketshare in apps	Marketshare in installs
Android Support library	The Support Package includes static 'support libraries' that you can add to your Android application...	57.26% of apps	80.48% of installs
Android Cloud to Device Messaging (C2DM)	Android Cloud to Device Messaging (C2DM) is a service that helps developers send data from servers ...	15.44% of apps	43.79% of installs
Flurry Analytics	Flurry Analytics delivers powerful insight into how consumers interact with your mobile application...	7.49% of apps	18.37% of installs
Google gson	Gson is a Java library that can be used to convert Java Objects into their JSON representation. It ...	9.05% of apps	14.54% of installs
BoltsFramework	Bolts is a collection of low-level libraries designed to make developing mobile apps easier.	4.44% of apps	14.42% of installs
Google Analytics	The Google Analytics SDK for Android makes it easy for native Android developers to collect user en...	12.41% of apps	13.80% of installs
NineOldAndroids	Android library for using the Honeycomb (Android 3.0) animation API on all versions of the platform...	3.66% of apps	10.82% of installs
fmod	The FMOD Ex sound system is a revolutionary audio engine for game developers, multimedia developers...	5.81% of apps	8.80% of installs
Google Guava	The Guava project contains several of Google's core libraries that we rely on in our Java-based	1.56% of apps	8.56% of installs

[View now!](#)

[Promote!](#)

[Earn money!](#)

[View now!](#)

[Sign up](#) [Log in](#)



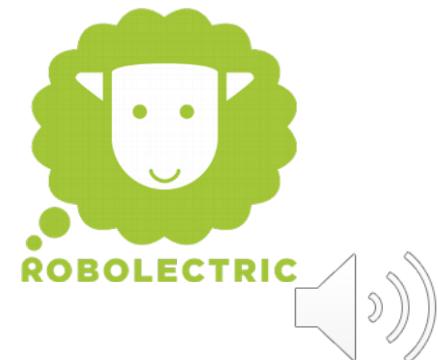
모바일 테스트 도구



uiAutomator



Calabash.sh



잘나가는 Testing tool

	Robotium	uiautomator	Espresso	Appium	Calabash
Android	Yes	Yes	Yes	Yes	Yes
iOS	No	No	No	Yes	Yes
Mobile web	Yes (Android)	Limited to x.y clicks	No	Yes (Android & iOS)	Yes (Android)
Scripting Language	Java	Java	Java	Almost any	Ruby
Test creation tools	Testdroid Recorder	UI Automator viewer	Hierarchy Viewer	Appium.app	CLI
Supported API levels	All	16 =>	8, 10, 15 =>	All	All
Community	Contributors	Google	Google	Active	Pretty quiet



대세는.. (실리콘밸리 기준)



<http://www.imaso.co.kr/?p=6176>

<http://www.imaso.co.kr/?p=6188>



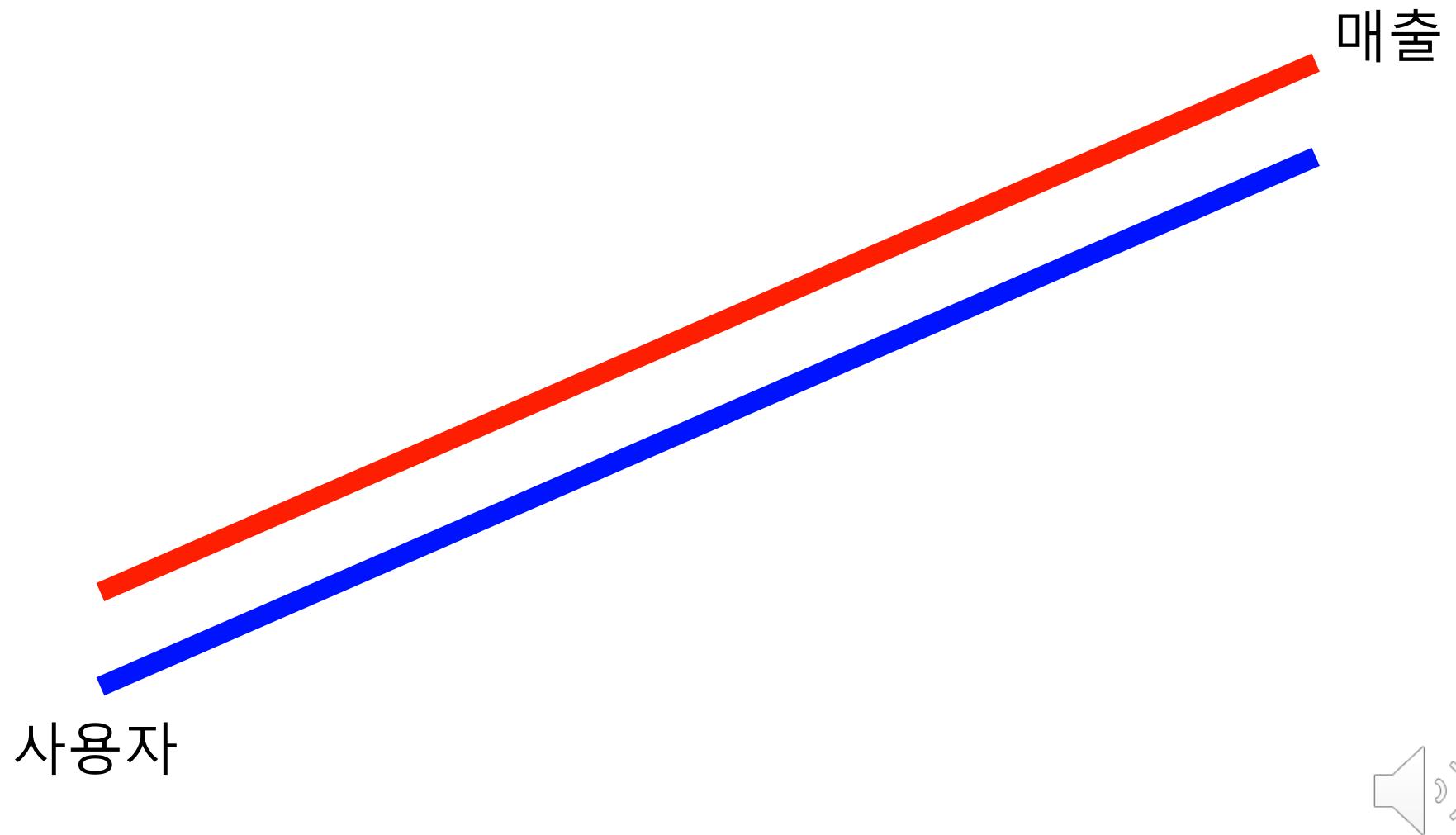
5. 모바일 때문에.. 백엔드에도 큰 변화가..



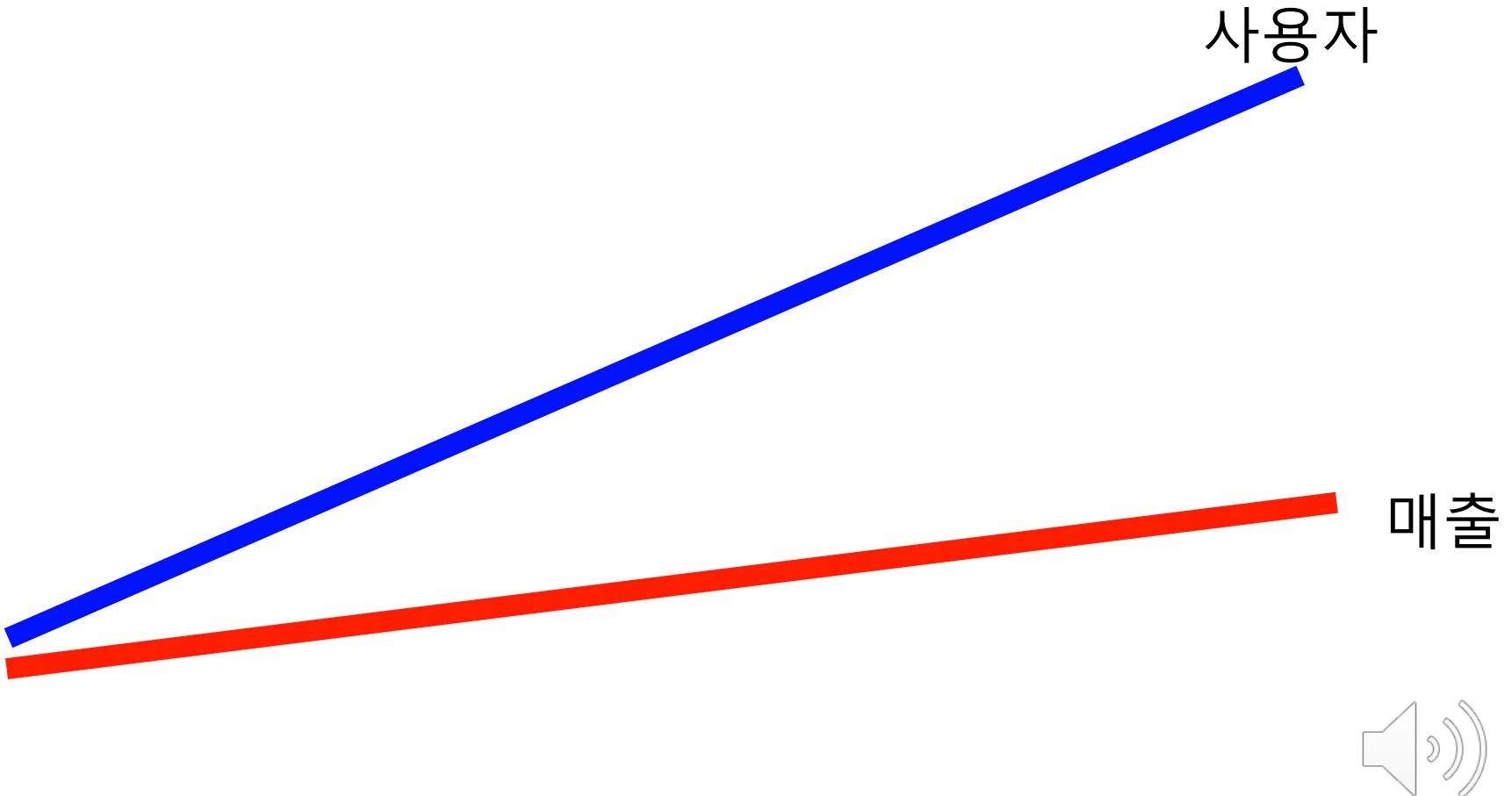
Backend란?



배경 - 예전에는.. (매출과 사용자는 비례관계)



배경 - 모바일이 나온후
(사용자가 많다고 매출이 좋은건 아니다..)



사용자가 많아도 수익이 별로다..

- 값 비싼 하드웨어 장비, 오라클.. 을 살수 없는데..
- 저렇게 많은 사용자를 어떻게 견디지??
- 소프트웨어, 하드웨어 구매할 비용이 안된다..



속내는..

- 비싼 하드웨어 장비 + 오라클 너무 비싸다.
- 대신 저렴하며 비슷한 효과를 낼 만한 것은?
- 클라우드 서버를 늘려가면서. (Scale Out) + 오픈소스 솔루션으로..

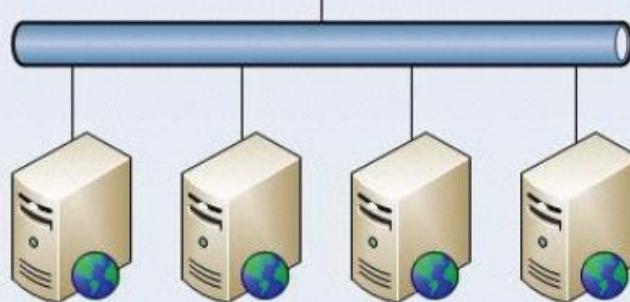


Scale Out vs Scale Up

Scale-Up vs. Scale-Out



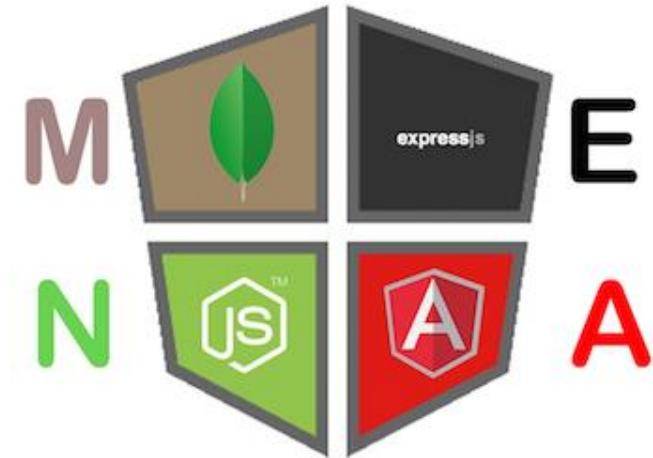
Scale Out



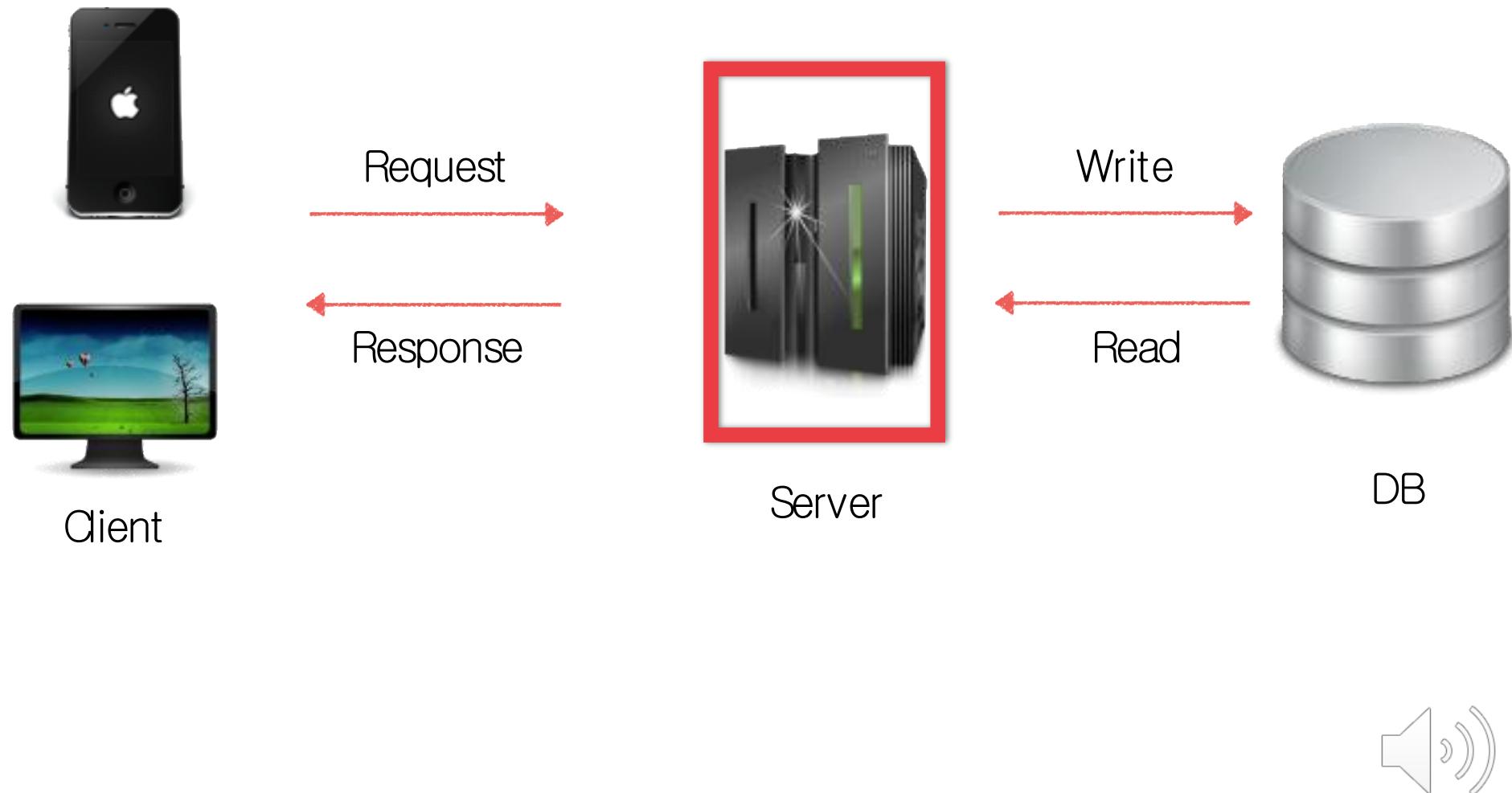
Scale Up



그래서 나온.. 여러가지 오픈소스 조합들...



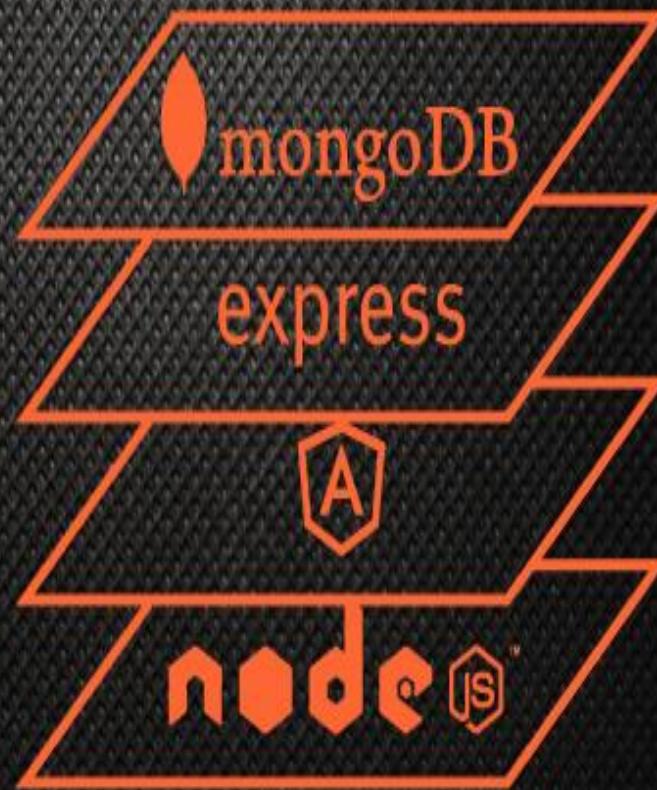
간략한 서비스 구성



LAMP vs MEAN



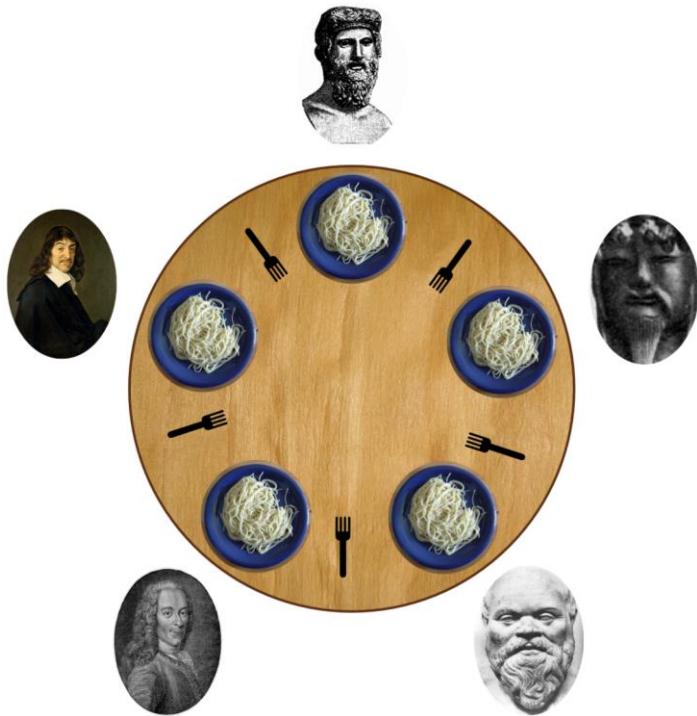
VS



대세 MEAN Stack



Node.js 장점



Single thread



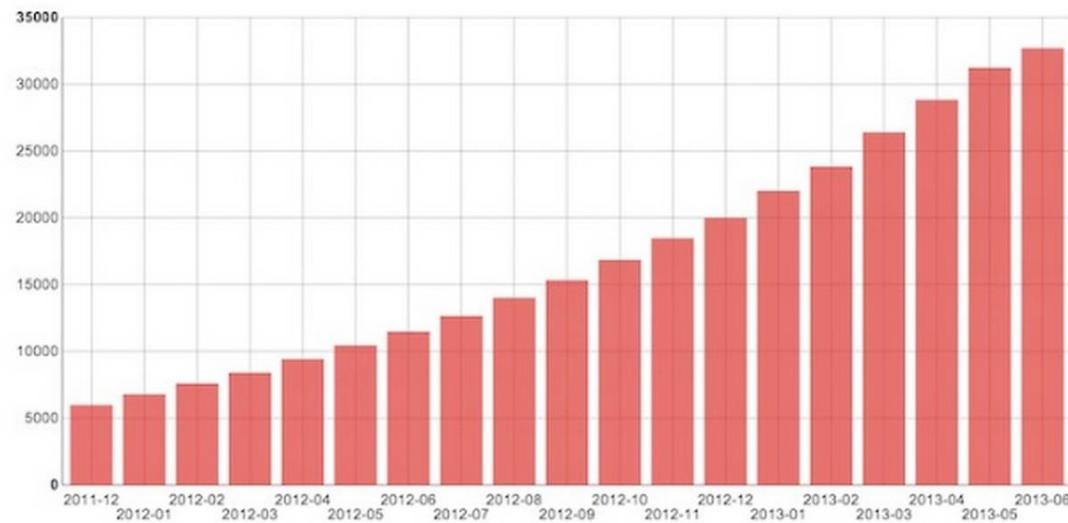
Node.js 장점

The screenshot shows the npm homepage. At the top left is the red 'npm' logo. To its right is a search bar with the placeholder 'find packages'. Next to the search bar is a magnifying glass icon. Further right are links for 'sign up or log in' and a user profile icon. Below the header, a large text area says 'npm is the package manager for javascript.' in black and red. Underneath this, there are four data cards: 1) A red-bordered box showing '116,156 total packages'. 2) A card showing '29,280,711 downloads in the last day'. 3) A card showing '123,347,027 downloads in the last week'. 4) A card showing '617,334,443 downloads in the last month'. Below these cards is a red text link that reads 'packages people 'npm install' a lot'.

Node Package Manager

Node.js 장점

node.js – npm modules The scale of npm modules



Maciej Lasyk, scaling&securing node.js apps, #AtmosphereConf 2014



이션

YAHOO!

yammerTM
The Enterprise Social Network

PayPalTM

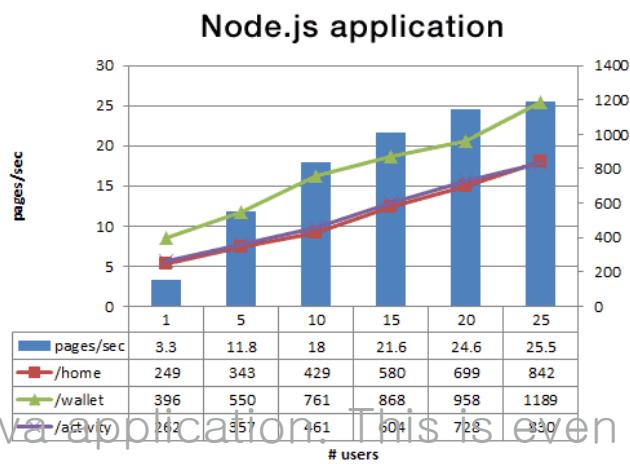
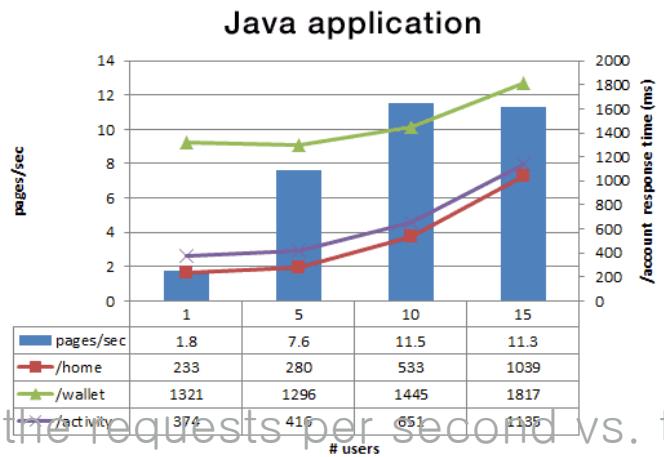
U B E R

LinkedInTM

Windows[®] AzureTM



Node.js Paypal

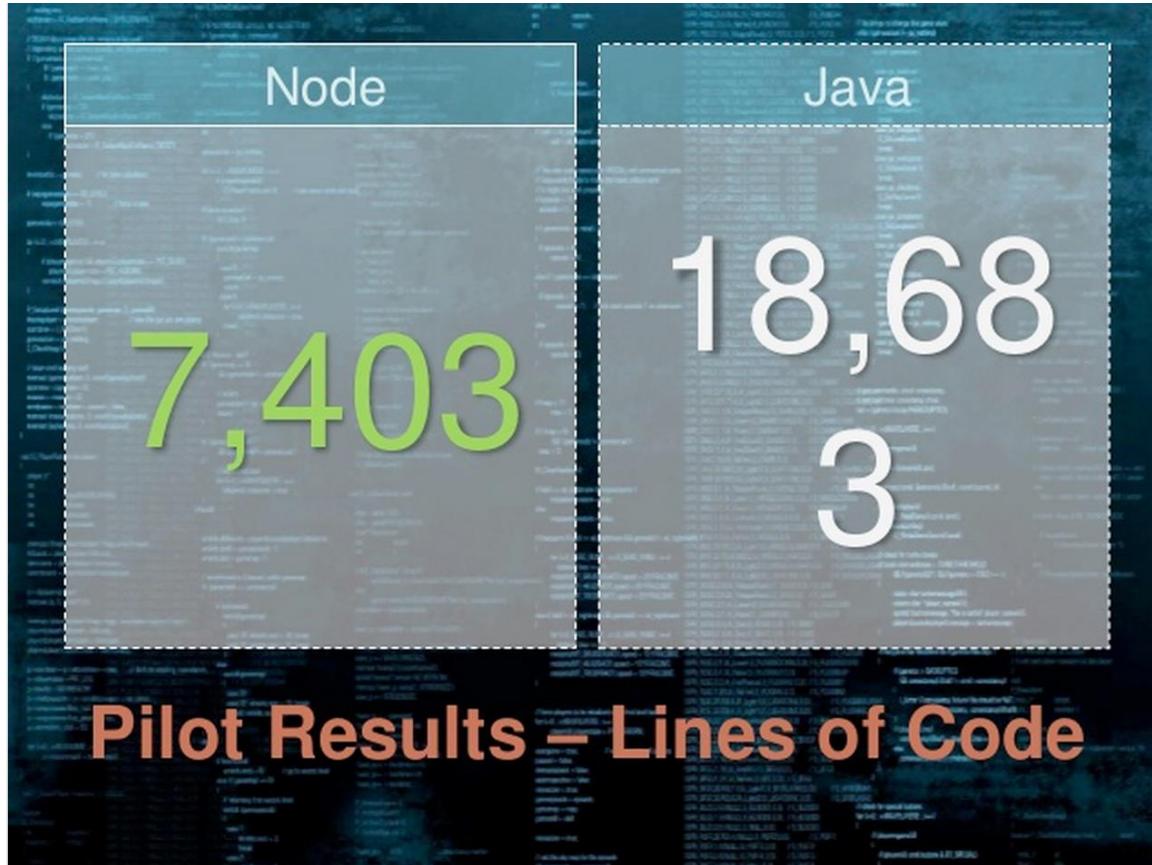


Double the requests per second vs. the Java application. This is even more interesting because our initial performance results were using a single core for the node.js application compared to five cores in Java. We expect to increase this divide further.

35% decrease in the average response time for the same page. This resulted in the pages being served 200ms faster— something users will definitely notice



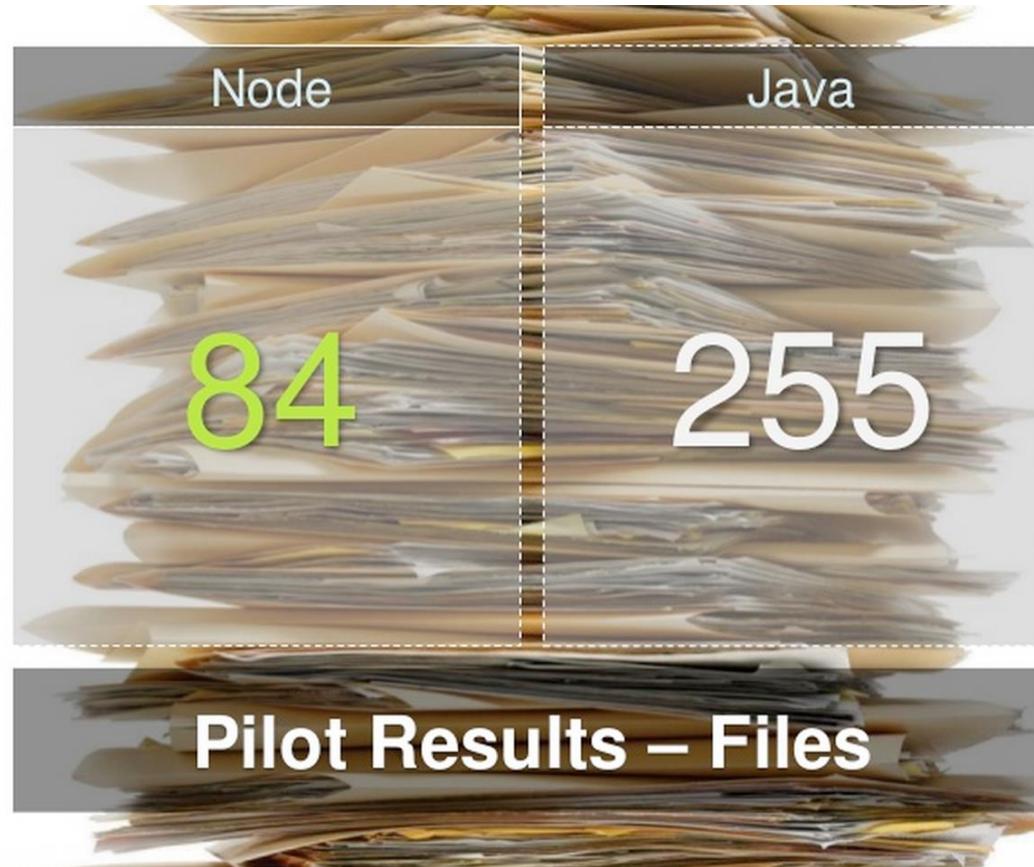
Node.js Paypal



[출처]<http://www.slideshare.net/lennymarkus/node-js-at-paypal>



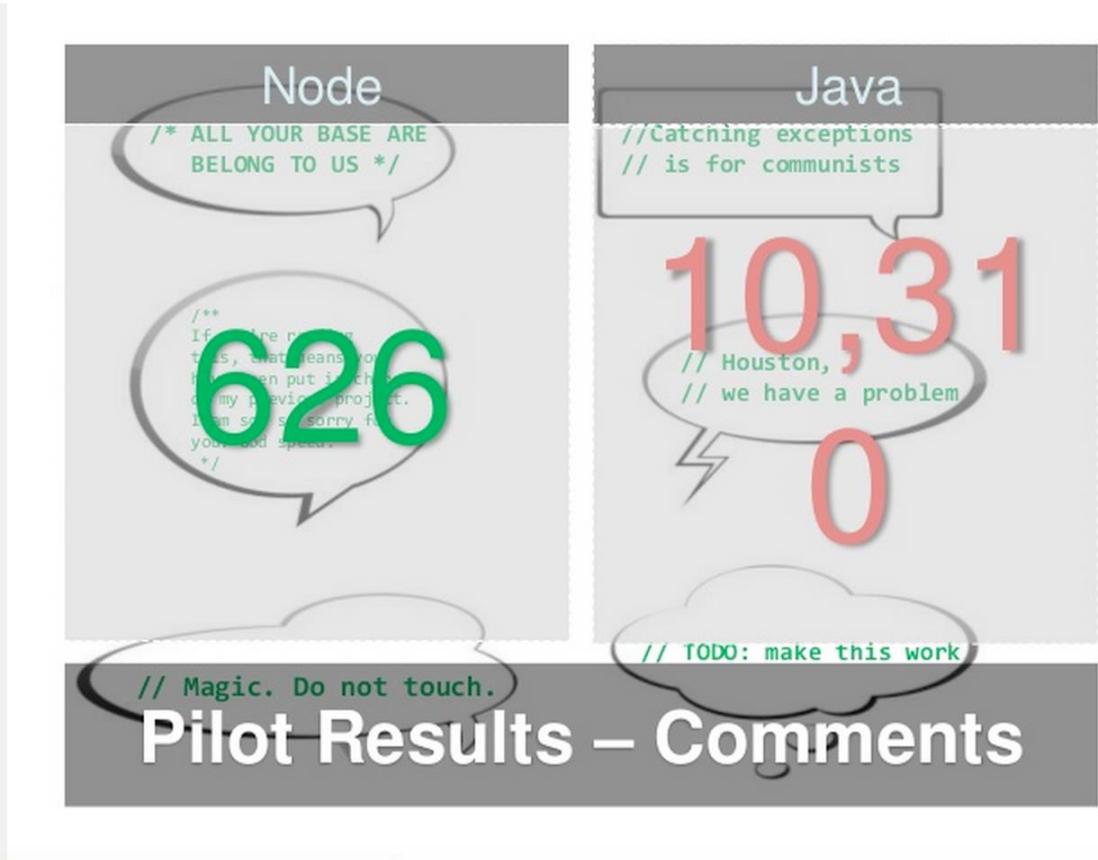
Node.js Paypal



[출처]<http://www.slideshare.net/lennymarkus/node-js-at-paypal>



Node.js Paypal



[출처]<http://www.slideshare.net/lennymarkus/node-js-at-paypal>



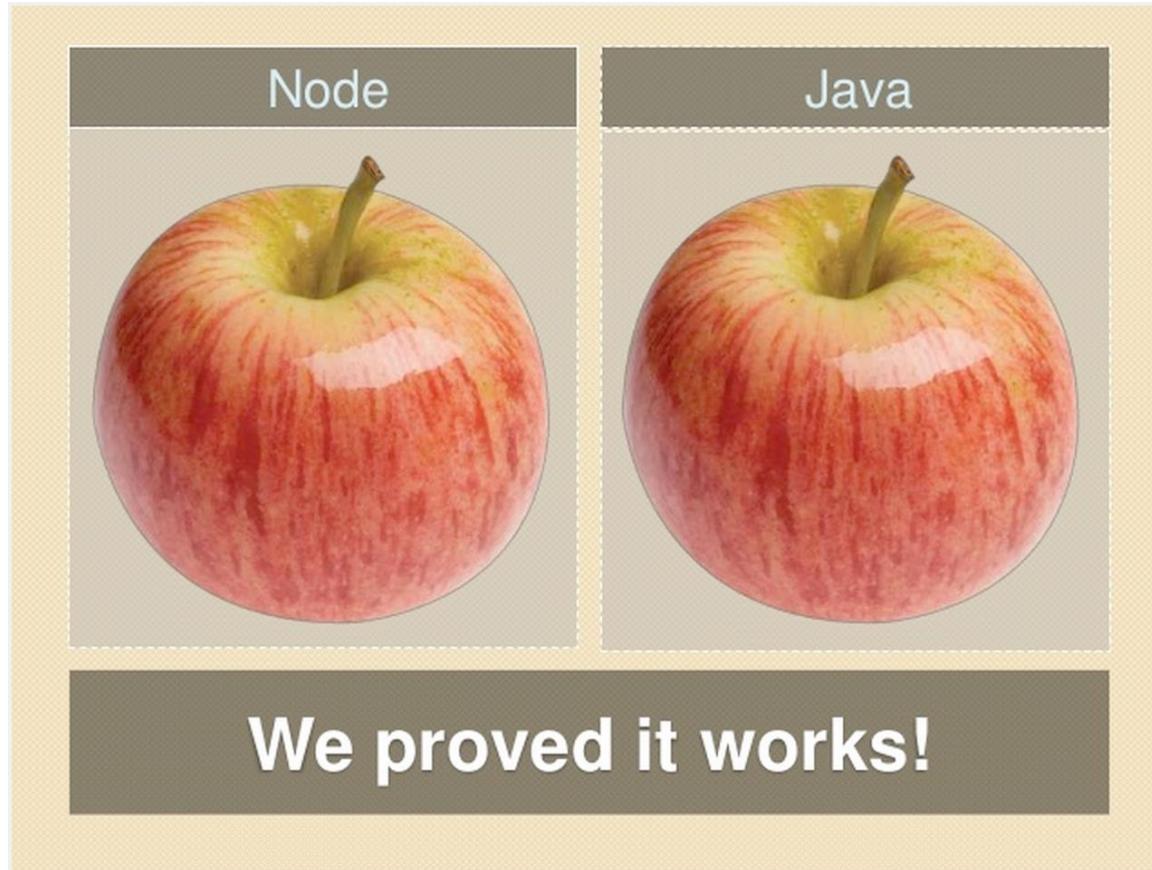
Node.js Paypal



[출처]<http://www.slideshare.net/lennymarkus/node-js-at-paypal>



Node.js Paypal



[출처]<http://www.slideshare.net/lennymarkus/node-js-at-paypal>



Node.js 좋은 활용

- REST + JSON APIs
- Single-page web app
- Real-time web app
- 빠른 prototyping
- 많은 I/O 처리

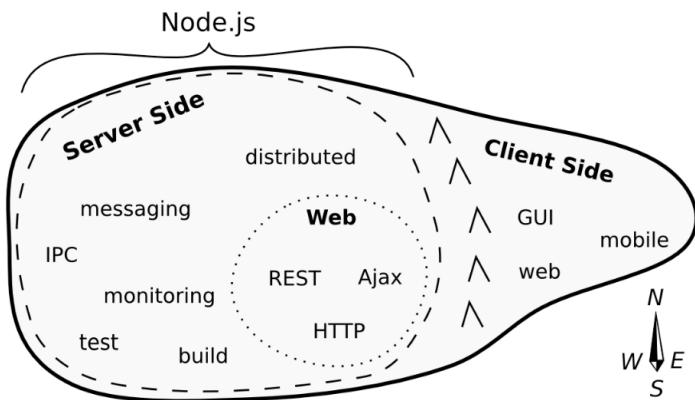


Node.js 나쁜 활용

- CPU 부하 걸리는 작업
- Multi-thread app
- 큰데이터 연산
- 복잡한 비지니스 로직



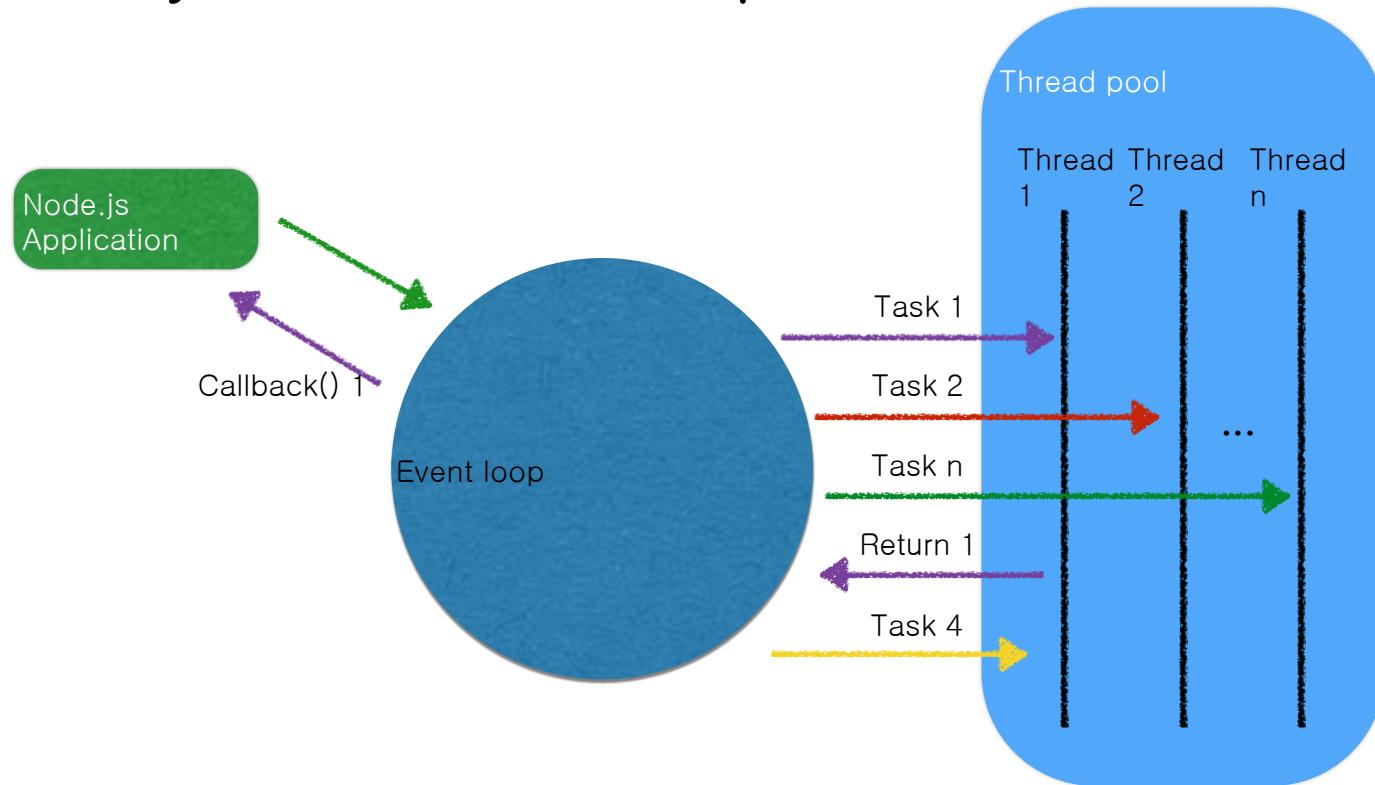
#02. Node.js의 철학 및 아키텍처



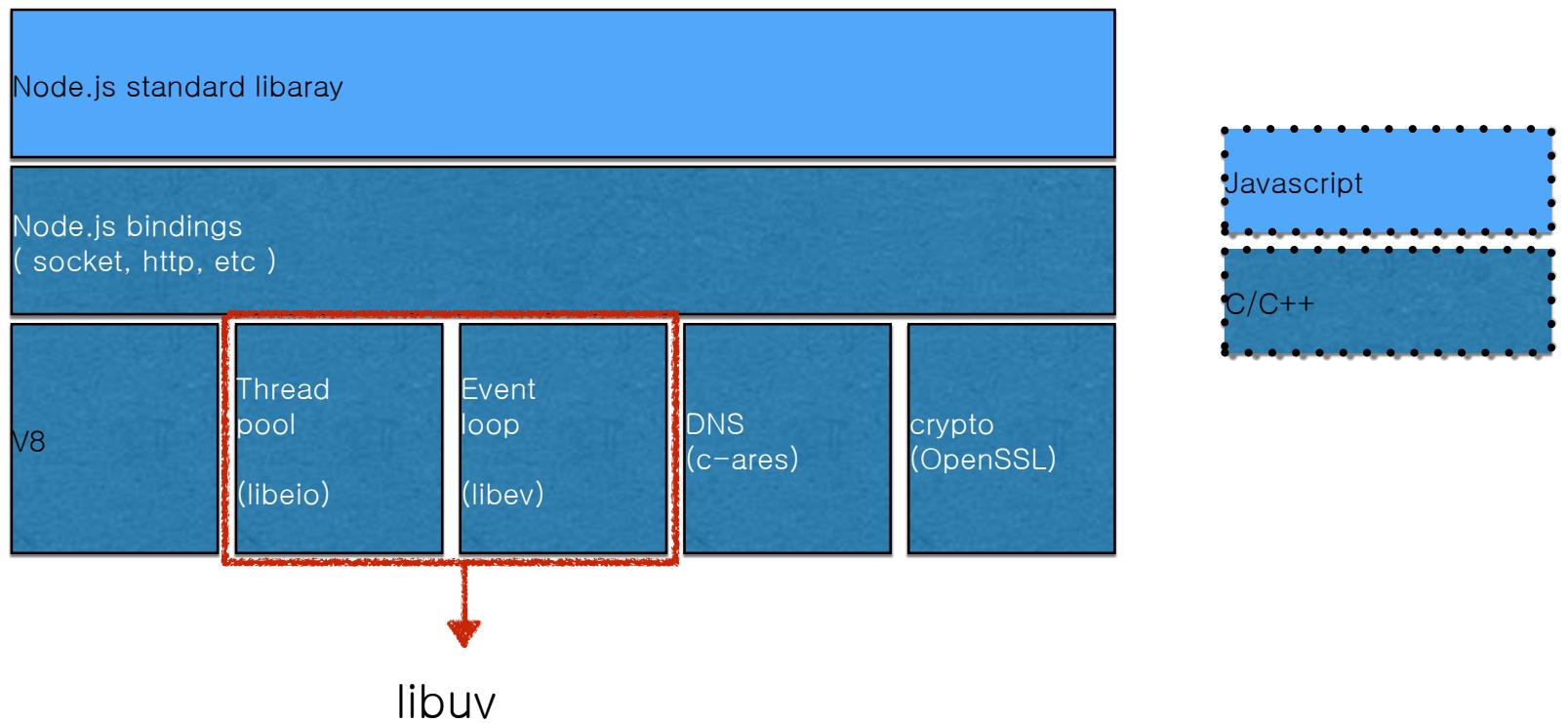
- Server side Javascript
- Event driven
- Asynchronous
- Non-Blocking I/O
- Single Threaded
- Lightweight
- Fast



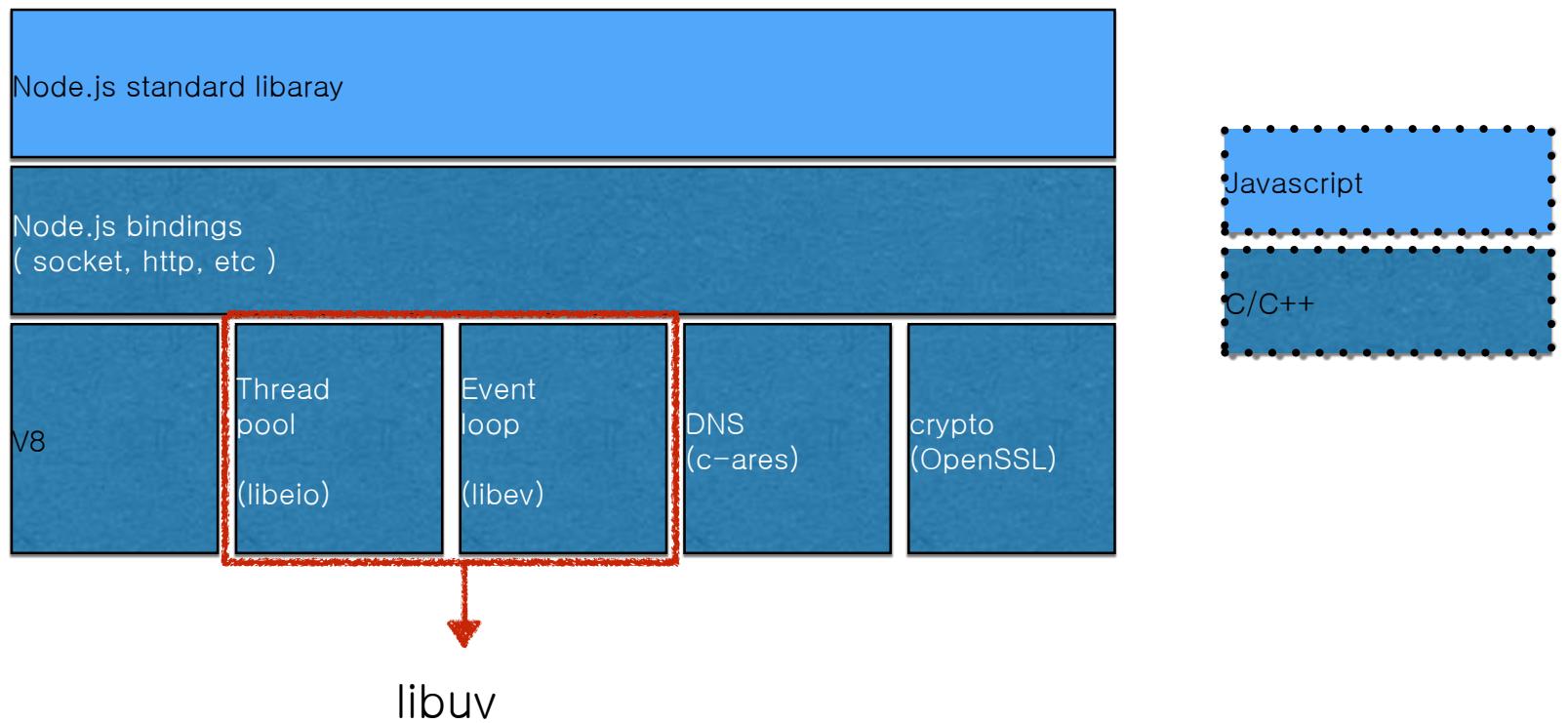
Node.js Event Loop



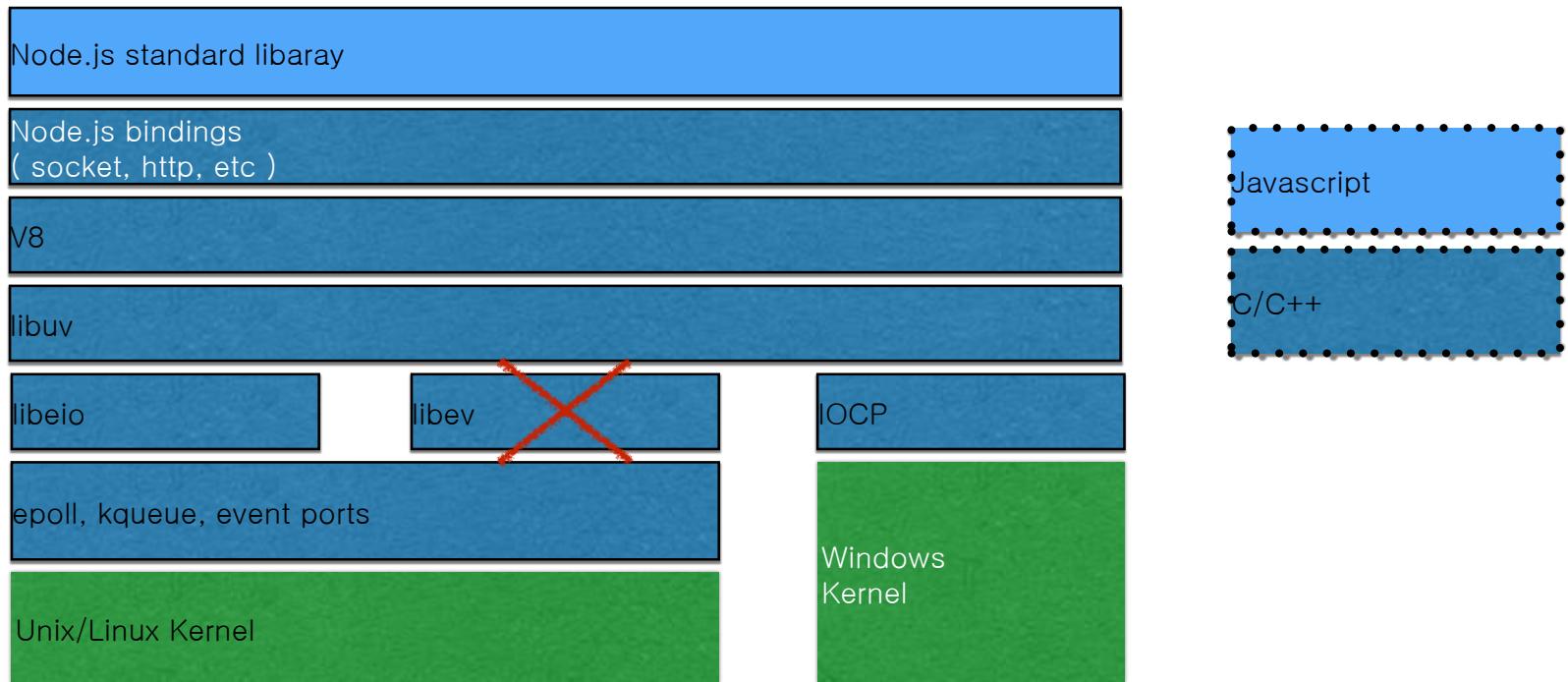
Node.js – Architecture



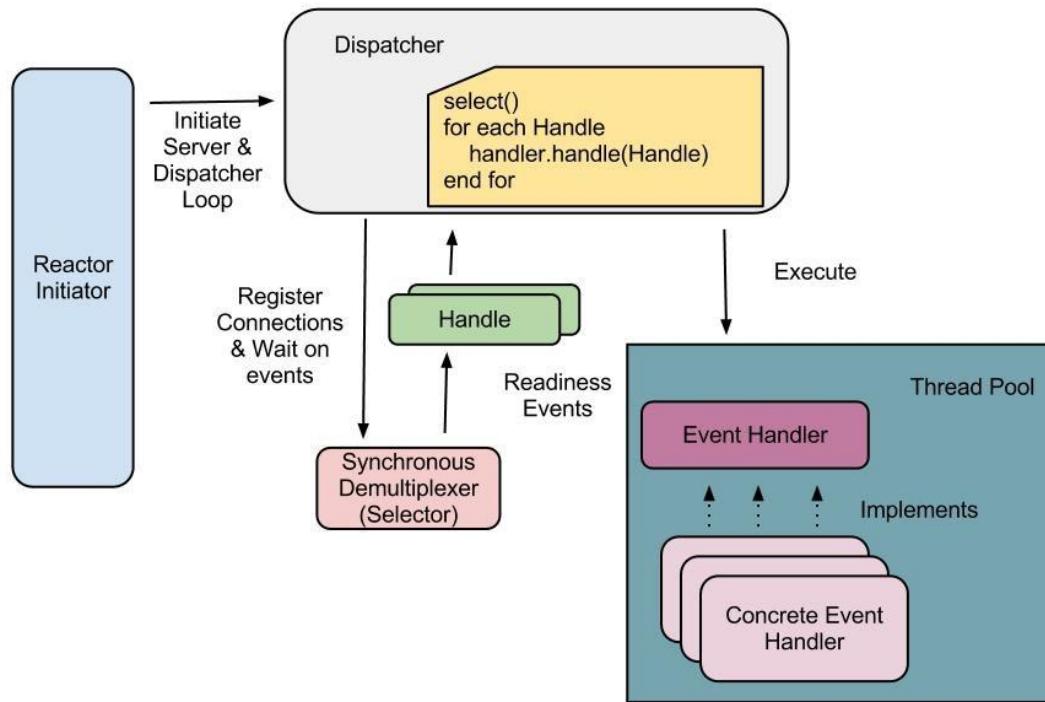
Node.js – Architecture



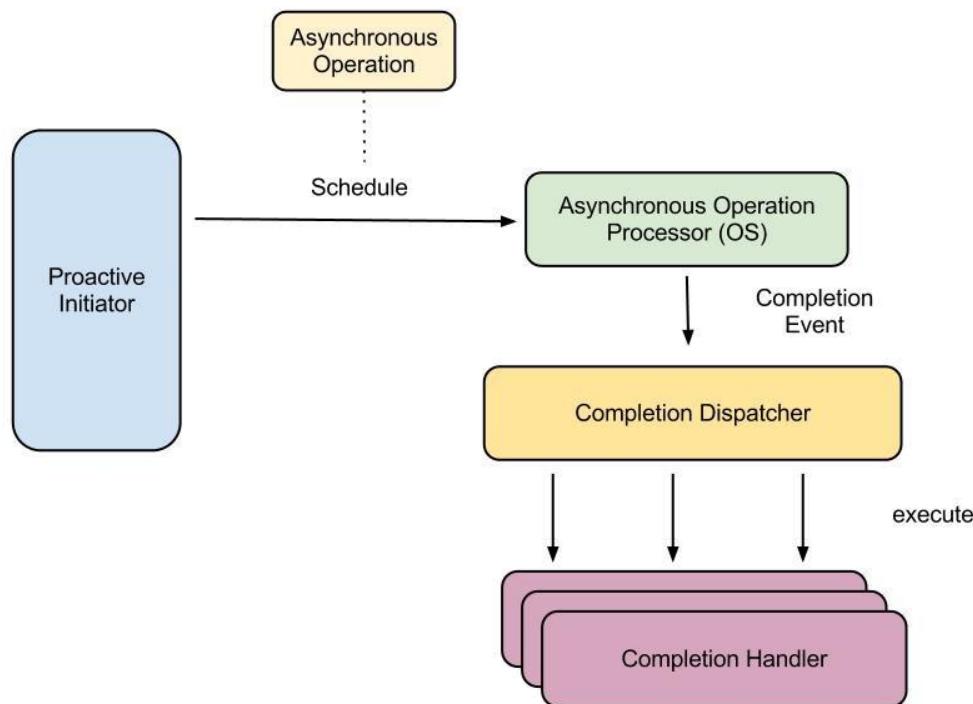
Node.js – Architecture



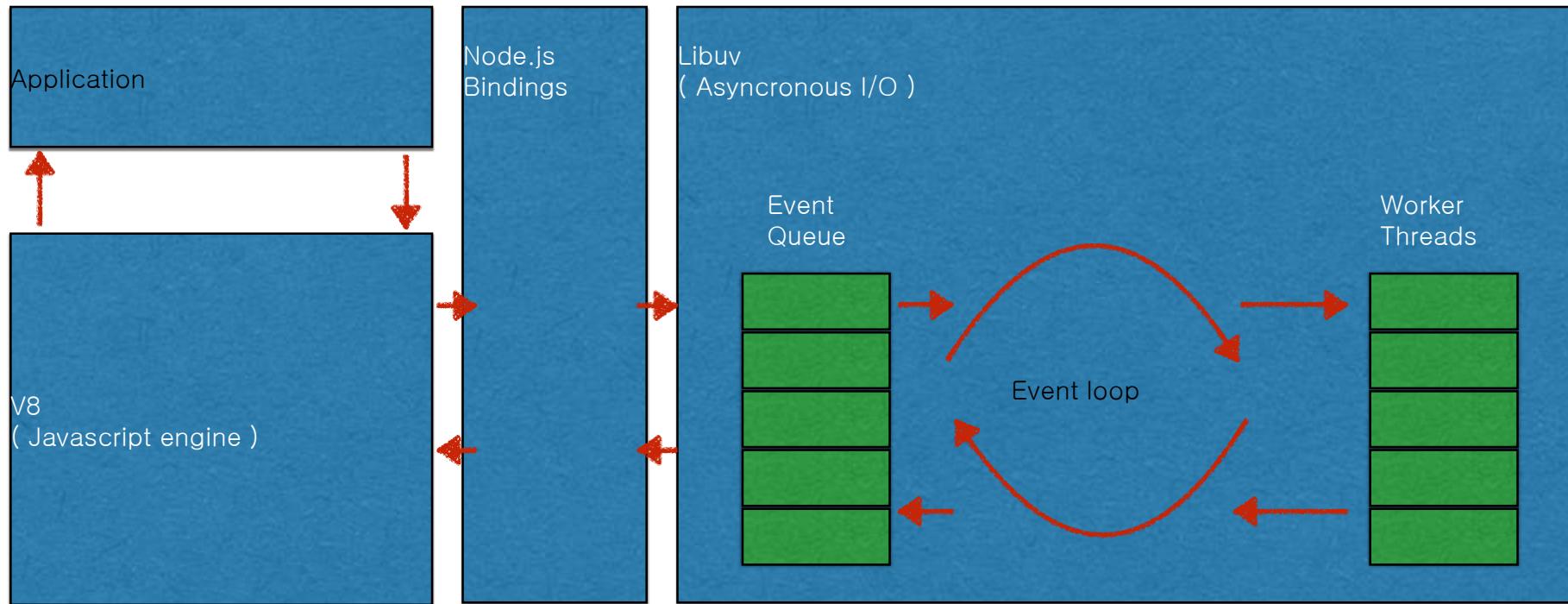
Reactor – select , epoll



Proactor – IOCP



Node.js System

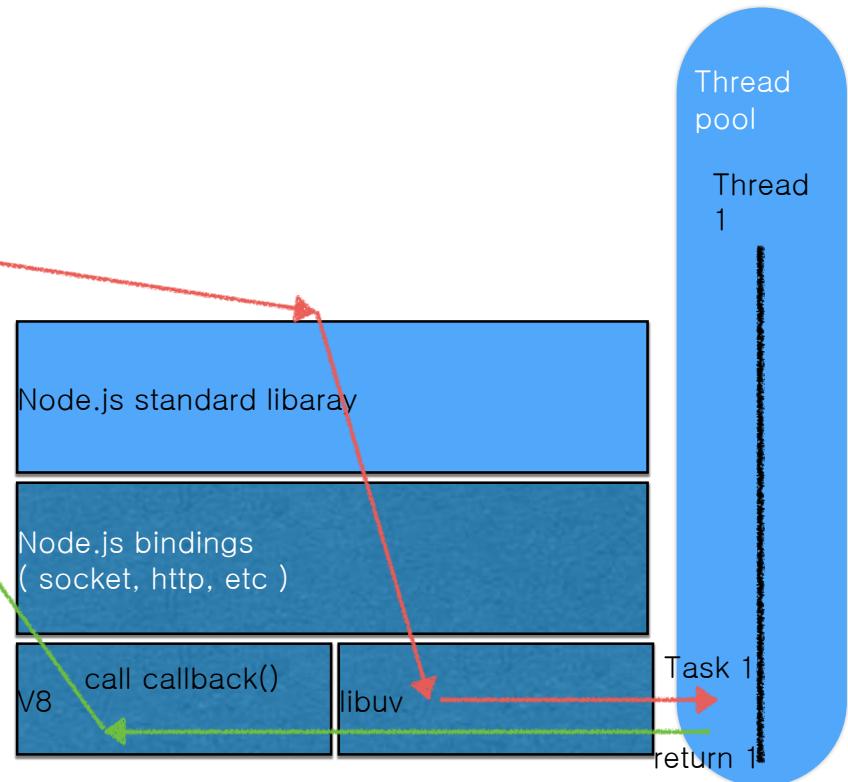


Node.js Event Loop

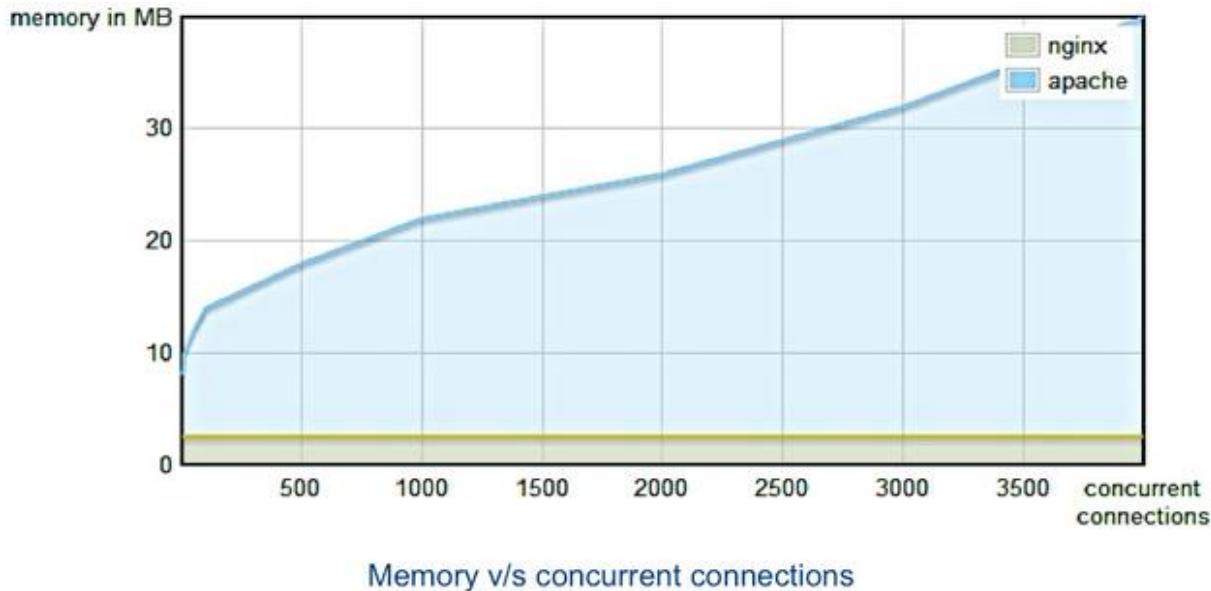
```
var fs = require('fs');

fs.readFile('./data.txt', function(err, data) {
  if( err ) { throw err; }

  console.log(data);
});
```



Memory Usage : Nginx vs Apache

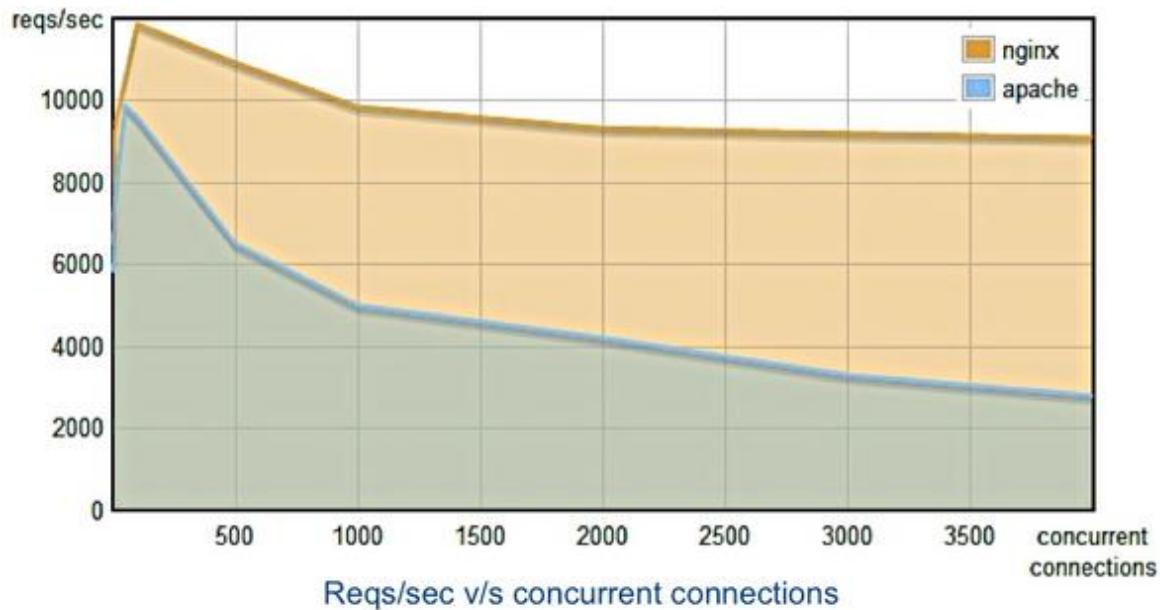


At ~4000 concurrent connections,
- Nginx uses 3MB memory
- Apache uses 40MB memory

Ref: <http://blog.webfaction.com/a-little-holiday-present>



Performance : Nginx vs Apache

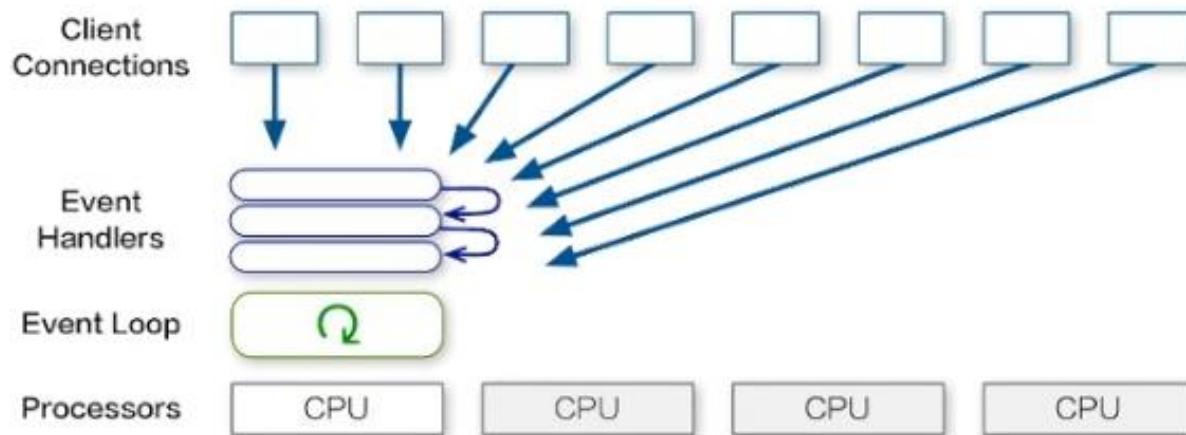


At ~4000 concurrent connections,
- Nginx can serve ~9000 reqs/sec
- Apache can serve ~3000 reqs/sec

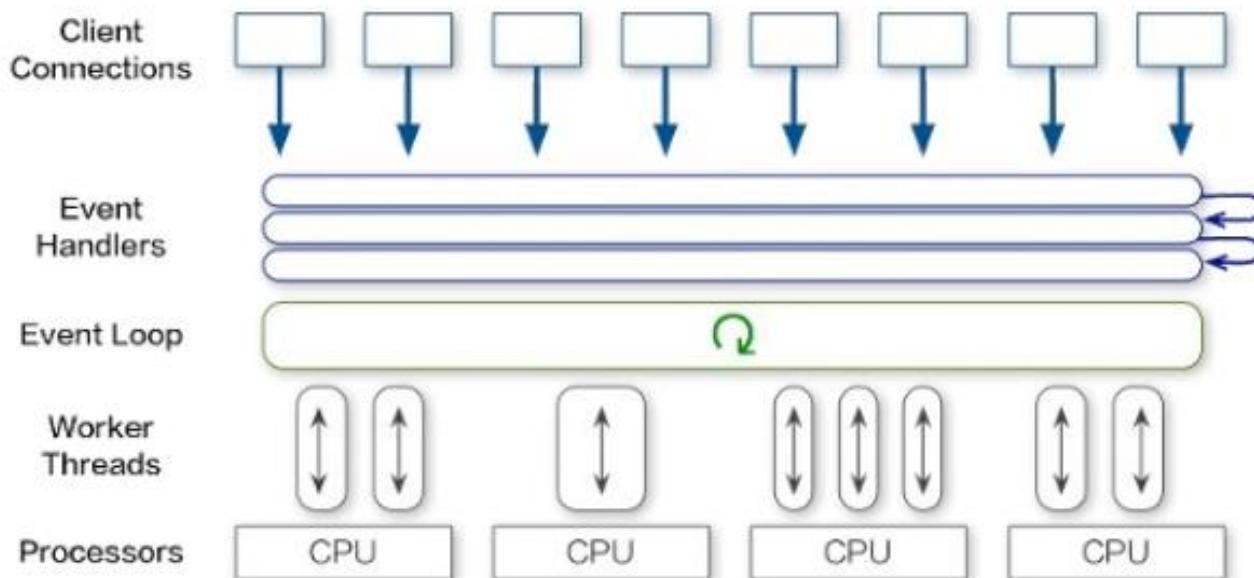
Ref: <http://blog.webfaction.com/a-little-holiday-present>



Single Thread가 성능이나요? CPU가 놀지 않나요?



Node.js Production 레벨에서는
CPU 갯수 + @ 만큼 Single Thread를 뛰워요!



Benchmark 1

json 1초에 hello world 반환하는 실험

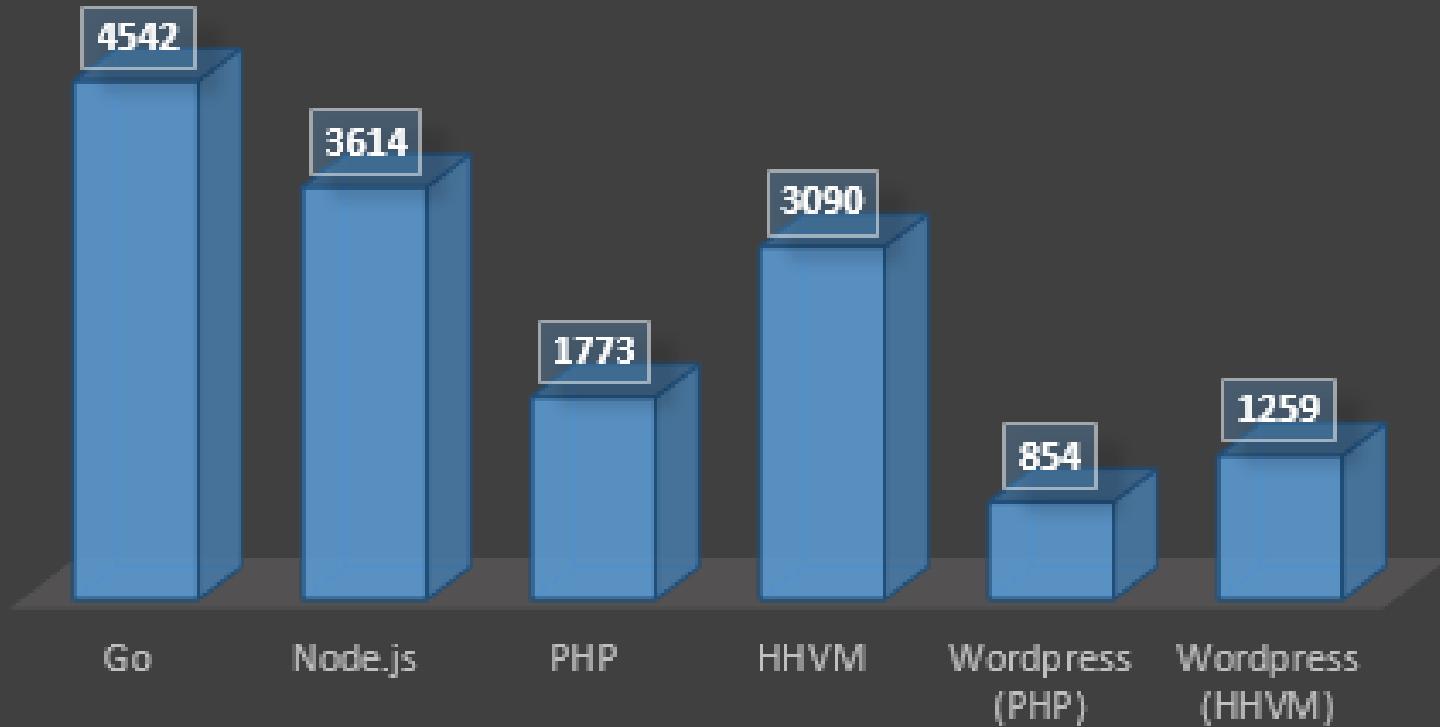
- node.js 225,328
- spring : 97,354
- php : 167,888
- <https://www.techempower.com/benchmarks/#section=data-r10&hw=peak&test=json>



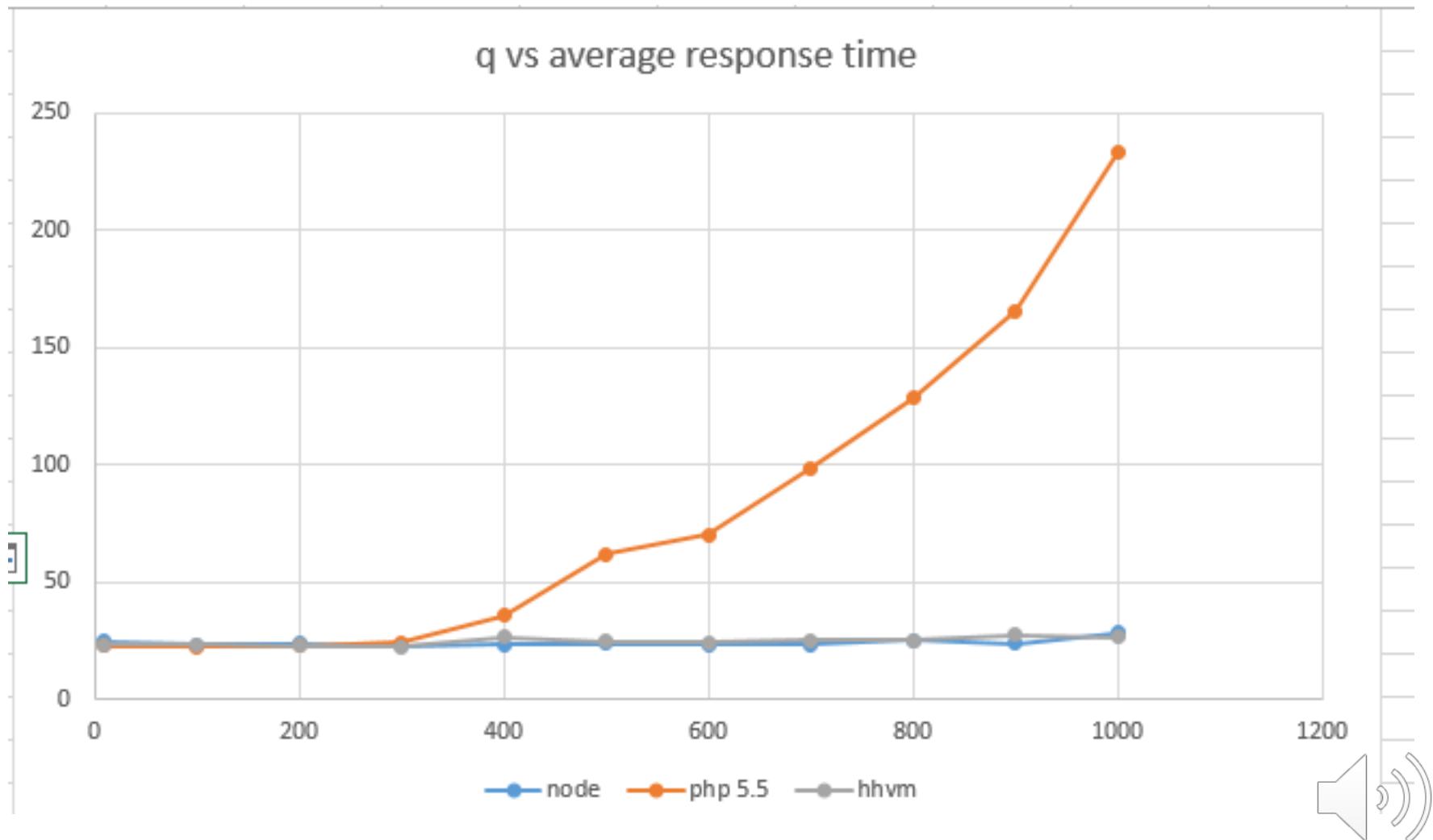
Benchmark 2 : 간단한 http 요청

<http://www.hostingadvice.com/blog/comparing-node-js-vs-php-performance/>

REQUESTS PER SECOND



Benchmark 3 : HTTP & CPU Task



Benchmark 4 : 정렬 CPU 테스트

	CPU time	System time	RAM
PHP 5.6.4	102.69s	104.20s	2497508 KB
HHVM 3.5.0	12.56s	14.83s	362488 KB
Node.js v0.10.35	2.64s	2.64s	92240 KB



Golang

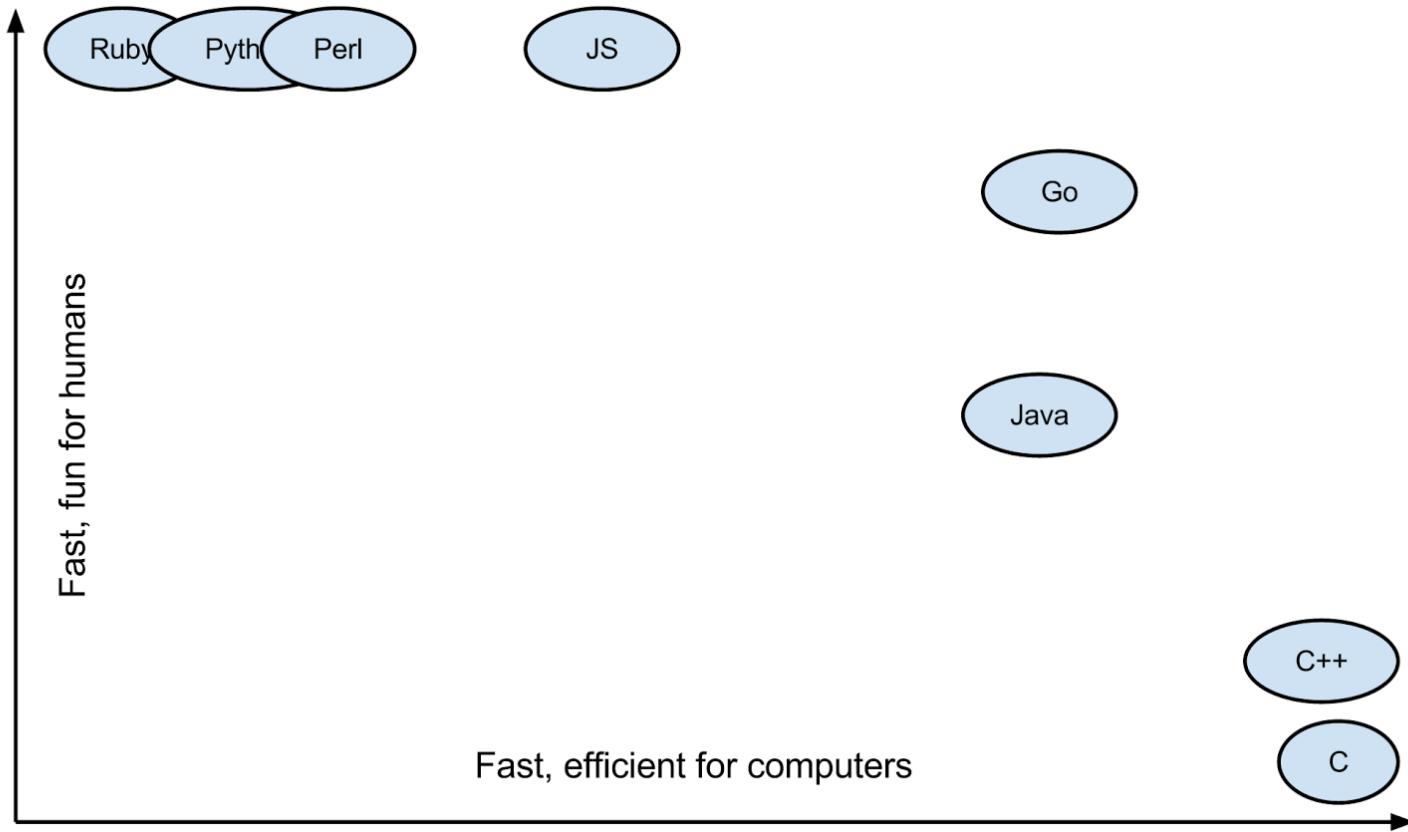


Go의 디자인 철학

performant
multiple-cores
concurrency
compiled
network



Go의 목적



Who made this thing?

- Ken Thompson (B, C, Unix, UTF-8)
- Rob Pike (Unix, UTF-8)
- Robert Griesemer (Hotspot, JVM)
- And a few other engineers at Google



Go 사용하는 곳

Clever

bitly

DigitalOcean

APCERA®

rawstream

STAT HAT

Bitbucket



Couchbase

Iron.io

aruba®
NETWORKS
an HP company

Booking.com

BBC
Worldwide

Beachfront
Media

B

shopify

MOOVWEB



Bigcommerce



Bowery

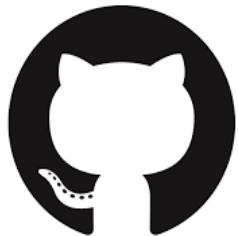
CLOUDFLARE

avocet



Go 사용하는 곳

Google



YouTube



Dropbox

heroku

gettyimages

The Economist

심지어 Azure도..

Microsoft Azure SDK for Go

This project provides various Go packages to perform operations on Microsoft Azure REST APIs.

[godoc](#) [reference](#) [build](#) passing

See list of implemented API clients [here](#).

NOTE: This repository is under heavy ongoing development and is likely to break over time. We currently do not have any releases yet. If you are planning to use the repository, please consider vendoring the packages in your project and update them when a stable tag is out.

Installation

```
go get -d github.com/Azure/azure-sdk-for-go/management
```

Usage

Read Godoc of the repository at: <http://godoc.org/github.com/Azure/azure-sdk-for-go/>

The client currently supports authentication to the Service Management API with certificates or Azure `.publishSettings` file. You can download the `.publishSettings` file for your subscriptions [here](#).



Go 언어의 목적은?

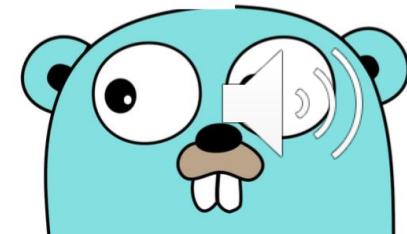
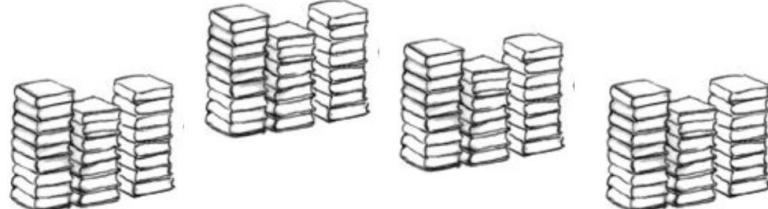
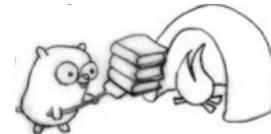
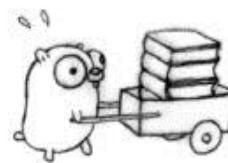
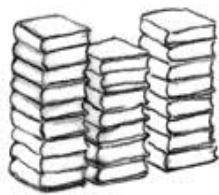
Concurrency!!



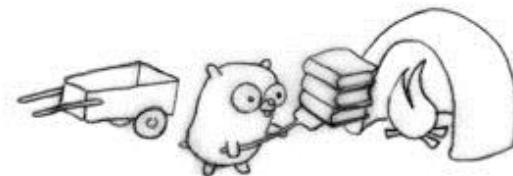
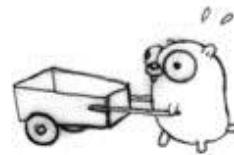
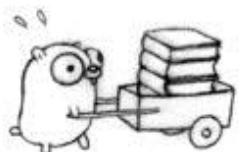
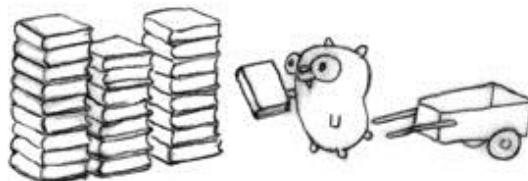
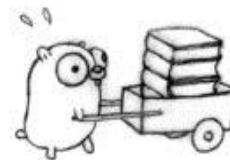
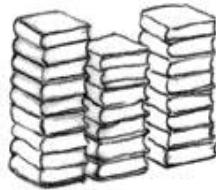
Concurrency 와 Parallelism



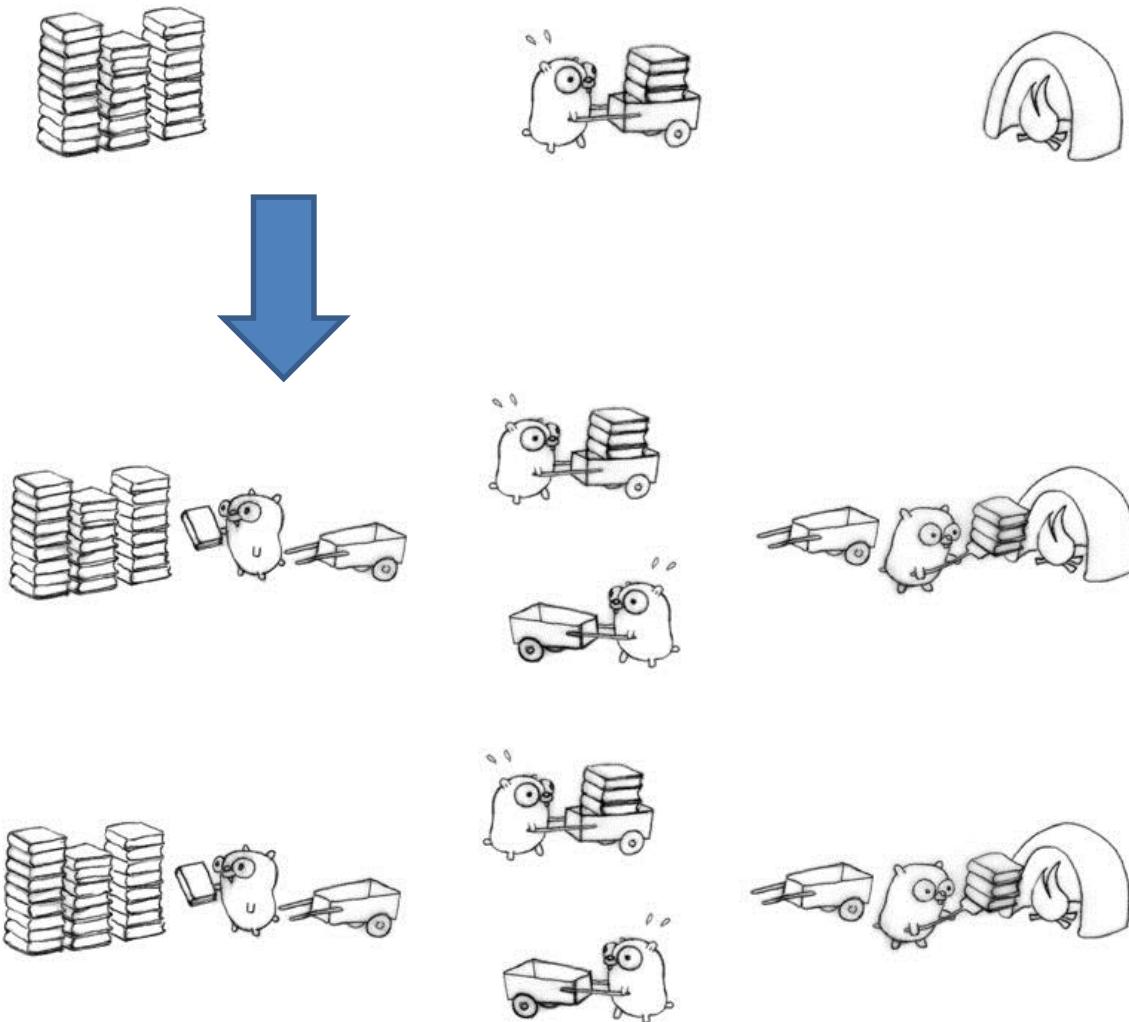
Parallelism



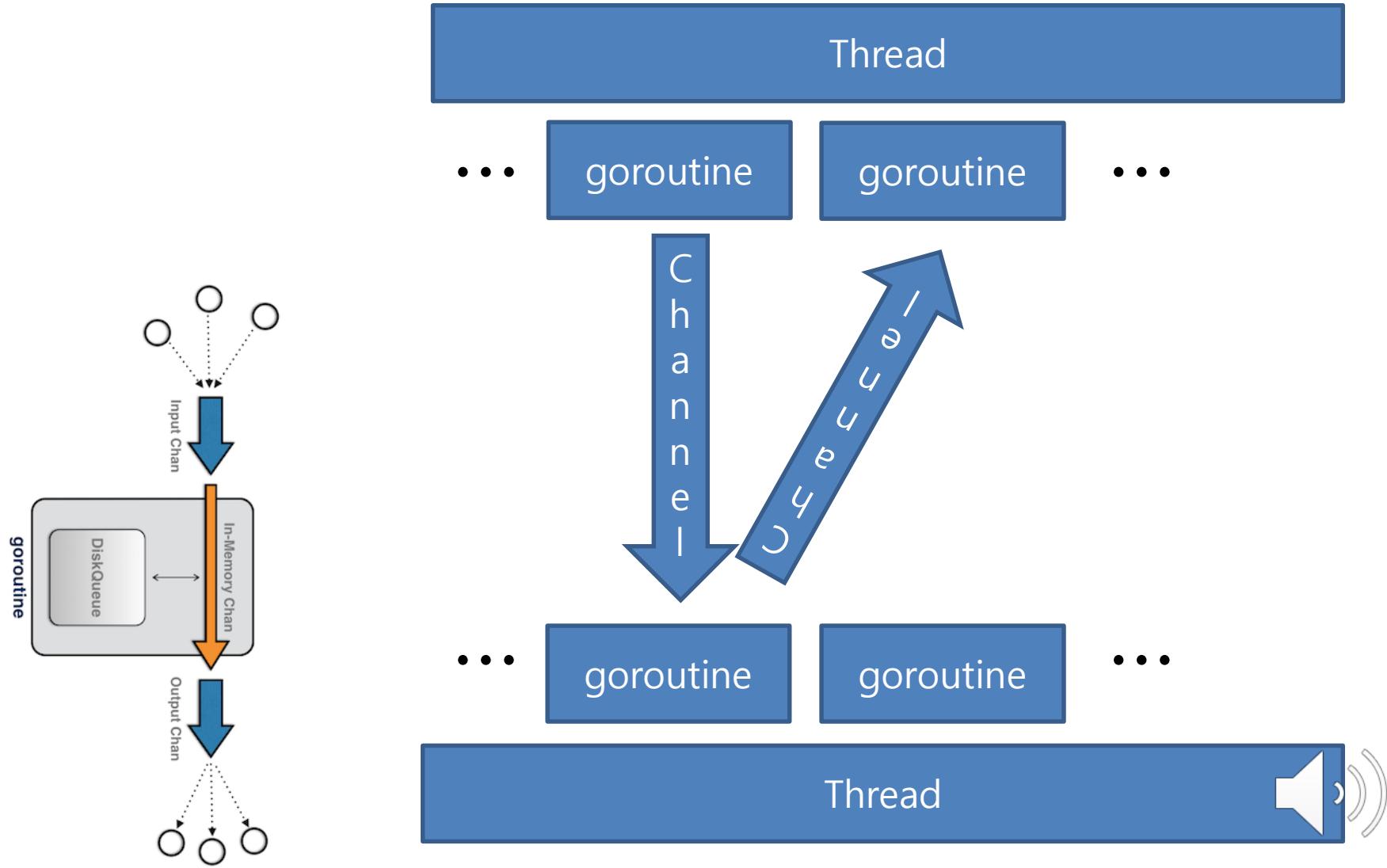
Concurrency



Parallelism + Concurrency



Go의 핵심 컴포넌트

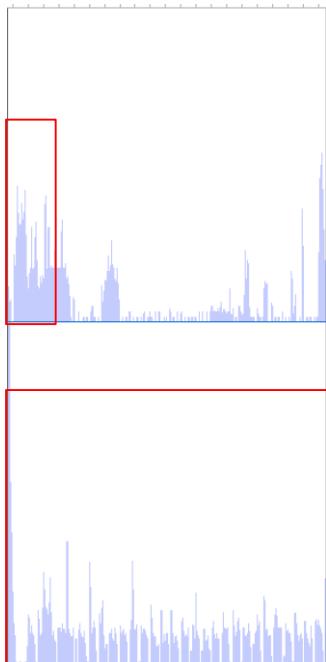


Go와 Java Thread 성능 차이

Hello 10만개 출력하기

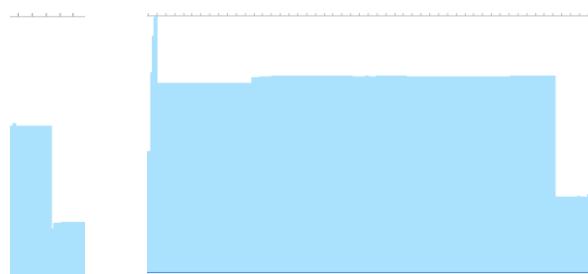
<http://www.slideshare.net/seungkyupark/go-langgoroutine-channel-56542876>

<CPU>

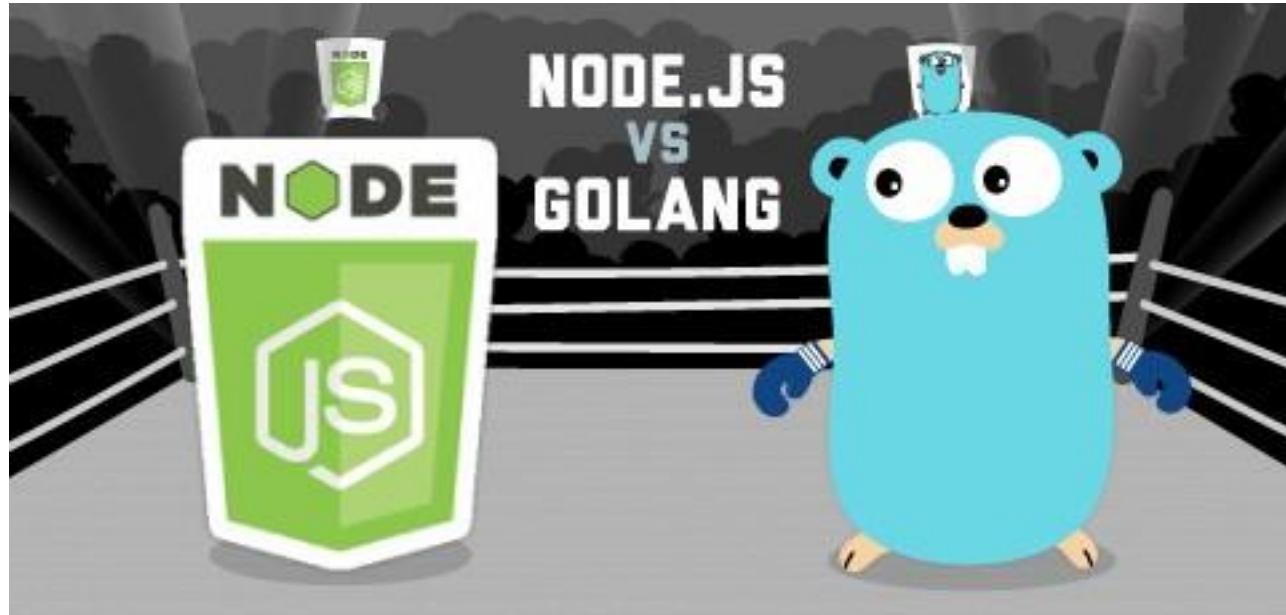


	Thread(Java)	Goroutine
개수	100000	100000
처리 속도	5.902s	226.750829ms

<Memory>



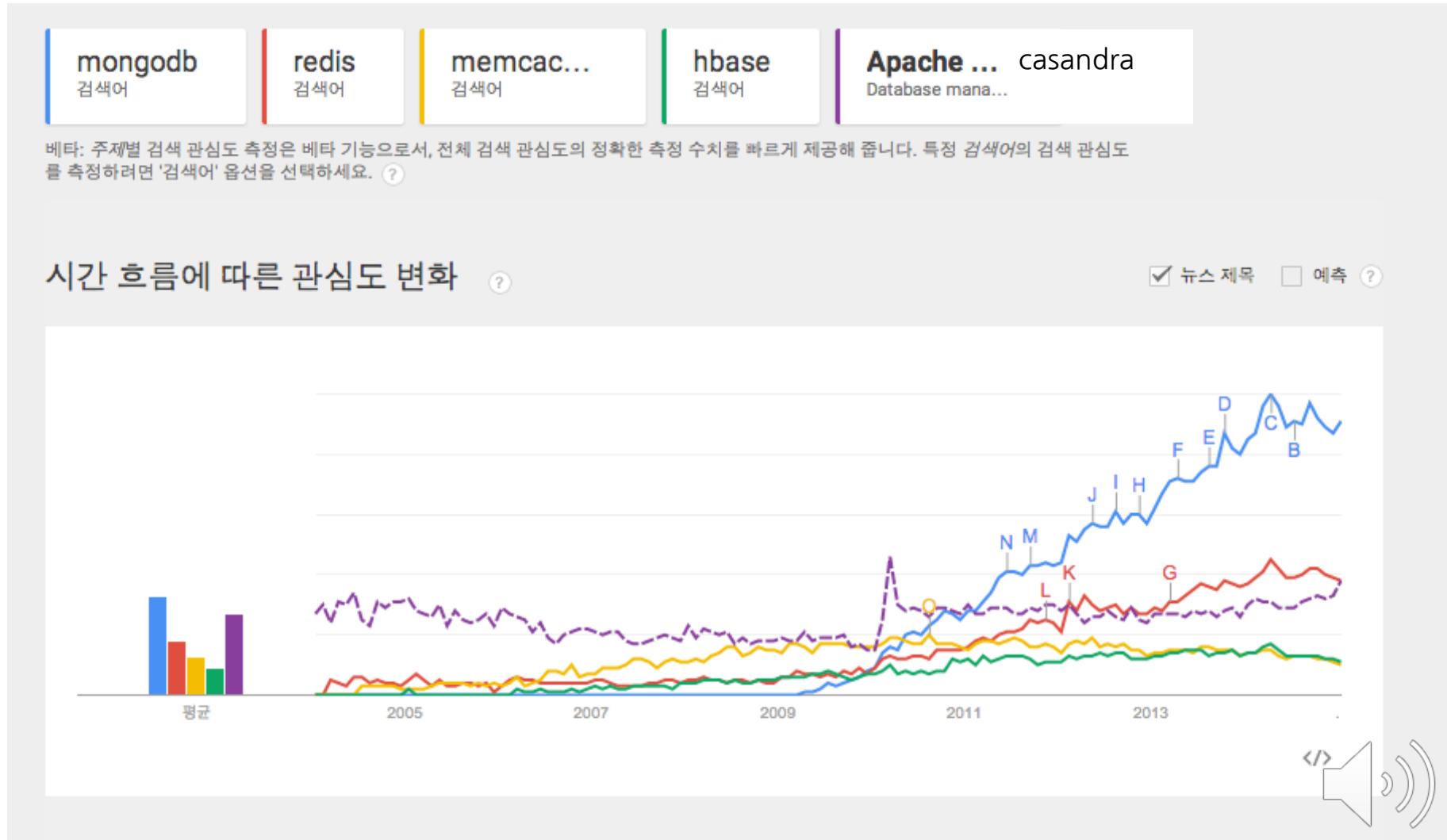
한동안 node.js 와 golang의 조합이 많이 애용될듯..

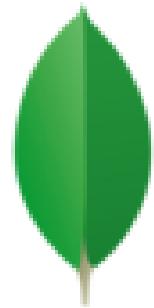


서비스 구성



가장 많이 사용되는 DB는 (mongodb , redis..)





mongoDB

10gen이라는 회사에서 만들었습니다.

클라우드 서비스를 SaaS로 제공하고 싶었는데 ~~망했고~~
사람들은 그 중에서 DB만 쓰고 싶어했습니다.

그래서 남은 DB가 mongoDB가 되었습니다.



SQL Terms/Concepts

database

table

row

column

index

table joins

primary key

Specify any unique column or column combination as primary key.

aggregation (e.g. group by)

MongoDB Terms/Concepts

database

collection

document or BSON document

field

index

embedded documents and linking

primary key

In MongoDB, the primary key is automatically set to the `_id` field.

aggregation pipeline

See the [SQL to Aggregation Mapping Chart](#).



mongoDB의 특징

- document data model
- 스키마가 없다 (삽입시 생성된다..)
- json / bson
- 애드혹 쿼리 및 세컨더리 인덱스 지원
- 복제 / 색인 지원
- 하나의 마스터가 존재 cf. Casandra
- 일부 원자적 연산 지원 (트랜잭션 X)



Document Data Model

```
{ _id: ObjectId('5d41402abc4b2a76b9719d911017c592'),  
    title: '실종 슈퍼마리오 4-1 리커버 발견, 몸안의 초록 버섯도 일치',  
    url: 'http://mario.game/0401',  
    author: 'kuppa',  
    vote_count: 7777,  
    tags: ['마리오','쿠파', '만우절', '사망'],  
    image: {  
        url: 'http://naver.com/logo1.png',  
        caption: '로고',  
        width: 480  
    },  
    comment: [  
        { user: '루이지',  
            text: '마리오 본인도 인정, 논란 그만했으면..' },  
        {user: '모그레인',  
            text: '일어나라, 나의 쿠파여~'  
        }  
    ]  
}
```



Document

도큐먼트는 { 로 시작하고 } 로 끝난다.

속성의 이름 : 값의 집합을 저장한다.

속성은 , 로 분리한다.

값에는 배열(리스트)가 올 수 있다.

값에는 다른 도큐먼트가 올 수 있다.



JSON

모든 도큐먼트는 JSON 형식으로 표현됨



JSON: JavaScript Object Notation.

JSON is a **syntax** for storing and exchanging data.

JSON is an **easier to use** alternative to XML.

This JSON syntax defines an employees object, with an array of 3 employee records (objects):

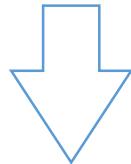
JSON Example

```
{"employees": [
    {"firstName": "John", "lastName": "Doe"},
    {"firstName": "Anna", "lastName": "Smith"},
    {"firstName": "Peter", "lastName": "Jones"}
]}
```



내부적으로 저장될 때는 BSON으로 됩니다.

```
{“hello”: “world”}
```



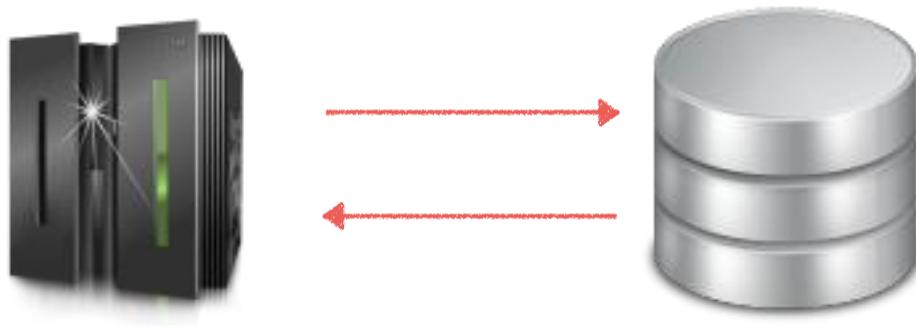
```
\x16\x00\x00\x00\x02hello
```

```
\x00\x06\x00\x00\x00world
```

```
\x00\x00
```



일반적인 DB구성

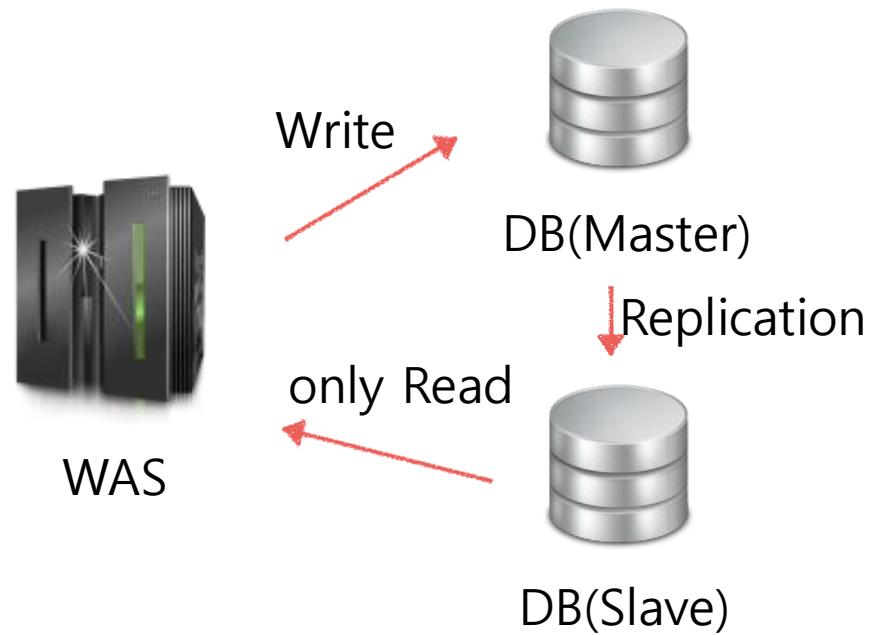


WebServer, WAS

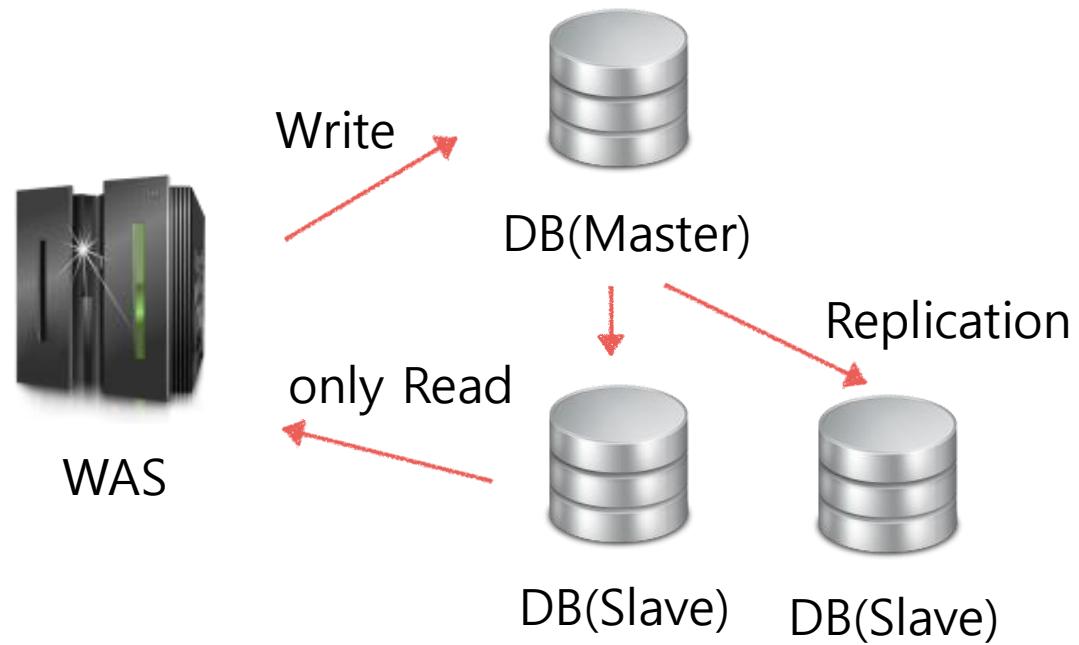
DB(Master)



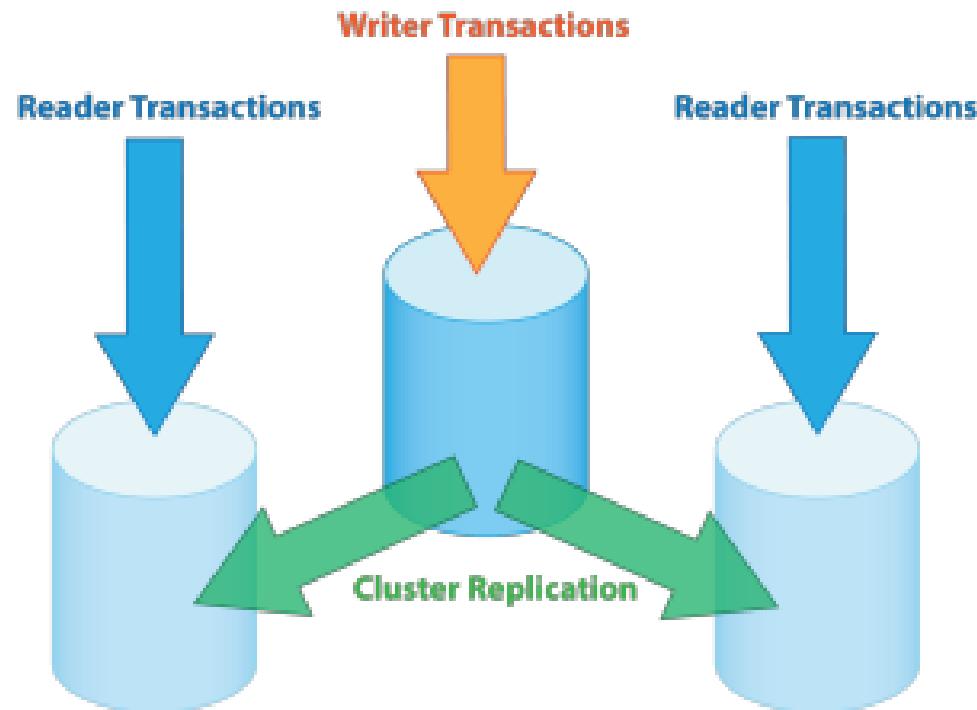
Write/Read를 분산



Write/Read를 분산 구성

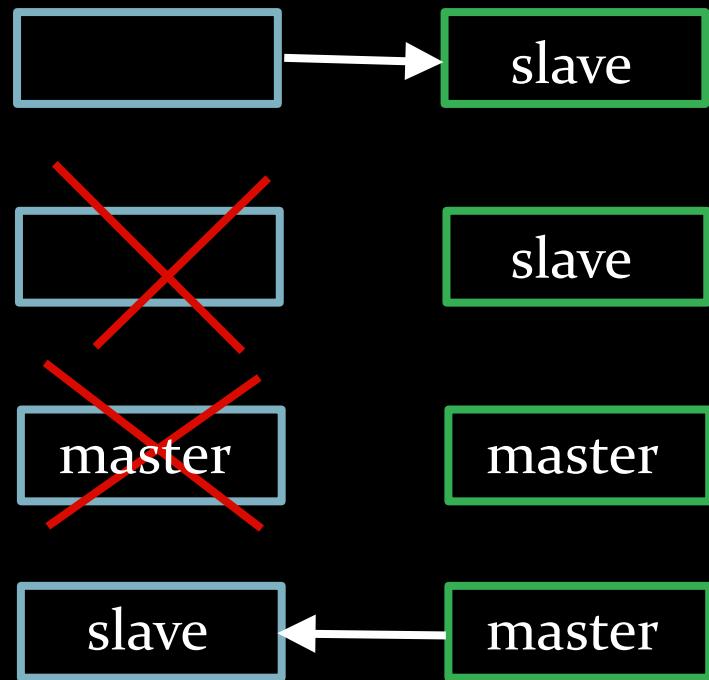
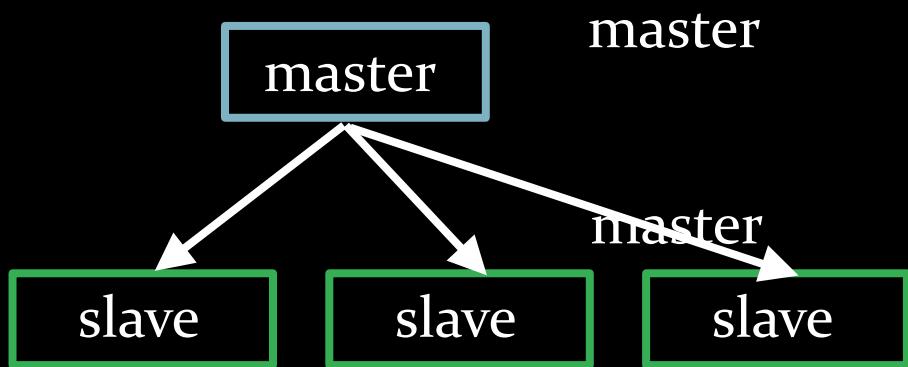


Master-Slave Replication



Mongo Replication 01

매우 쉽다



데이터 베이스 분할..

수직 분할

- 팩트 테이블의 기본키가 중복
- 전체적인 공간은 더 많이 차지
- 특성에 따른 컬럼 또는 자주 사용하지 않는 컬럼을 분할함으로써 크기를 줄이고 검색 비용이 줄어 듈다

기간키	매장키	제품키	매출액	매출원가	운송비
20120101	110	1101	27,800	12,100	780
20120301	120	1200	39,000	15,500	810
20120401	210	100	27,800	13,000	810
20120405	120	1200	39,000	14,500	930
20120508	210	1301	30,300	14,300	840

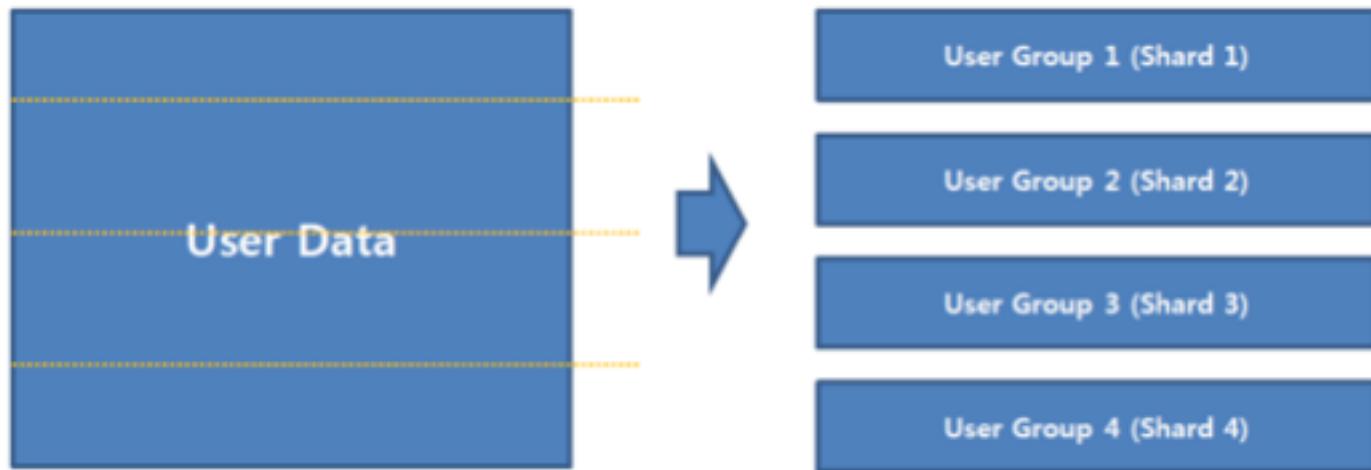
수평 분할

- 하나 이상의 비즈니스 차원에 따라 행(ROW) 수준에서 팩트 테이블을 분할.
- 분할 시 안정적인 애트리뷰트(기간, 원, 분기)로 분할.



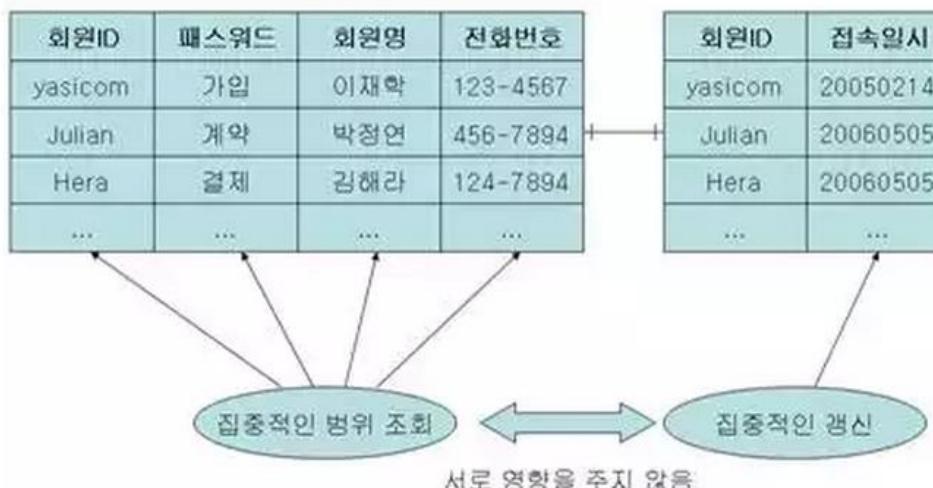
데이터 베이스 분할..

- 수평 분할(horizontal partitioning = Sharding)은 하나의 테이블의 각 행을 다른 테이블에 분산시키는 것이다. 예를 들어 방대한 고객 데이터 테이블을 성별에 따라 '남녀'로 나누어 CustomerMen과 CustomerWomen 두 개의 테이블로 분할한다. 테이블은 2개로 분할되지만, 모든 고객을 나타내기 위해 양자를 결합한 뷰를 생성하면 된다.

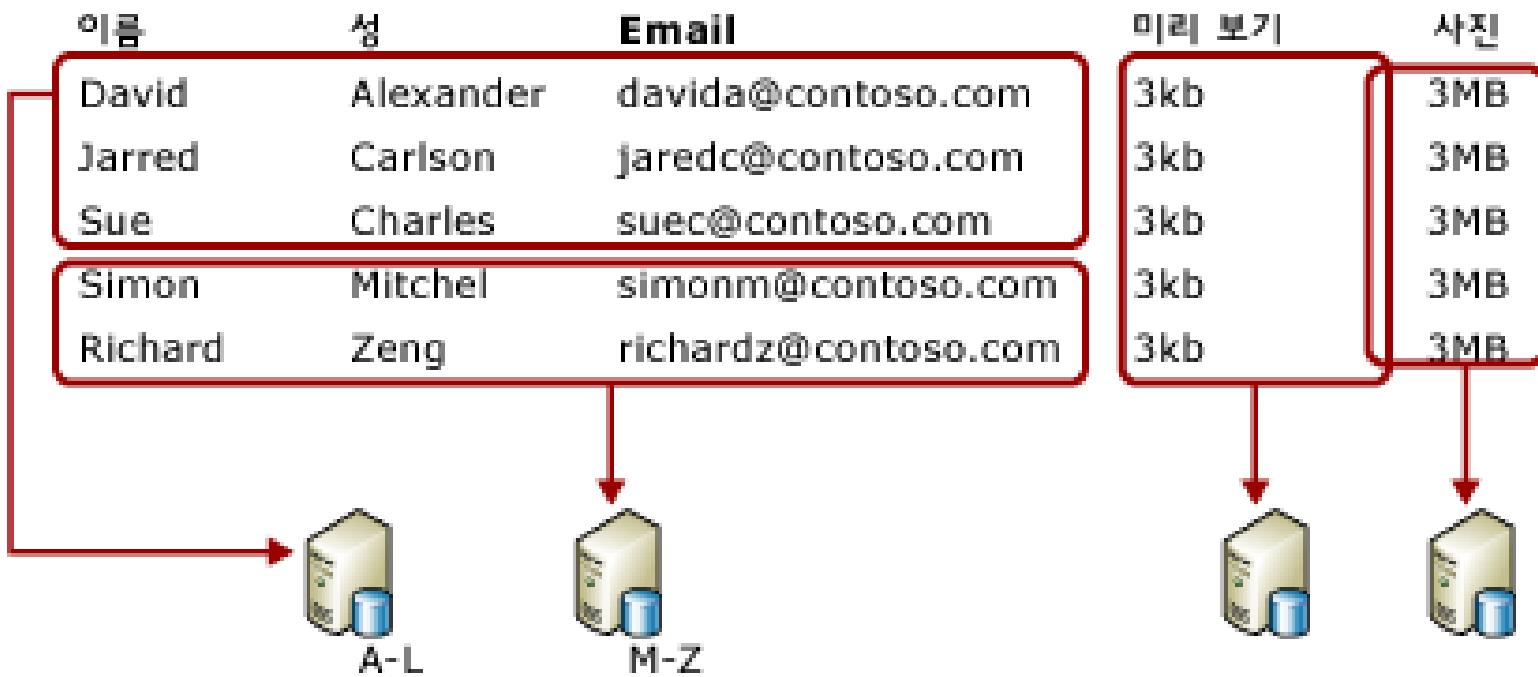


데이터 베이스 분할..

- 수직 분할(vertical partitioning)은 테이블의 일부 열을 빼내는 형태로 분할한다. 관계의 정규화는 본질적으로 수직 분할에 관련된 과정이다.
- 사용 빈도가 높은 데이터에만 액세스할 경우 성능이 향상된다. 예를 들어, 뉴스를 서비스할 때, 고객들은 최근의 데이터를 가장 많이 조회할 것이다. 이 경우 1개월 전의 데이터를 다른 테이블에 두면, 훨씬 효율적으로 검색할 수 있을 것이다.

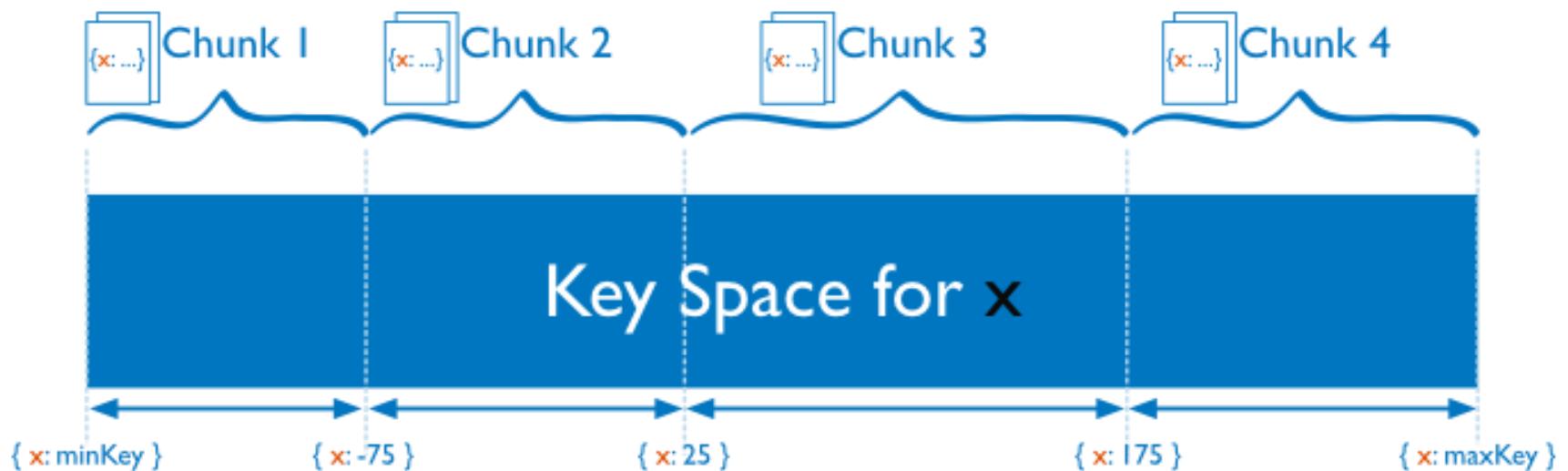


실제 어떻게 쓰나?



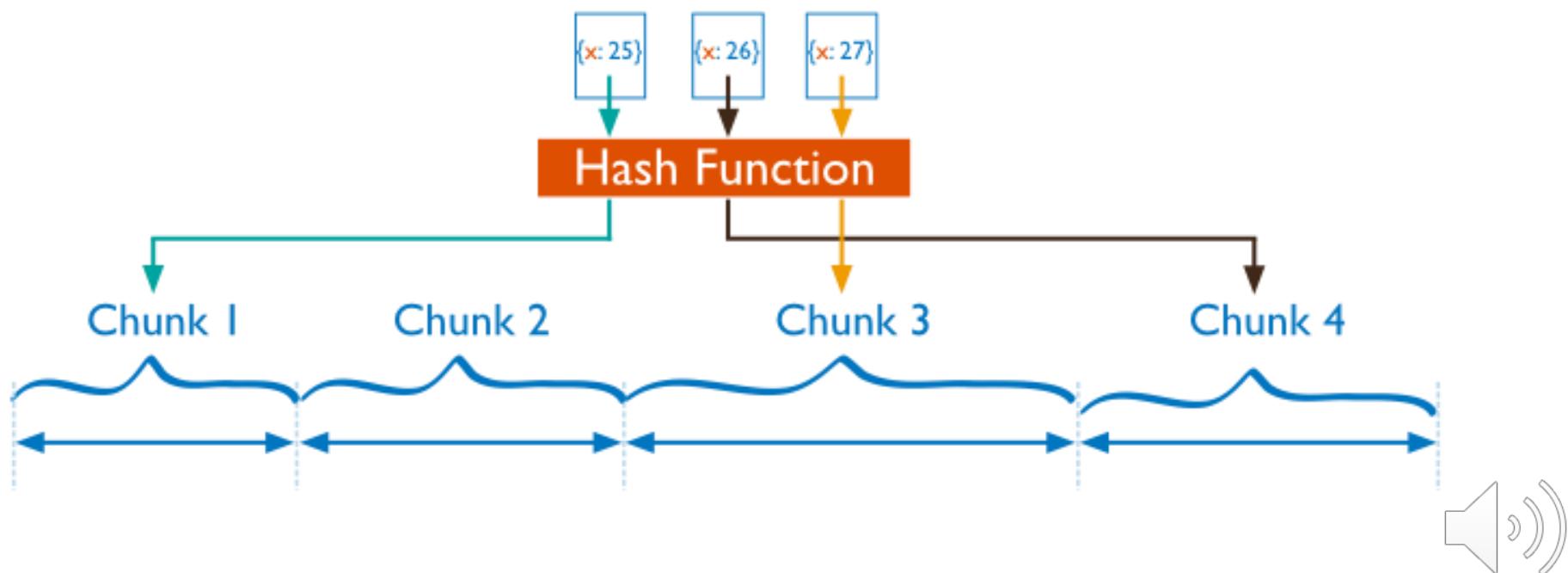
Sharding

- Range based partitioning



Sharding

- Hash based partitionning

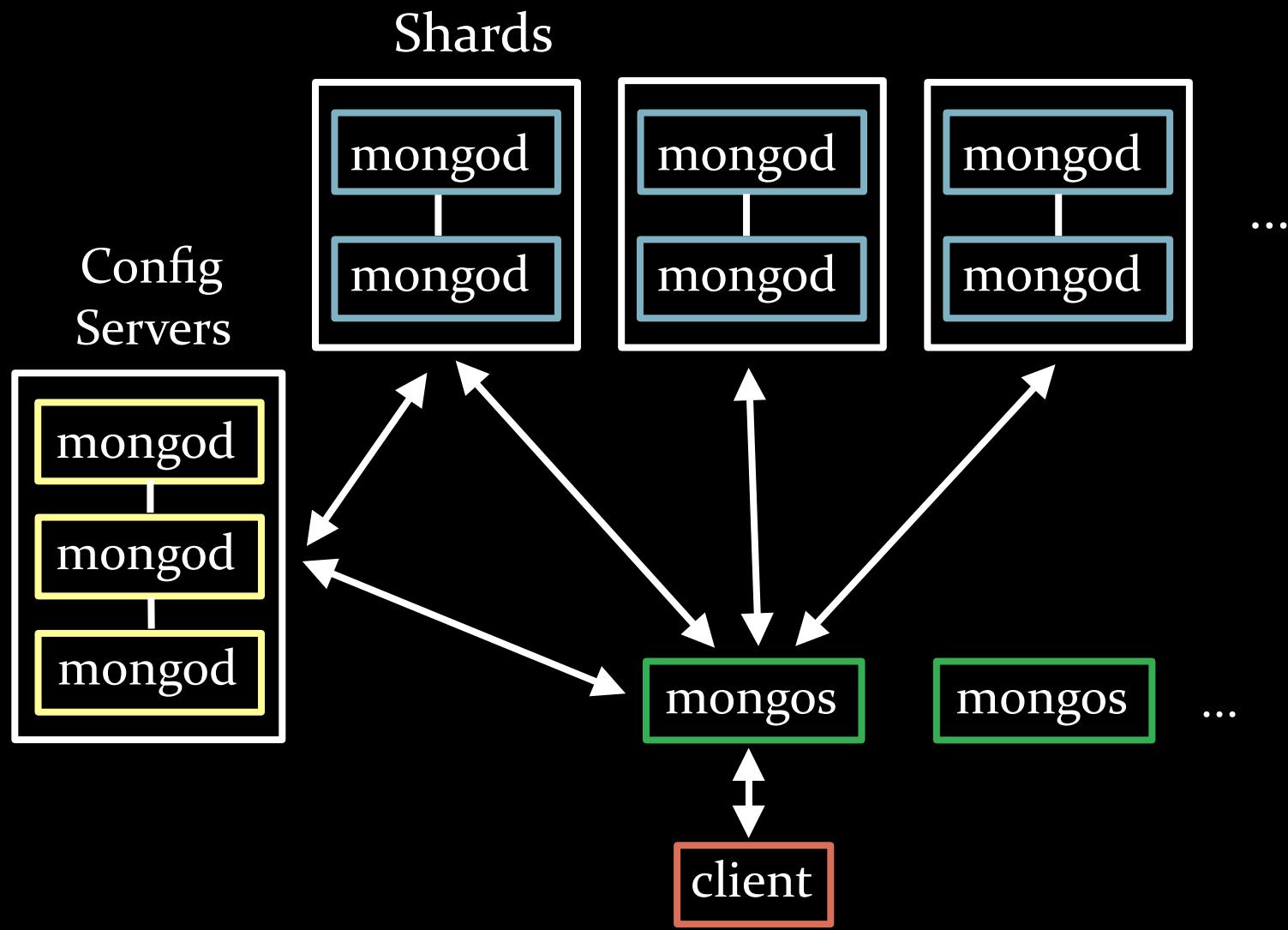


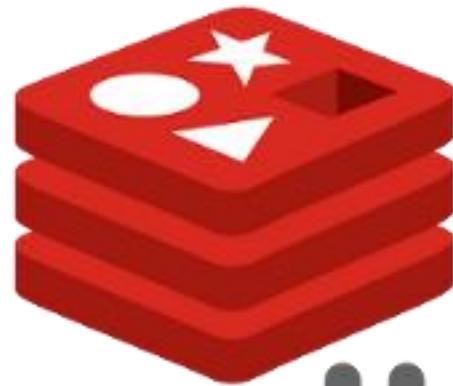
Sharding

- Ranged + Hash based partitionning



Mongo Auto-sharding 지원





redis



Redis란?

- Remote Dictionary Server의 출임말
- Key–Value Cache and Store
- Persistence을 지원하는 In-Memory Dataset
 - 다양한 자료구조 지원(List, Hash, Set, Sorted Set, bitmaps and hyperloglogs)
- ANSI C로 작성
- Open Source(BSD licensed)
- Single Thread
- Transactions, Pub/Sub, Lua scripting 등 지원

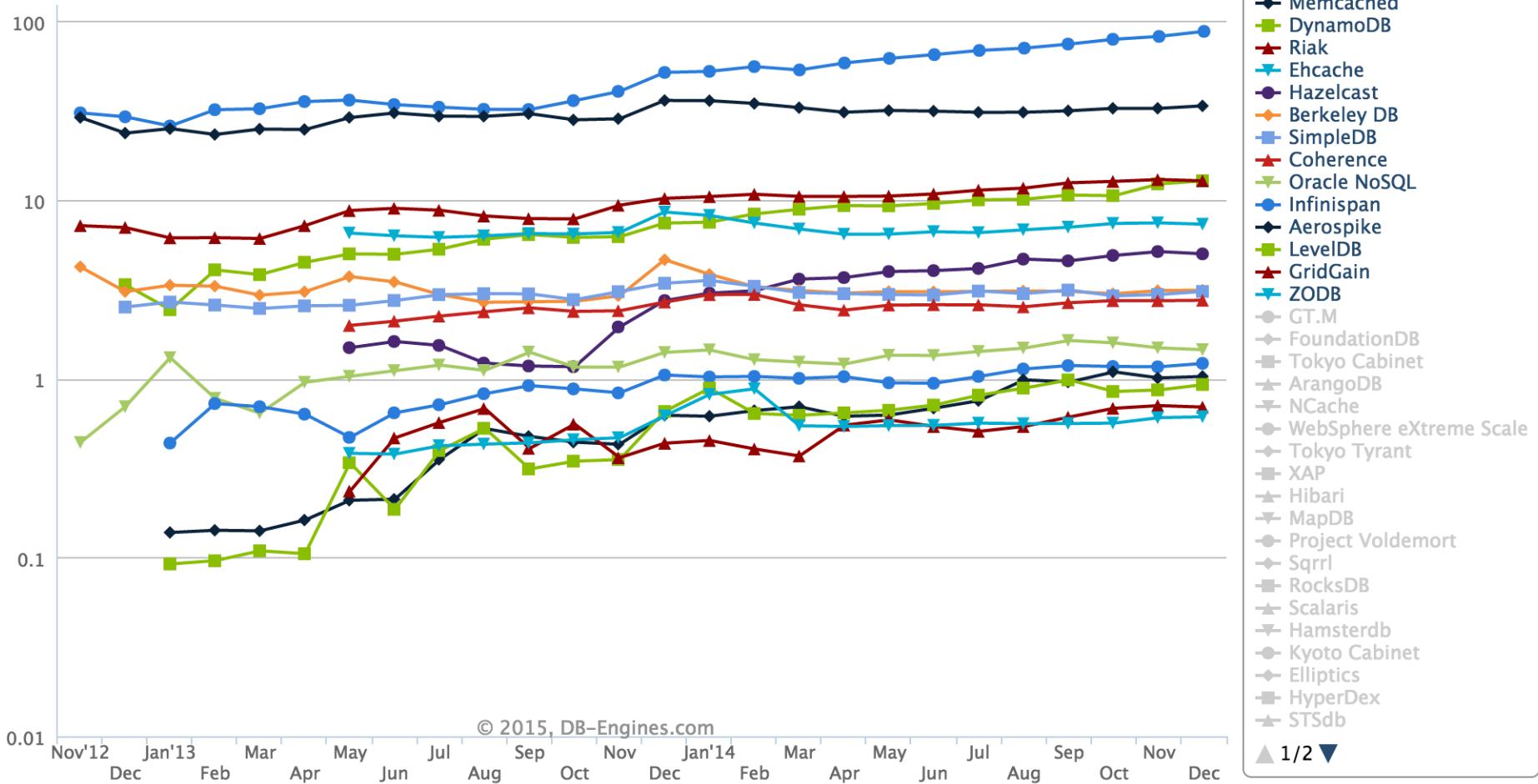


다른 NoSQL 비교

	Redis	Memcached	MongoDB
In-memory	✓	✓	
Persistent	✓		✓
Key-value store	✓	✓	
Supports more than strings	✓		✓
Multithreaded		✓	✓
Supports larger-than-memory dataset			✓
As fast as	Memory	Memory	Disk



DB-Engines Ranking of Key-value Stores

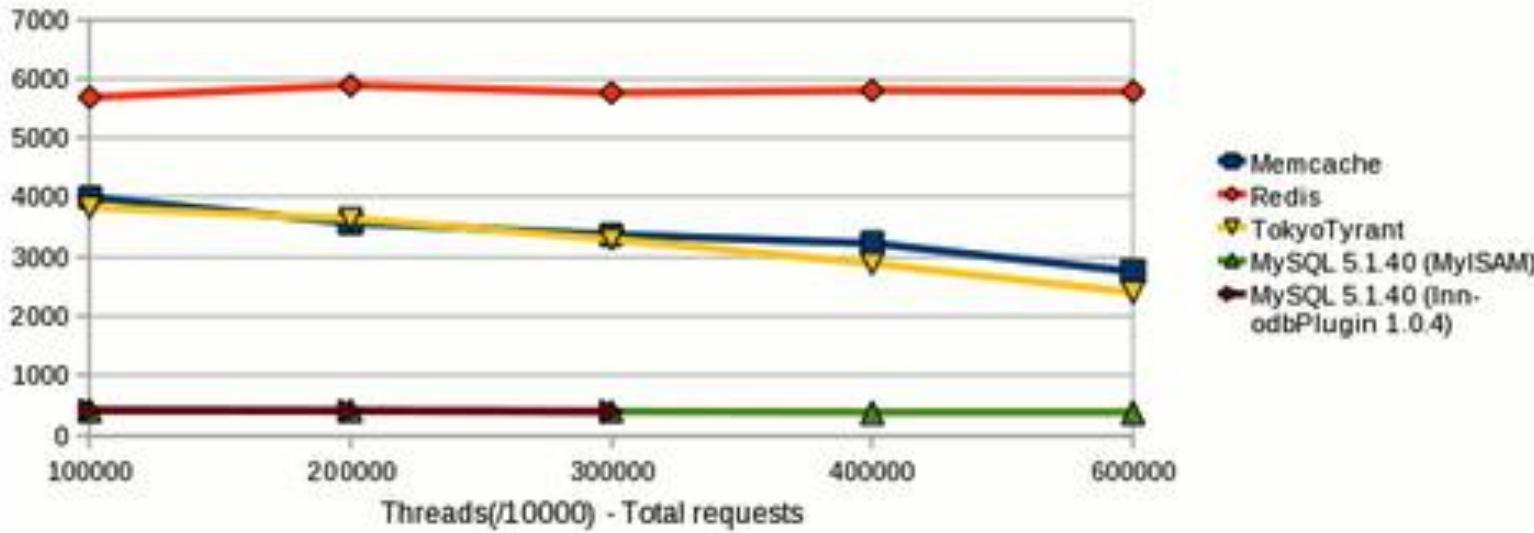


© 2015, DB-Engines.com

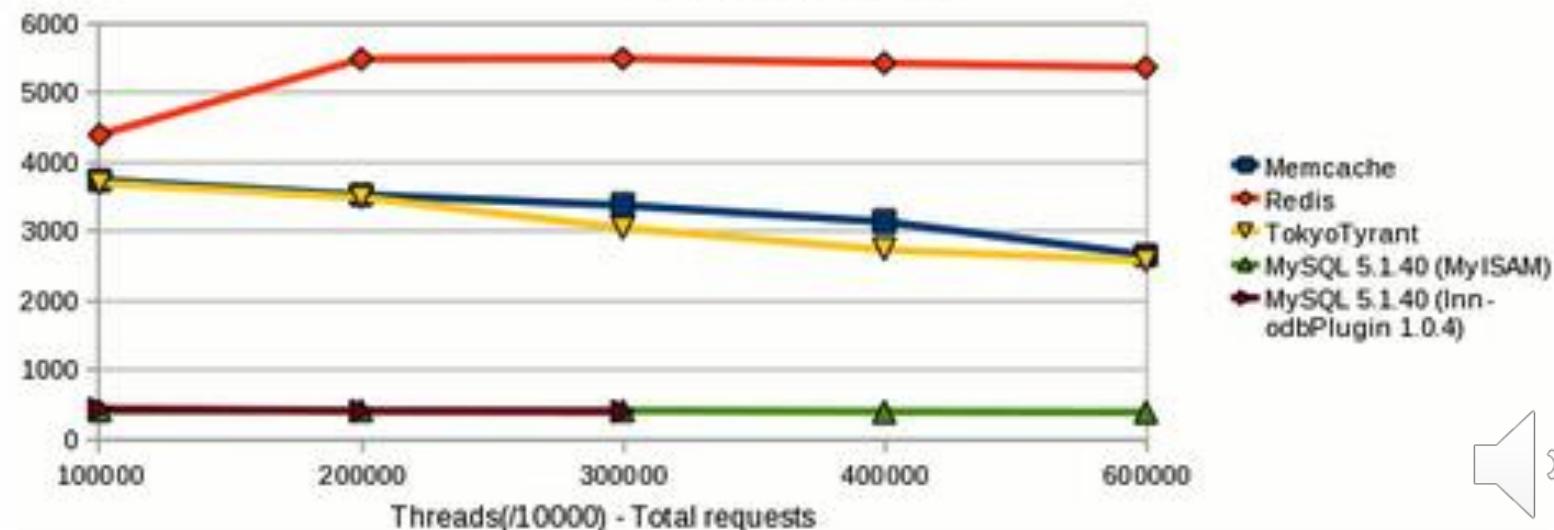


성능 비교

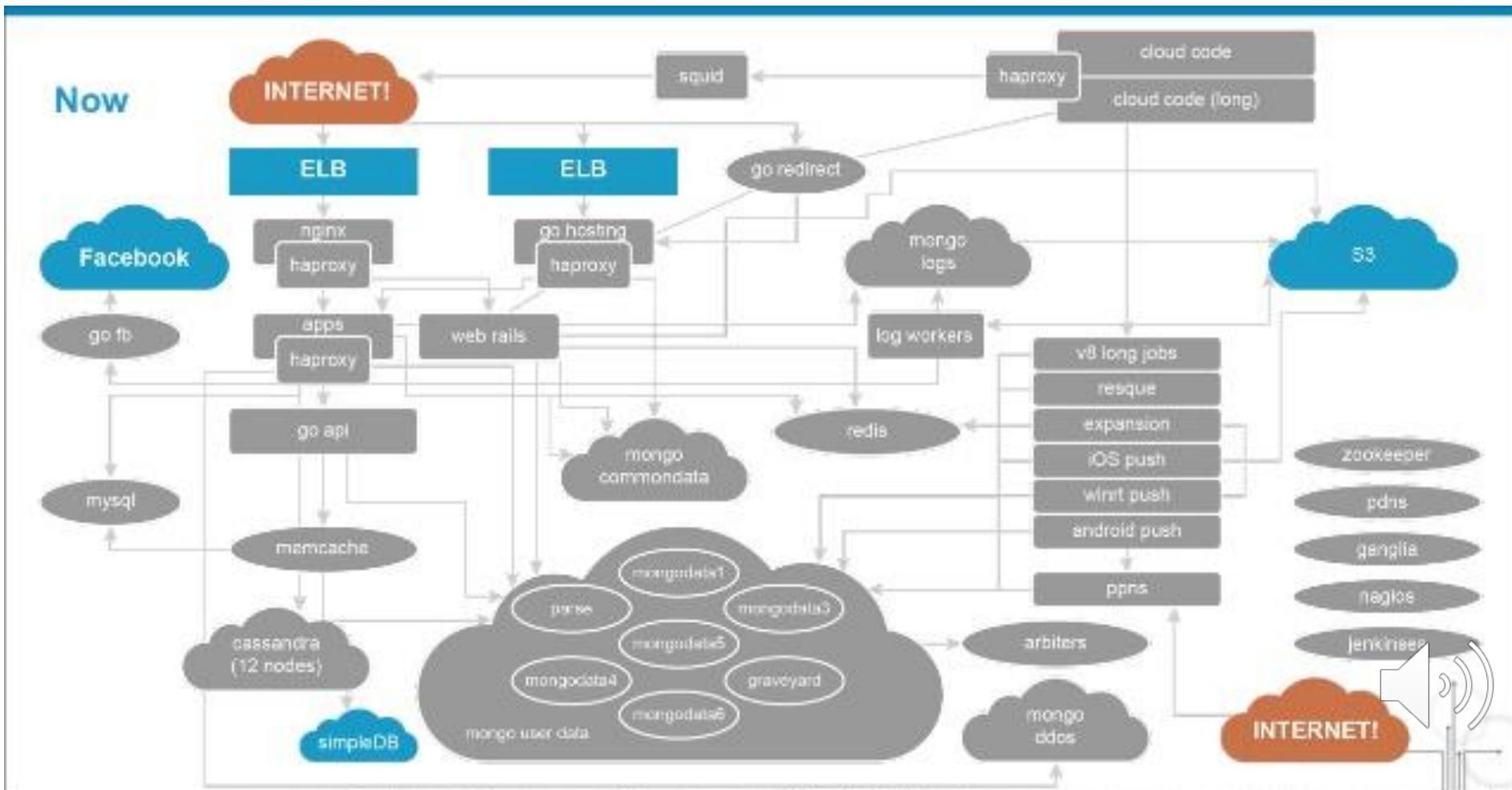
Throughput (SET)



Throughput (GET)



엄청나게 많은 리소스 누가 관리?



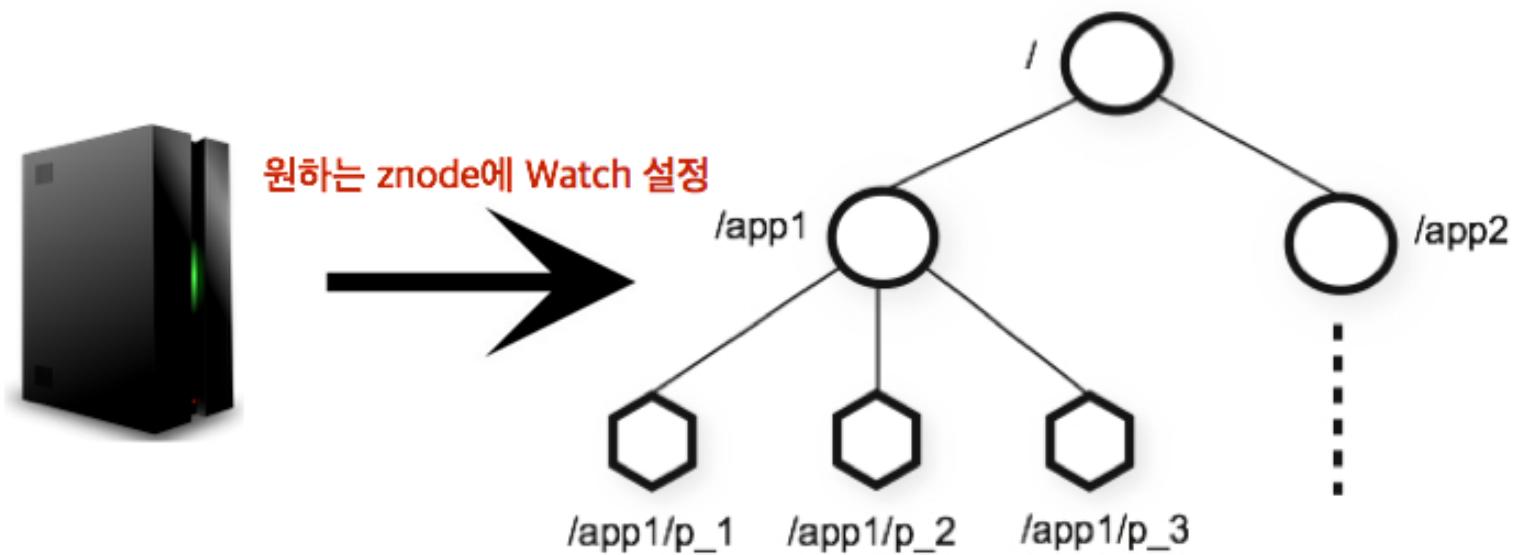
ZooKeeper

- 분산 처리 환경에 사용할수 있는 코오디네이터



ZooKeeper Watch

- Watch 기능은 Zookeeper 클라이언트가 특정 znode에 Watch를 설정해 놓아야 사용할 수 있습니다.



ZooKeeper가 감지하는 이벤트

- Watcher가 감지하는 znode의 '변경' event는 4가지 종류가 있습니다.
 - **NODE_CREATED** : 노드가 생성 됨을 감지.
 - **NODE_DELETED** : 노드가 삭제 됨을 감지.
 - **NODE_DATA_CHANGED** : 노드의 데이터가 변경 됨을 감지.
 - **NODE_CHILDREN_CHANGED** : 자식 노드가 변경 됨을 감지.



ZooKeeper의 단점?

- 일회성 Trigger..
- 해당 Watcher가 삭제되면, 다시 Watcher 를 만들고 걸어줘야 한다..
- 프로그래밍 스타일이.. 번거롭게 된다.



etcd

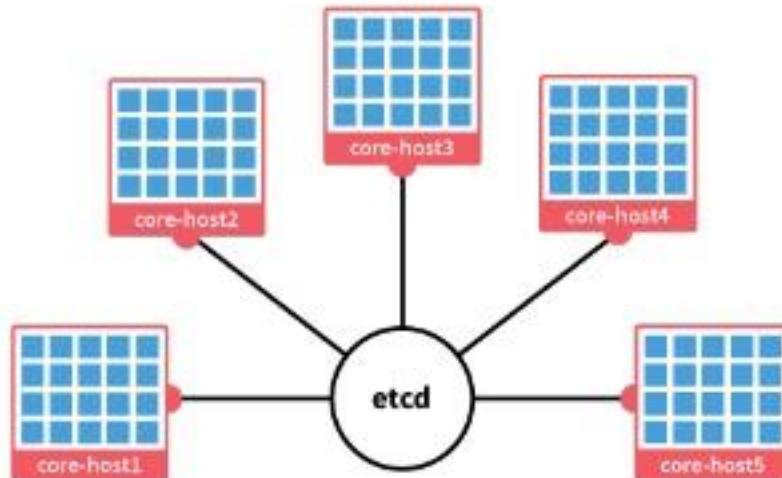
- 빠르고 간편한 프로그래밍 모델을 제시



ETCD



- Distributed key/value store
- Like a directory tree
- JSON/REST API
- Uses a Discovery URL



images: coreos.com



관련 샘플 코드

- Mongodb zookeeper 핸들링 코드

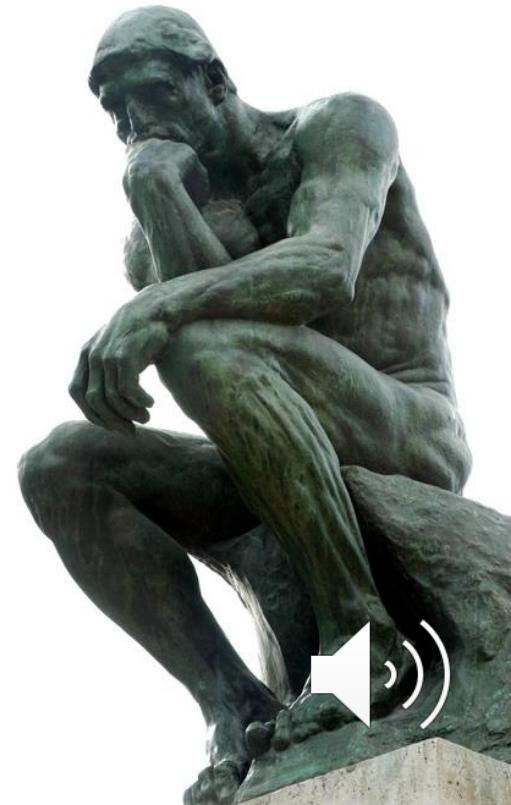
<https://github.com/swmaestro6th-crashreport/nodejs-mongodb-zookeeper>

- Mongodb etcd High 핸들링 코드

<https://github.com/swmaestro6th-crashreport/nodejs-etcd-mongodb>



6. 오픈소스 = 클라우드 시대 고민해야 하는 문제들..



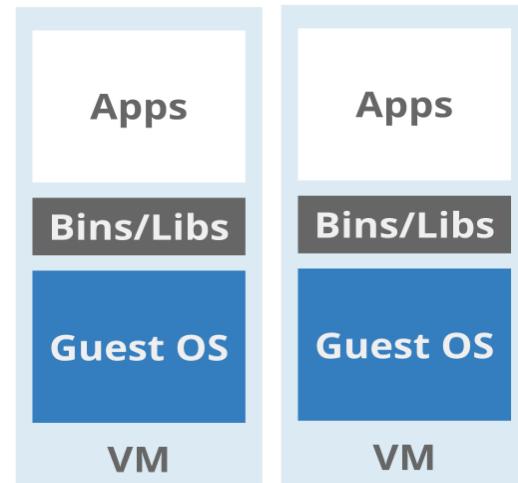


와 Cloud로 환경 변화가 필요하지?

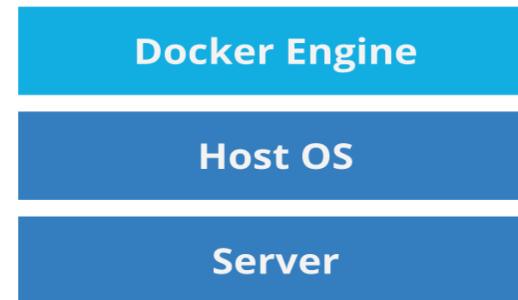
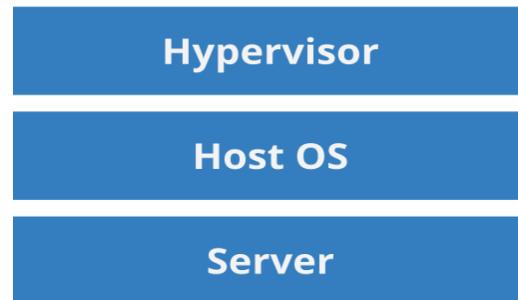
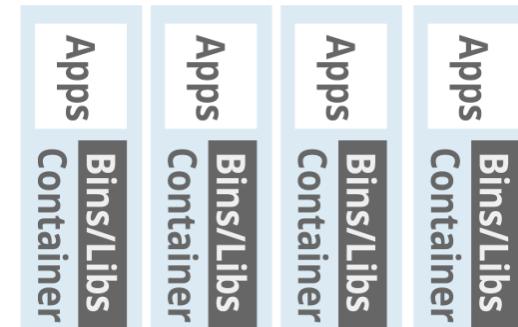


이걸까요드는 ...

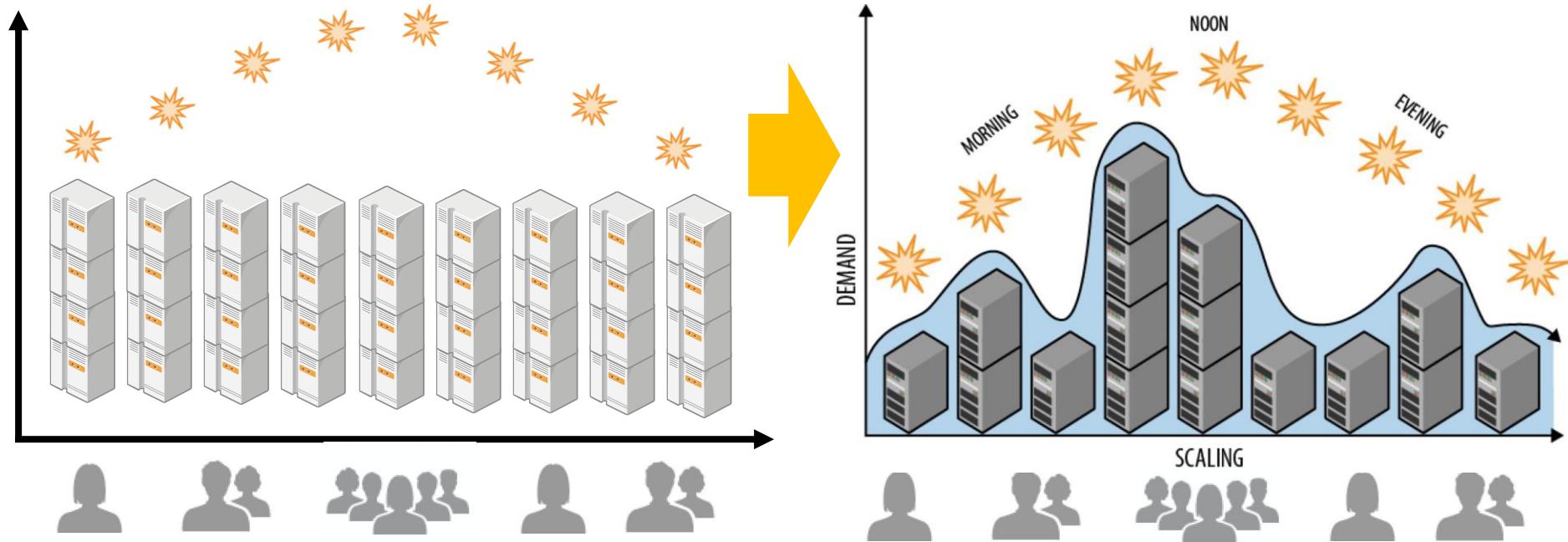
Virtual Machine



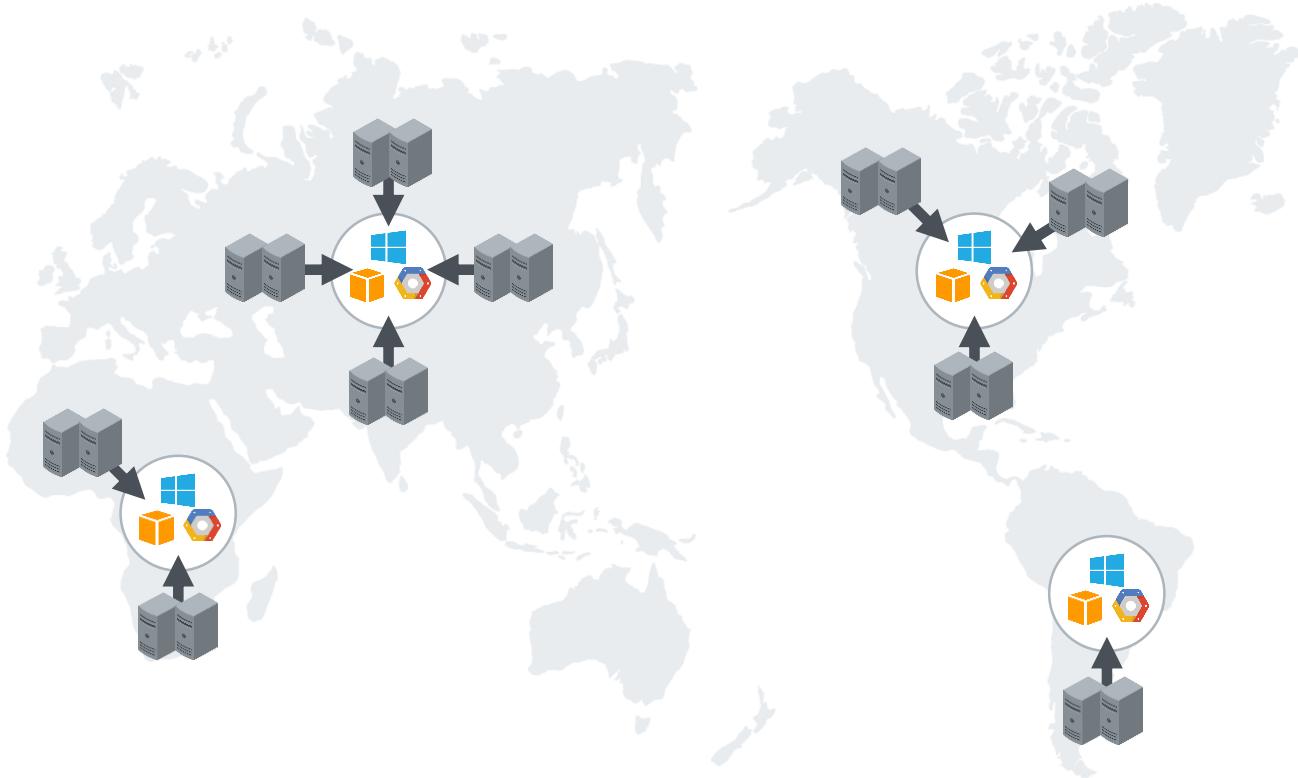
Docker Container



비용 효율 & 초기 도입 비용 - 거짓말??

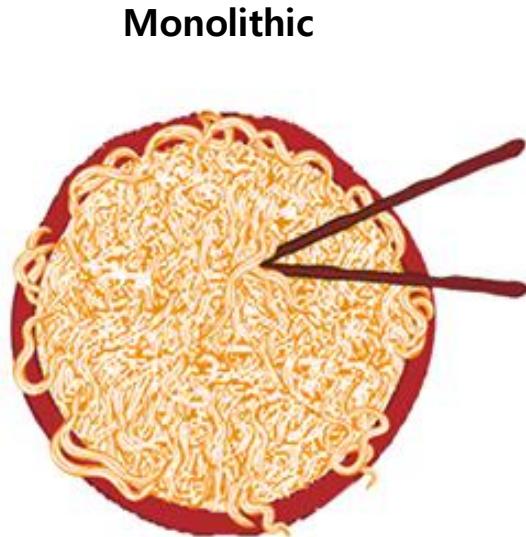


갑자기 성장 (멀티 리전을 사용)해야 되는 서비스



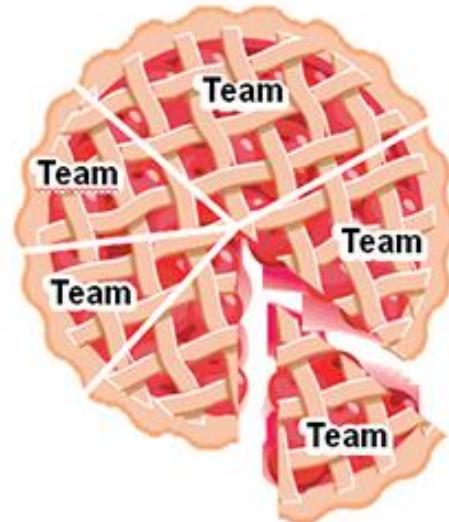
Agility를 확보하기 위해 MSA를 도입하는 게 한다면..

~ 1990s



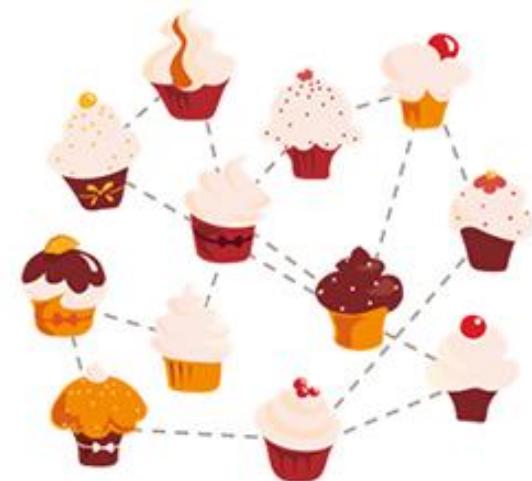
2000s

Looser Coupling

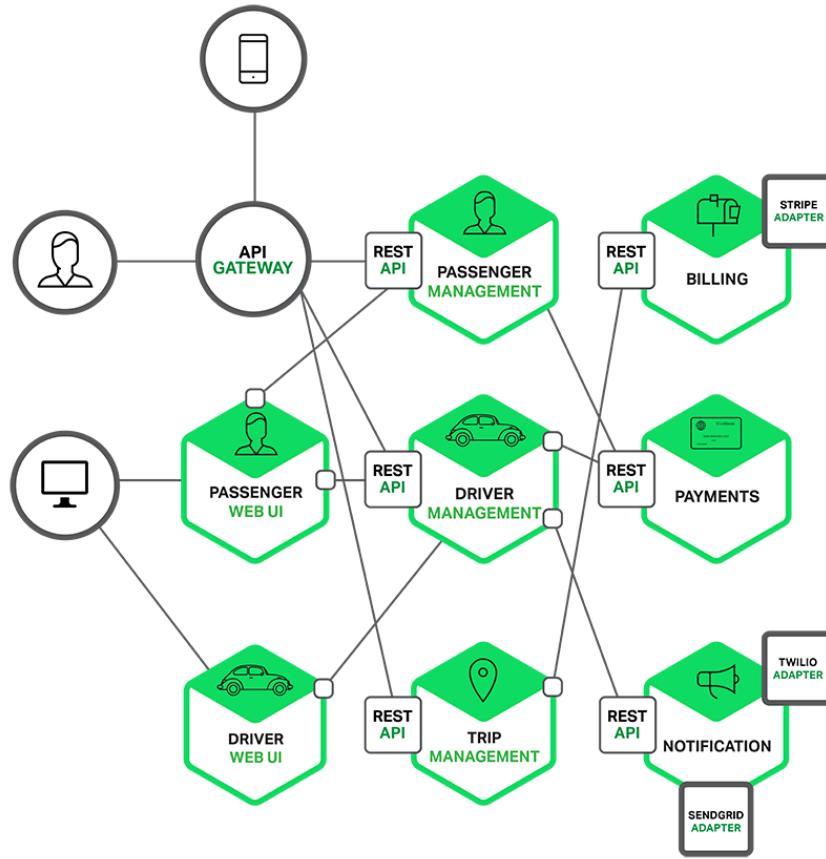


2010s

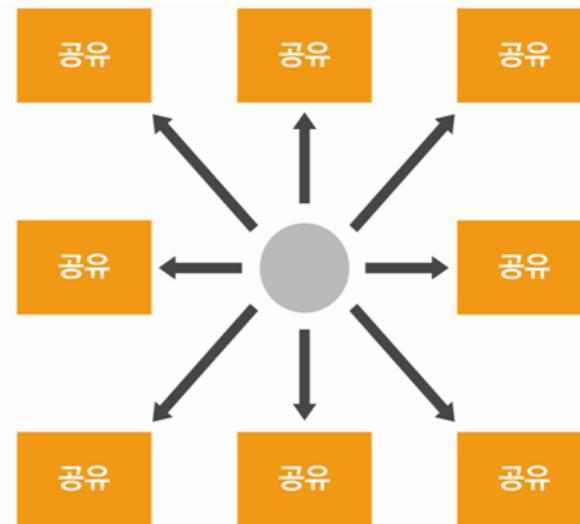
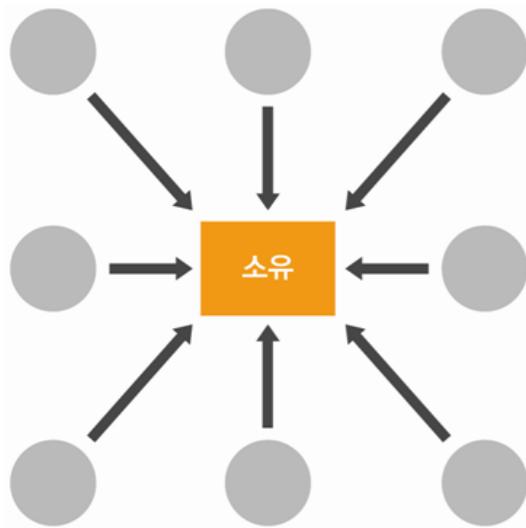
Micro Service Architecture



Micro Service Architecture..



클라우드는 공유 지원..



클라우드는 공유 가원..
요청이 많으면 즉 내 차례를 기다리거나..





클라우드는 공유 지원..

상황에 맞게 제한된 지원을 나눠가지
던가..

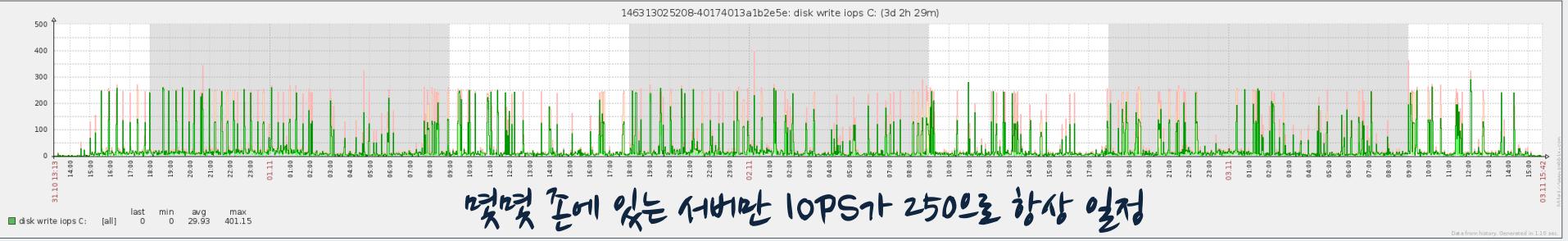


잘 동작하던
게임서비스에서

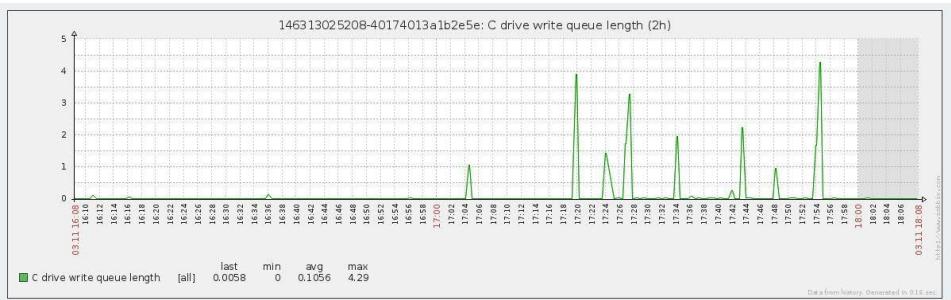
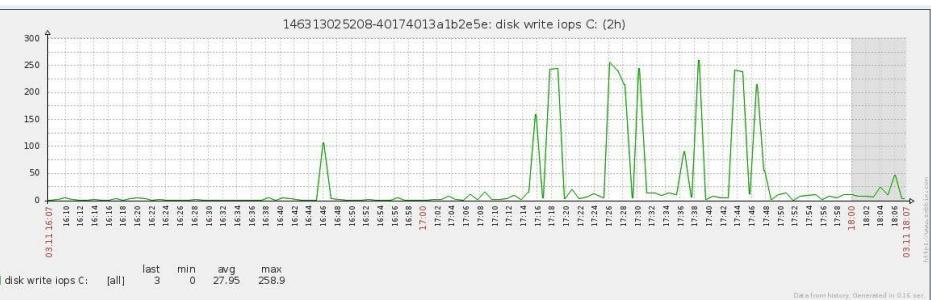
User들이
갑자기 터거나간 사이.



알고보니 차원 공유 계약..



몇몇 존에 있는 서버만 IOPS가 250으로 항상 일정



IOPS가 높을 때마다, Disk Queue Length가 높은 수치로 증가 → 클라우드에선 늘 발생할 수 있는 문제..



하나로 채..

만약 AM (AM?, AZ?) 제품이면 제한 주고 IOPS 보장 제품 구매

	Standard Volumes	Provisioned IOPS Volumes
Optimized for	Workloads with low or moderate IOPS needs and occasional bursts.	Transactional workloads requiring consistent IOPS.
Volume Attributes	Up to 1 TB, average 100 IOPS per volume. Best effort performance. Can be striped together for larger size and higher IOPS.	Up to 1TB, 2,000 IOPS per volume. Consistent IOPS. Can be striped together for larger size and higher IOPS.
Workloads	File server, Log processing, Websites, Analytics, Boot, etc.	Business applications, MongoDB, SQL server, MySQL, Postgres, Oracle, etc.



해결책..

만약 Ait (AM?, AZ?) 제품이면 제값 주고 IOPS 보장 제품 구이!

Premium Storage Disk Type	P10	P20	P30
Disk size	128 GiB	512 GiB	1024 GiB (1 TB)
IOPS per disk	500	2300	5000
Throughput per disk	100 MB per second	150 MB per second	200 MB per second



만약 이런 유형의 제품이 있다면.. or 내용이 있다면..



쾌적한 조건/손으로
계속 이사 다니기..

월세가 아닌?
일세 / 주세가
될지도..



클라우드 환경에서 수치화해야 되는 대표적인 지표 몇가지.

- IOPS
- Disk Queue Length (win) / iowait (linux)
- CPU Steal Time 등..
- <http://bencane.com/2012/08/06/troubleshooting-high-io-wait-in-linux/>



클라우드 고민할 거리들..

- 막대한 부화를 처리할 수 있는 만능 열쇠?
- 클라우드의 보안? (클라우드 회사 지원??)
- 국내 클라우드 경험 인력의 부족..
- 관리 유지 보수는? 어떻게?



클라우드 사건들.

- Heart Bleed
 - 2014년 9월 25일 ~ 10월 5일 : AWS 10일간 돌아가며 리부팅
 - 리부팅후 VM이 올라오지 않아. 서비스의 Downtime이 발생한 사건들..

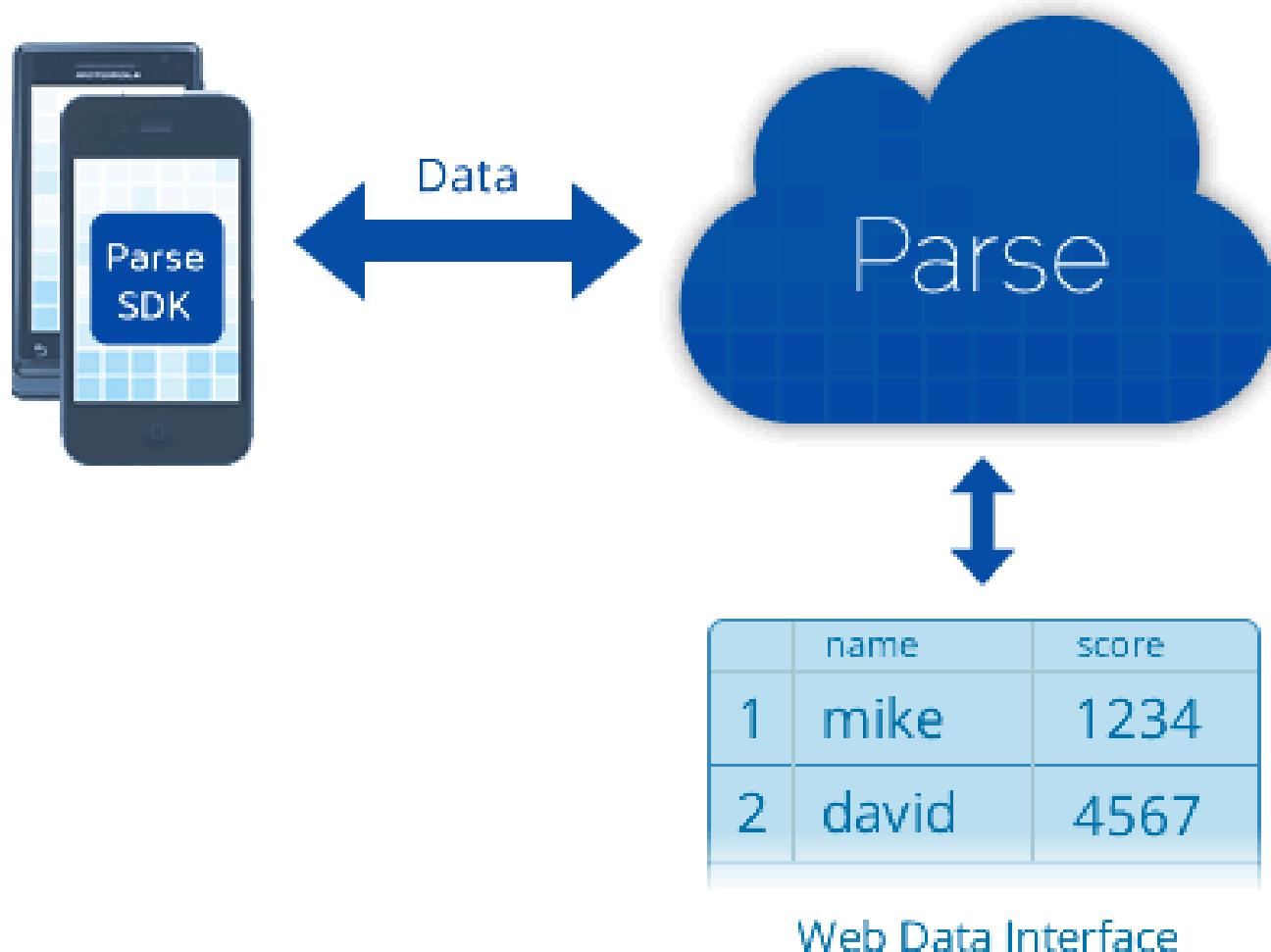


클라우드 사건들.

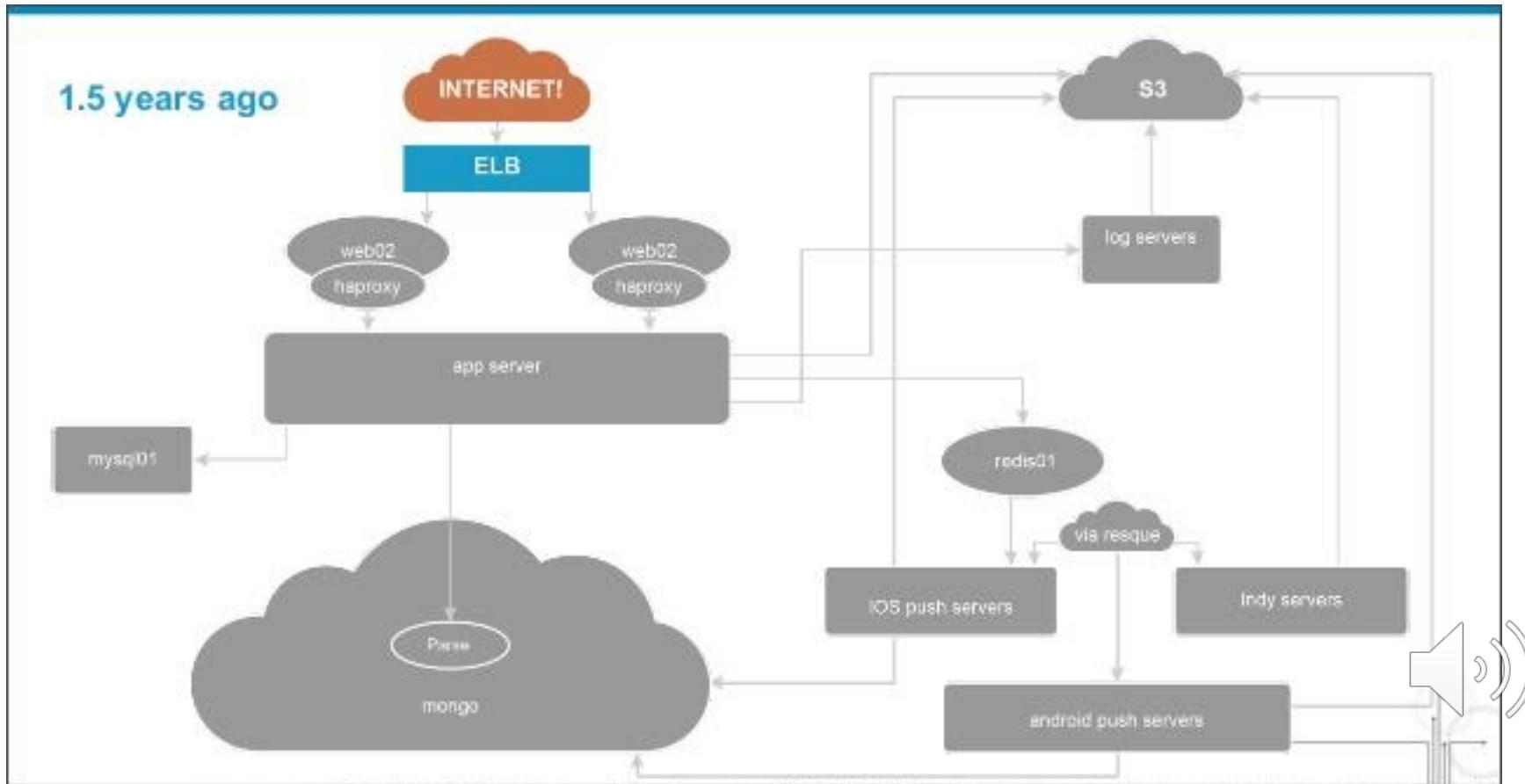
- 리부팅
 - 재부팅하면 시간이 변경되는 클라우드..
 - 1900.01.04 로..
 - 게임 아이템 / 캐릭터 레벨이 뒤죽박죽..
 - 파일 시스템 Mount 사라짐..
담당자 전화 후 5, 10시간 이후 정상화.



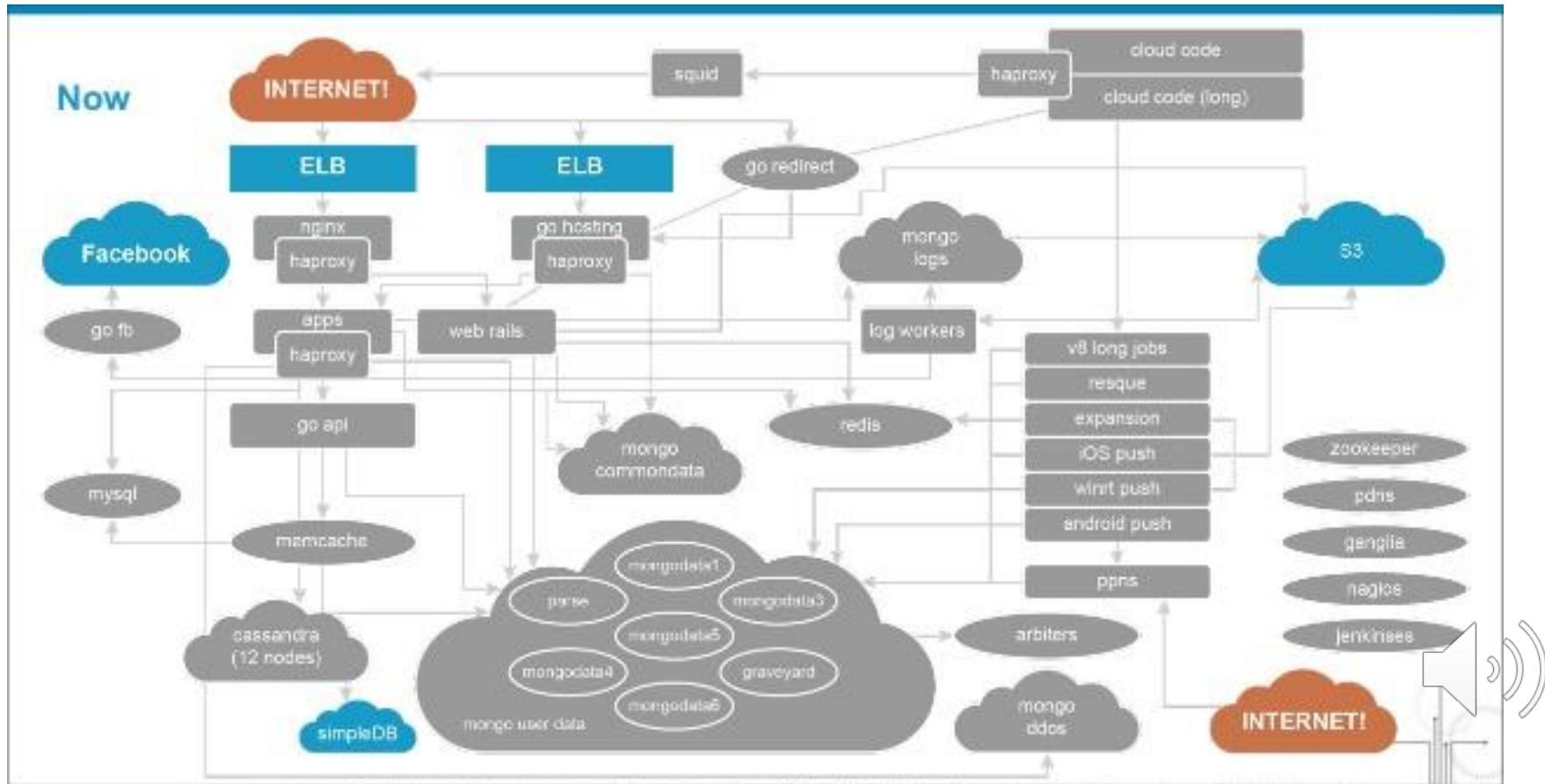
Parse 사례 (BaaS) Facebook 에 1조에 팔림.



처음엔 누구나 가볍게 시작



이런 아키텍처를 고민해본 개발자? 얼마나 있을까?



클라우드 활성화 걸림돌

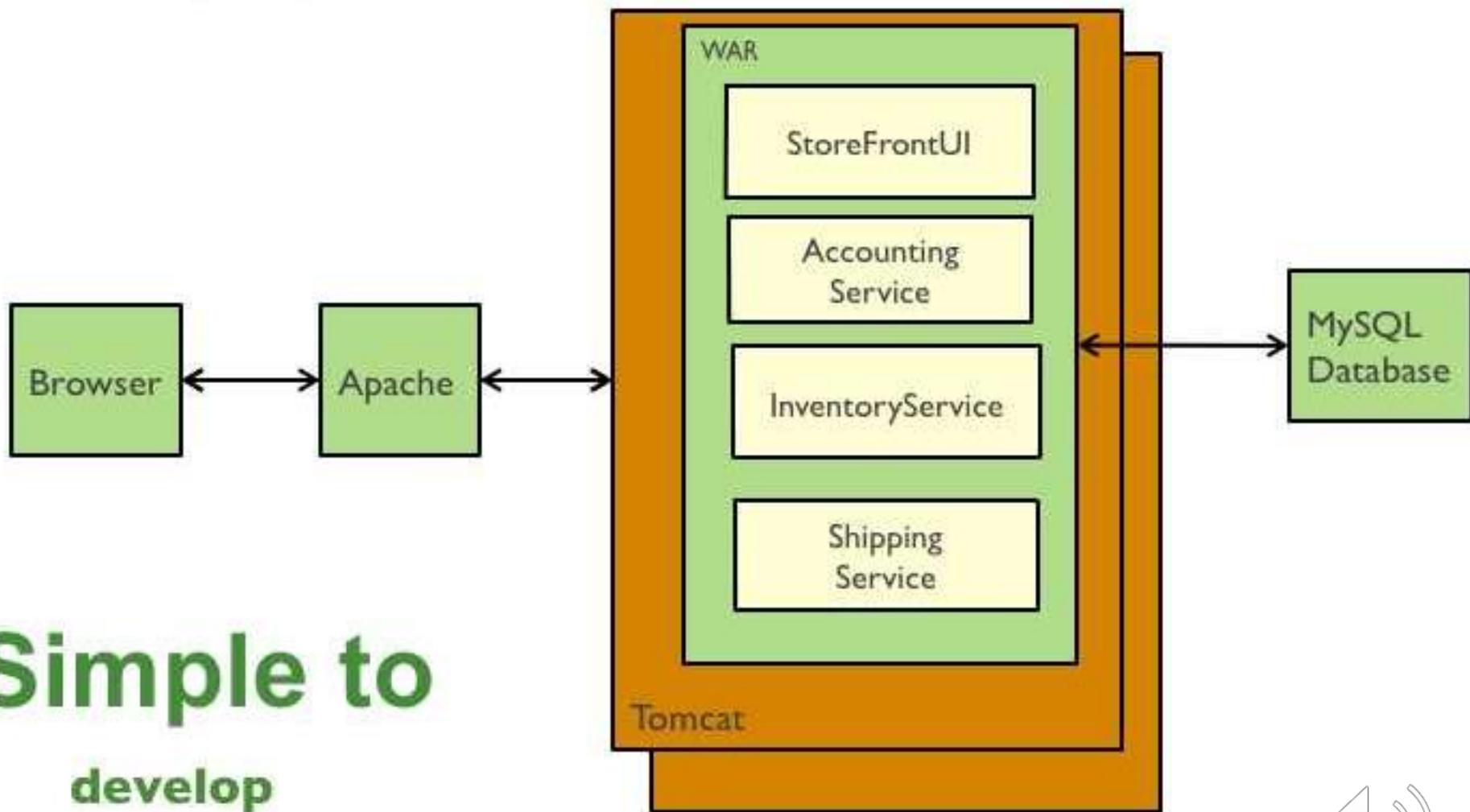
- 오픈소스 문제
 - 클라우드는 대부분 ubuntu , centos 만 제공 -> 결국 사업 솔루션을 오픈소스 내부 솔루션으로 대체해야 하는 이슈
 - 상업 솔루션 잘쓰는 부서가 오픈소스로 넘어올까?
 - 시스템 다운시 책임과 부서 매출과 관련된 깊은 이야기..



7. 클라우드가 가져오는 백엔드의 아키텍처 패러다임 변화



이전에는..

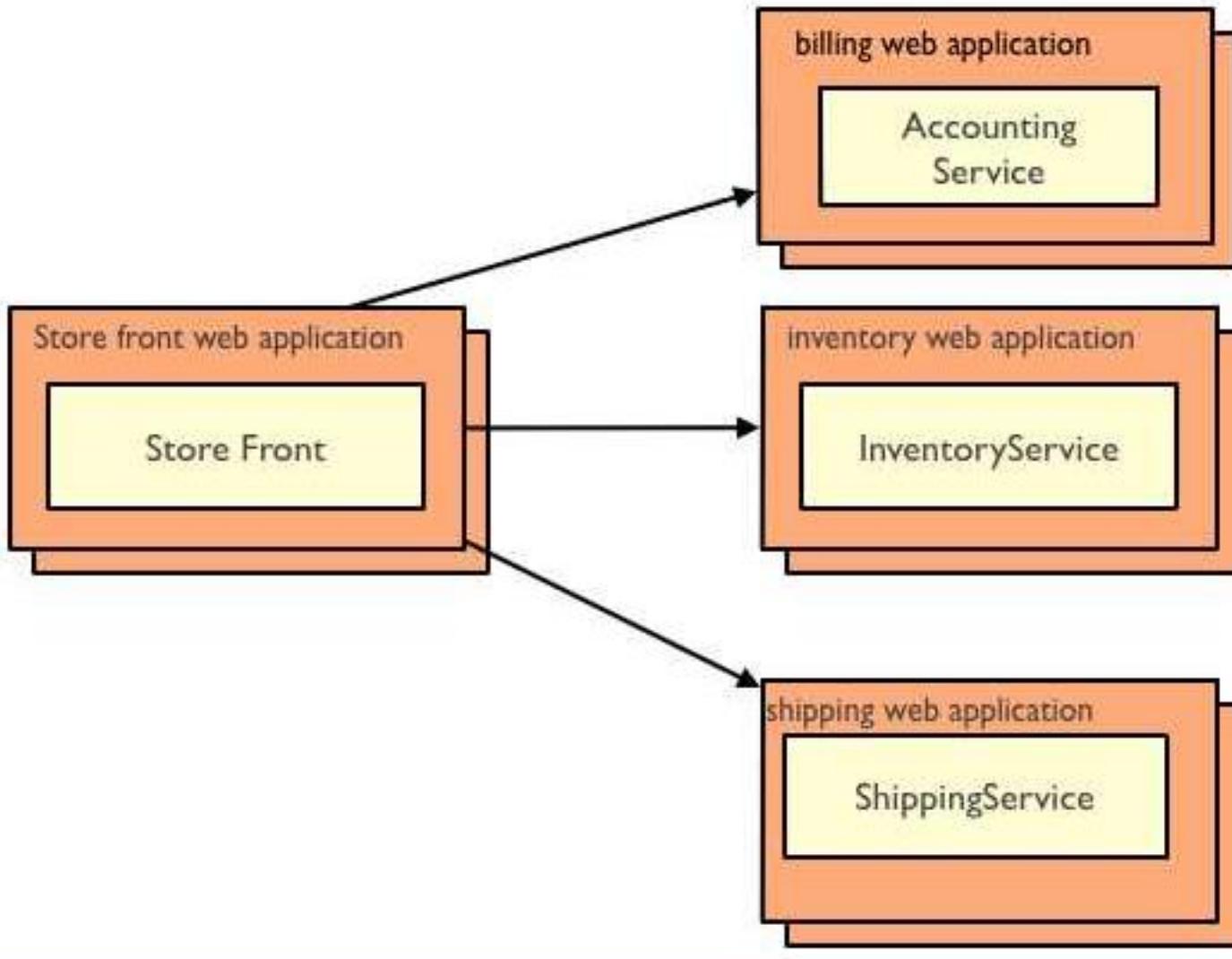


Simple to

**develop
test
deploy**

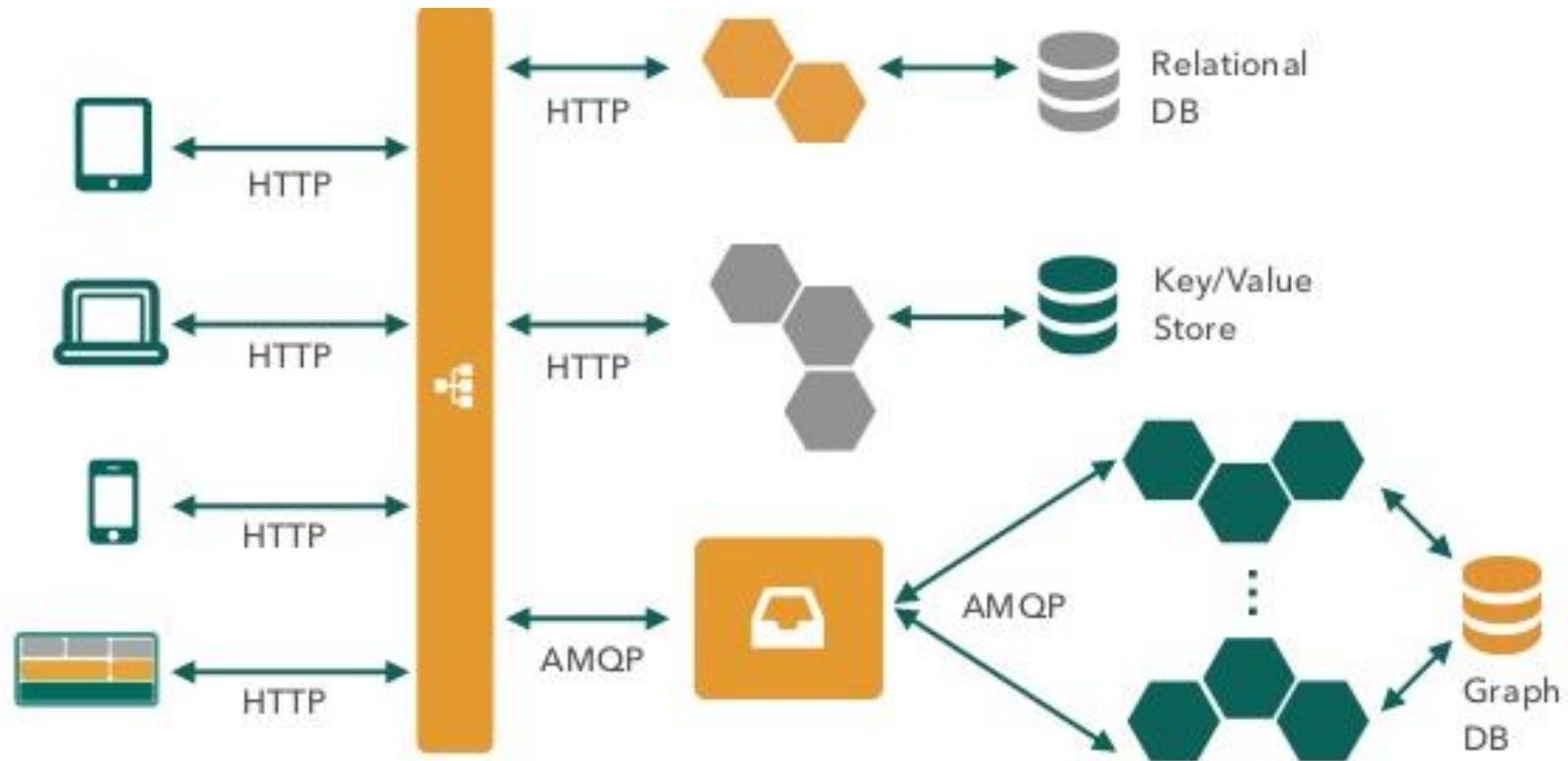


요즘에는

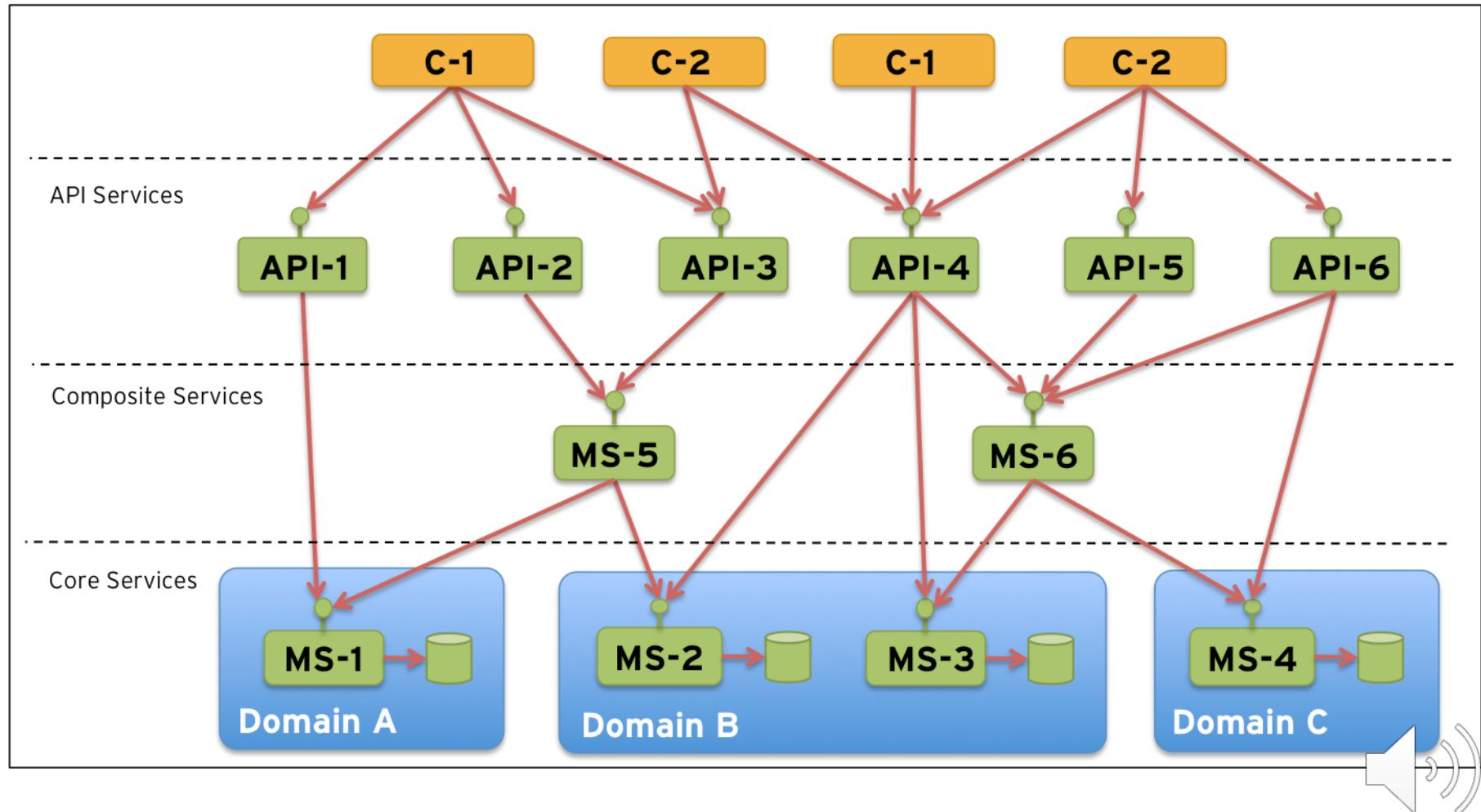


Apply X axis cloning and/or Z axis partitioning to each service

Micro Service Architecture



Micro Service Architecture



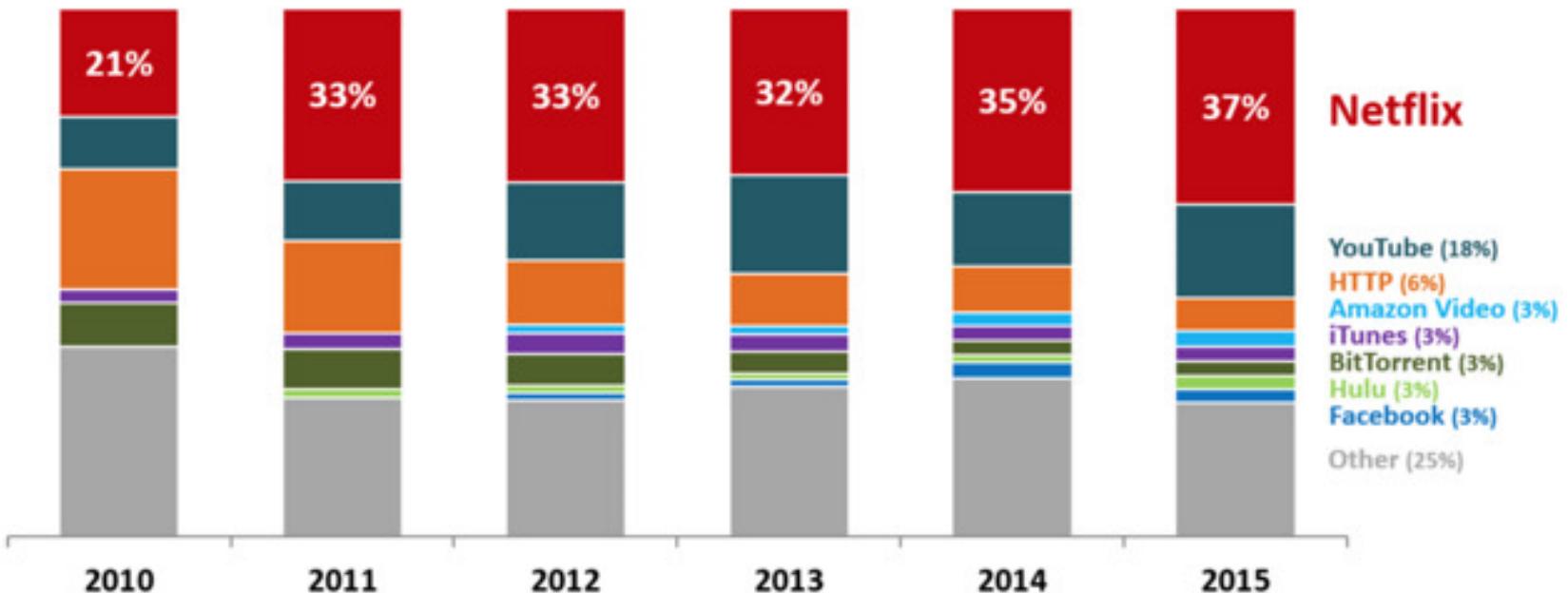
MSA의 해답은 무얼까?

현존하는 MSA의 끝판왕에게 조금만 노하우
를 전수받자!



MSA의 끝판왕

Share of Peak Download Internet Traffic in North America



Source: Sandvine



Netflix 7년간 노력끝에 전체 시스템 클라우드 이전 완료!

11 February 2016

Completing the Netflix Cloud Migration

Our journey to the cloud at Netflix began in August of 2008, when we experienced a major database corruption and for three days could not ship DVDs to our members. That is when we realized that we had to move away from vertically scaled single points of failure, like relational databases in our datacenter, towards highly reliable, horizontally scalable, distributed systems in the cloud. We chose Amazon Web Services (AWS) as our cloud provider because it provided us with the greatest scale and the broadest set of services and features. The majority of our systems, including all customer-facing services, had been migrated to the cloud prior to 2015. Since then, we've been taking the time necessary to figure out a secure and durable cloud path for our billing infrastructure as well as all aspects of our customer and employee data management. We are happy to report that in early January, 2016, after seven years of diligent effort, we have finally completed our cloud migration and shut down the last remaining data center bits used by our streaming service!

WRITTEN BY

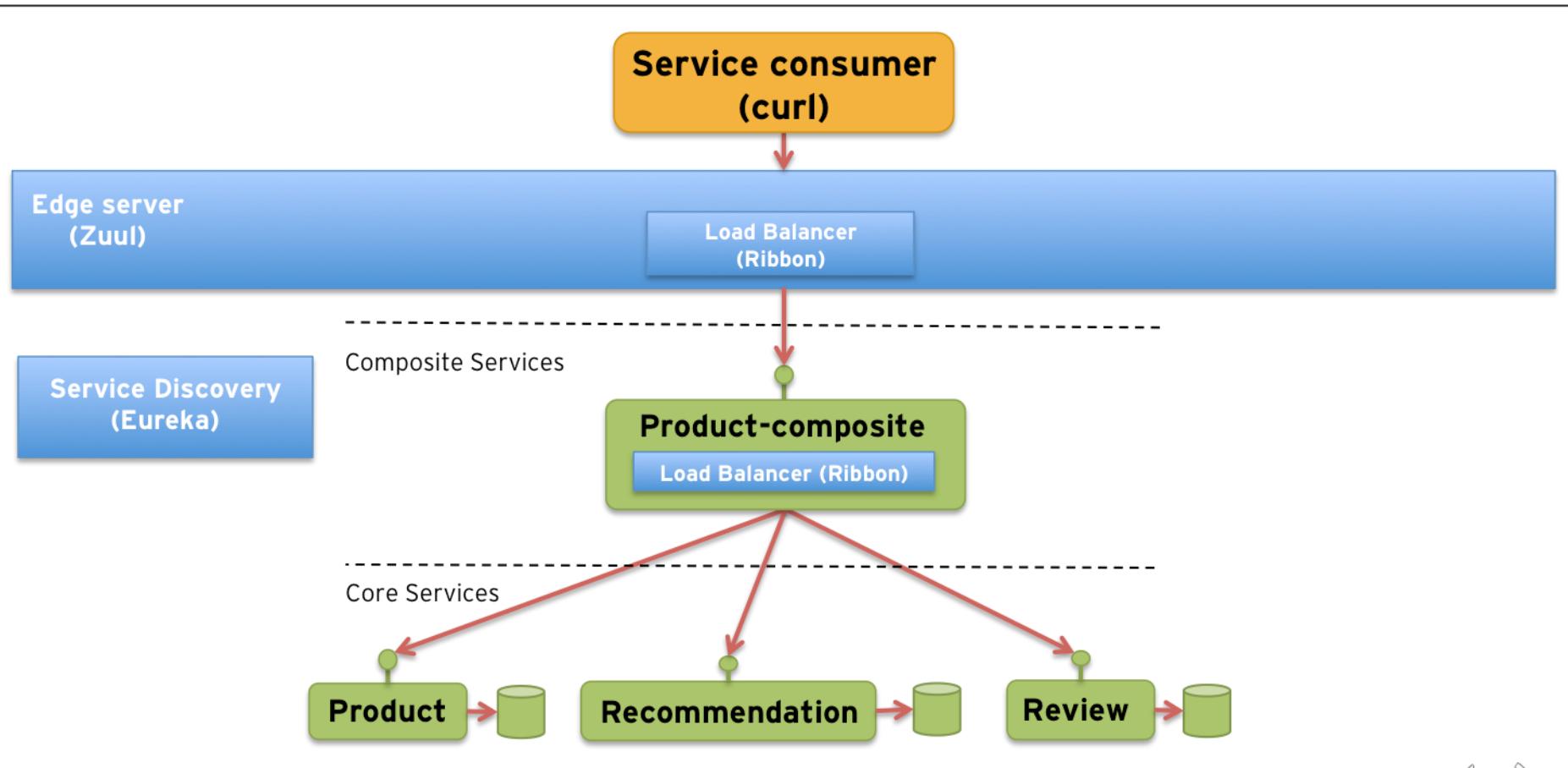


YURY IZRAILEVSKY

Vice President, Cloud and Platform Engineering



MSA에 대한 아마존에 대한 해답!



MSA에 대한 아마존에 대한 해답!

Operations Component	Netflix, Spring, ELK
Service Discovery server	Netflix Eureka
Dynamic Routing and Load Balancer	Netflix Ribbon
Circuit Breaker	Netflix Hystrix
Monitoring	Netflix Hystrix dashboard and Turbine
Edge Server	Netflix Zuul
Central Configuration server	Spring Cloud Config Server
OAuth 2.0 protected API's	Spring Cloud + Spring Security OAuth2
Centralised log analyses	Logstash, Elasticsearch, Kibana (ELK)



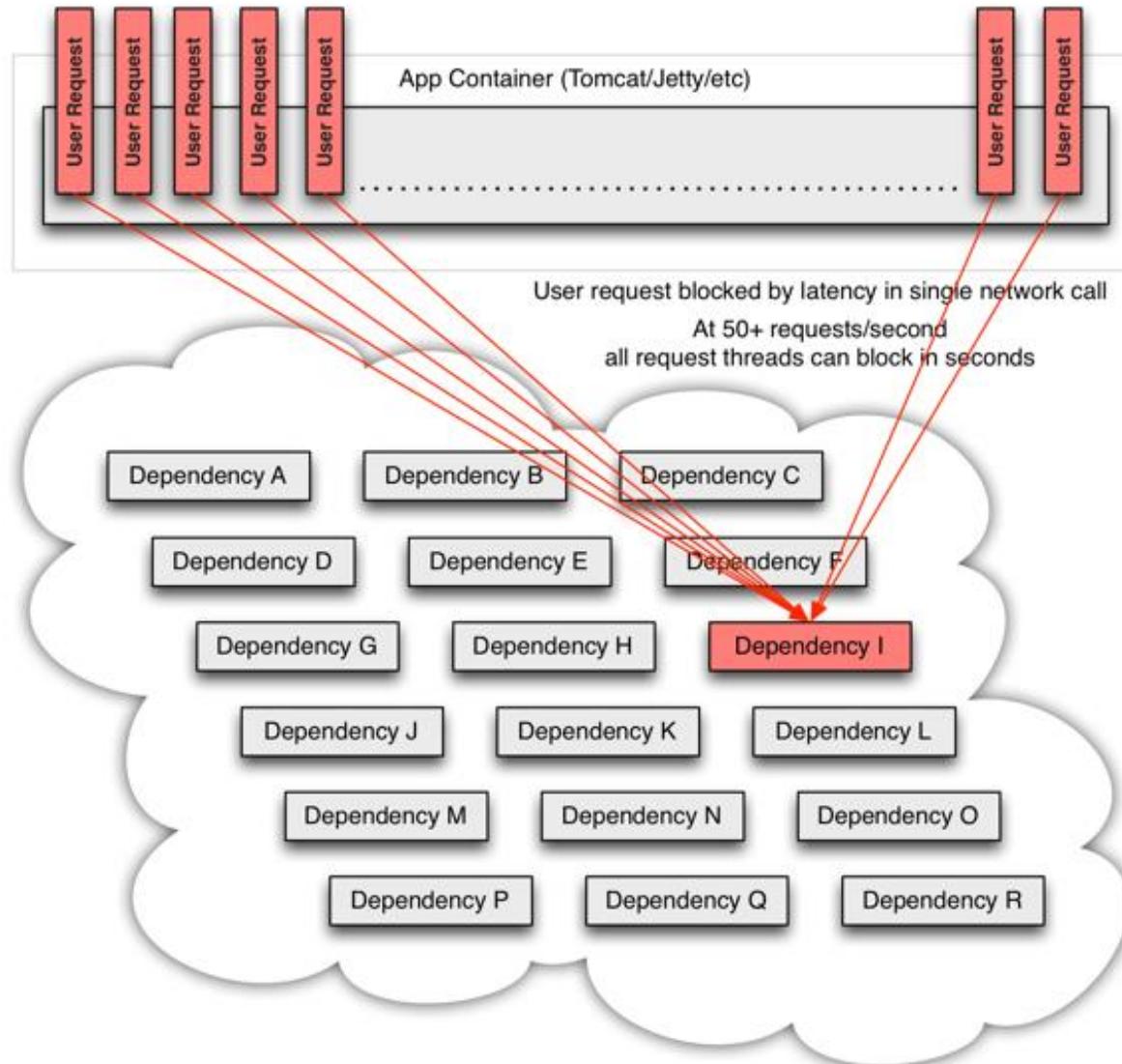
이 수많은 컴포넌트들을
어떻게 잘 관리할 것인가?

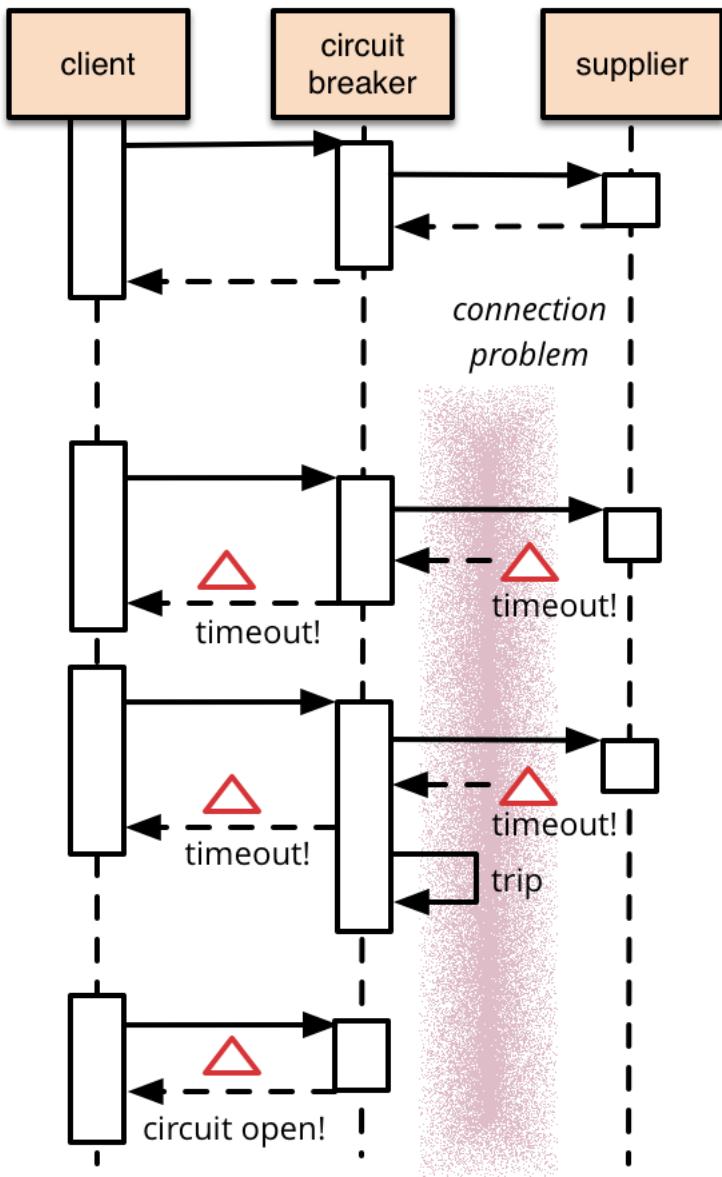


실패할려면 빨리 실패하자!



Hystrix





이러한 일이
자주 발생하면.



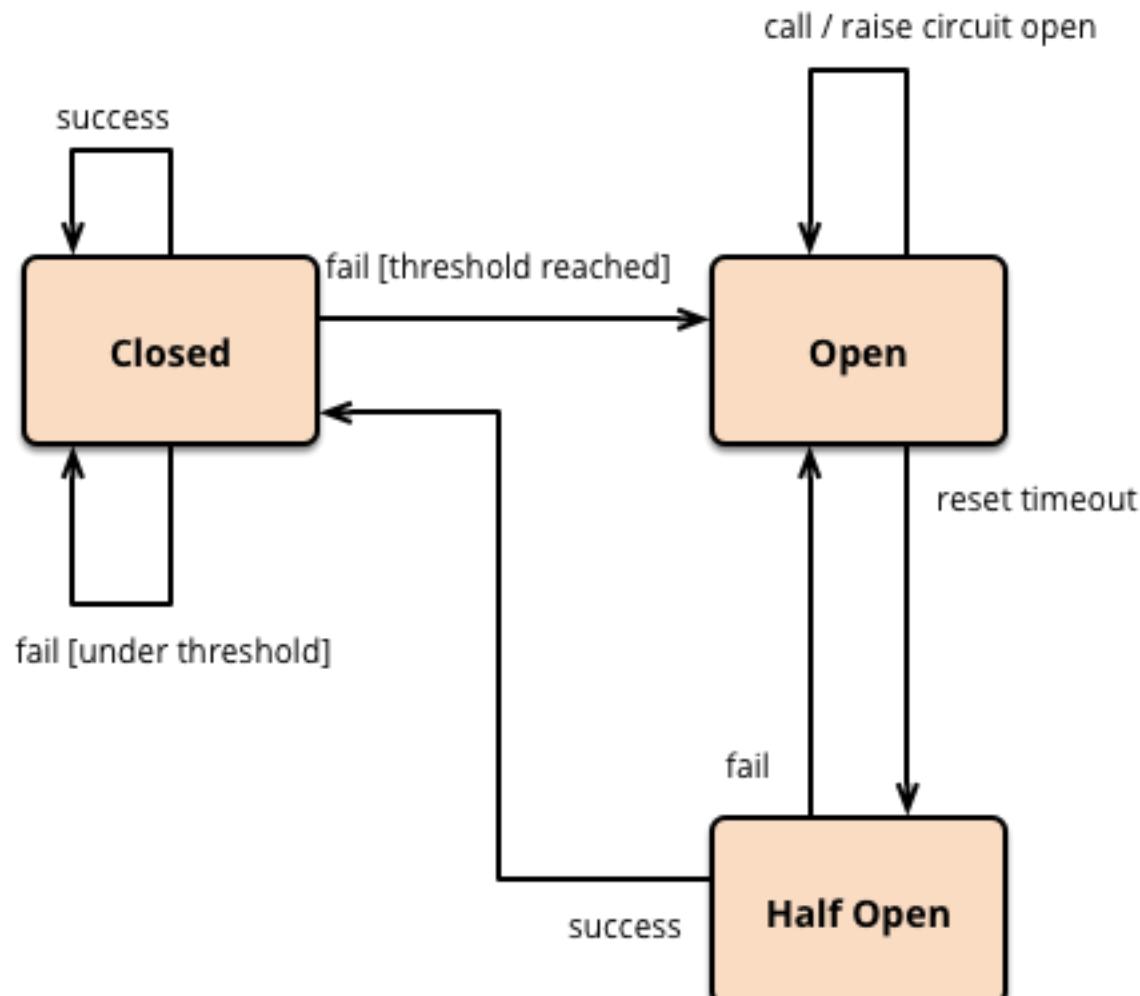
Connection을 얻지 못해 전체 요청에 80%가 거절..



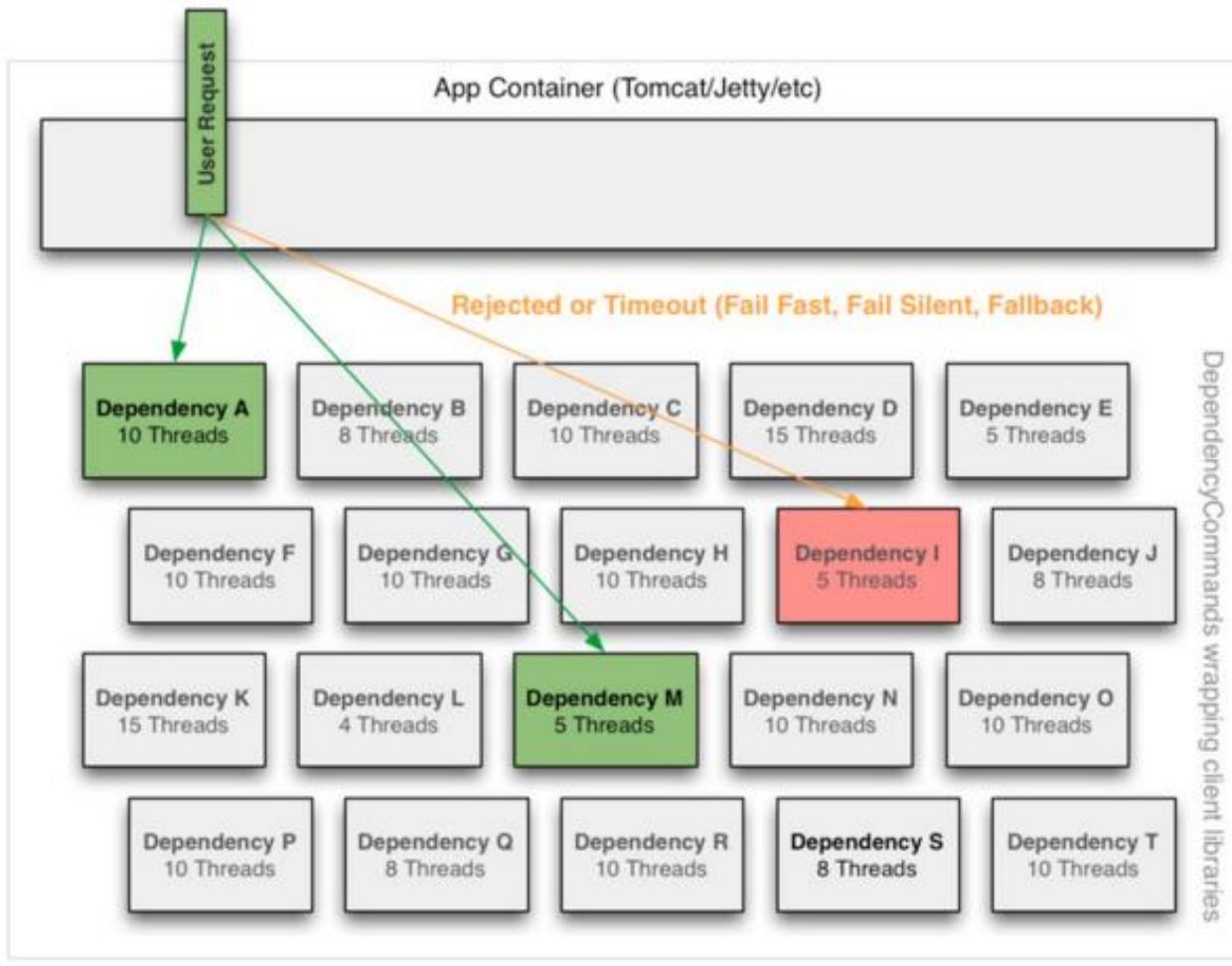
```
"Timeout guard" daemon prio=10 tid=0x00002aaacd5e5000 nid=0x3aac runnable [0x00002aaac388f000] java.lang.Thread.State: RUNNABLE
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:339)
- locked <0x000000055c7e8bd8> (a java.net.SocksSocketImpl)
at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:200)
at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:182)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:391)
at java.net.Socket.connect(Socket.java:579)
at java.net.Socket.connect(Socket.java:528)
at java.net.Socket.(Socket.java:425)
at java.net.Socket.(Socket.java:280)
at org.apache.commons.httpclient.protocol.DefaultProtocolSocketFactory.createSocket(DefaultProtocolSocketFactory.java:80)
at org.apache.commons.httpclient.protocol.ControllerThreadSocketFactory$1.doIt(ControllerThreadSocketFactory.java:91)
at org.apache.commons.httpclient.protocol.ControllerThreadSocketFactory$SocketTask.run(ControllerThreadSocketFactory.java:158)
at java.lang.Thread.run(Thread.java:722)
```



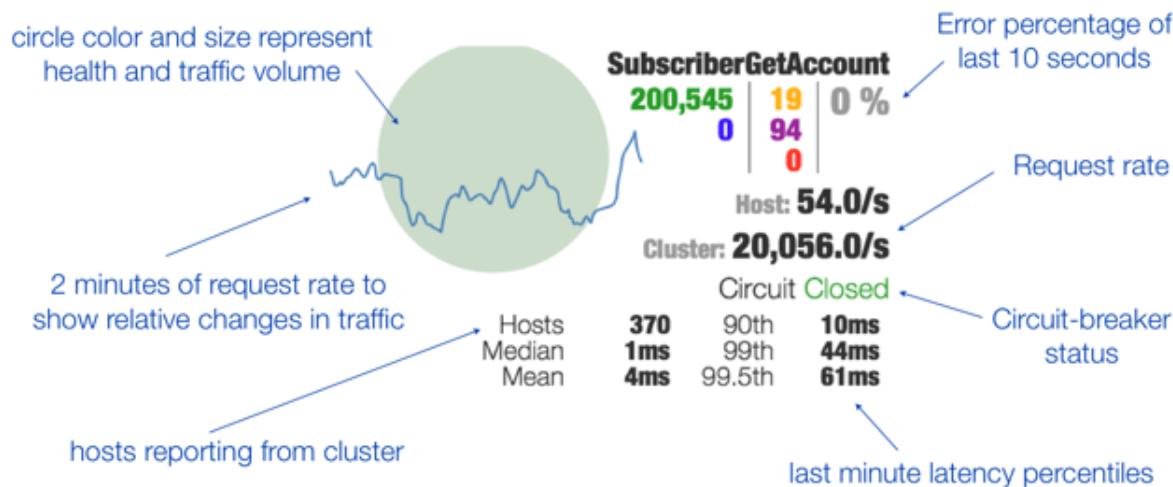
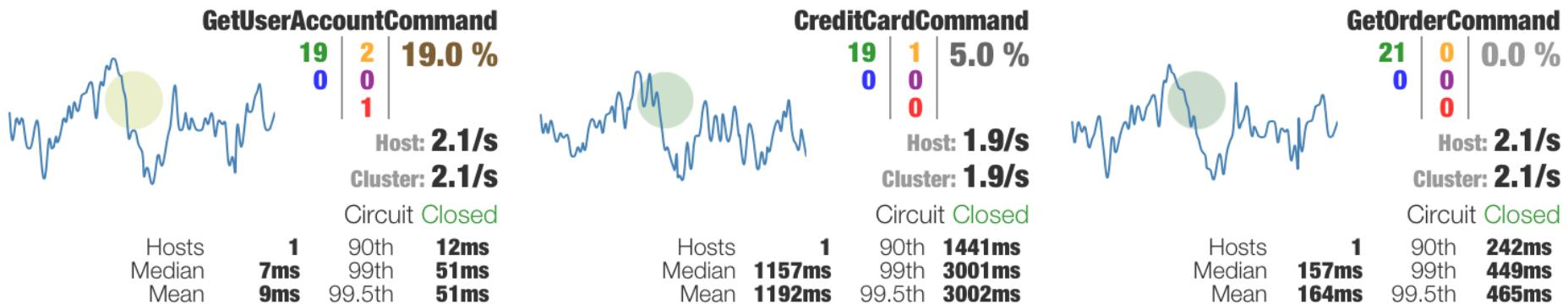
그래서 circuit break pattern도 입



임계값 정하기!



Hystrix Dashboard



Rolling 10 second counters
with 1 second granularity

Successes	200,545	Thread timeouts
Short-circuited (rejected)	0	94
		Thread-pool Rejections
		Failures/Exceptions



tical | Volume | Error | Mean | Median | 90 | 99 | 99.5

Success | Latent | Short-Circ



참고 자료 - OpenHub.NET

- 오픈소스SW의 탄생부터 현재까지의 역사를 한눈에 보여주는 박물관
- 오픈소스SW의 개발과 진화과정을 추적하고 공개하는 사이트
- 품질을 평가할 수 있는 다양한 지표들을 제공

The screenshot shows the homepage of OpenHub.NET. At the top left is a 'Join Now' section with three icons: a green plus sign with a checkmark, a green square with arrows, and a yellow person icon. Below these are links to 'Claim your contributions', 'Manage your project's data', and 'Highlight your use of FOSS'. A large green 'Join Now' button is at the bottom. To the right is a 'What's New' section with a message: 'Ohloh has changed' followed by 'It's now Open HUB' in blue text, with a large blue speech bubble containing the letters 'OH'. Below this are three main sections: 'Most Popular Projects', 'Most Active Projects', and 'Most Active Contributors'. Each section has a title, a list of projects or users with their logos and names, and a horizontal bar chart showing their popularity or activity levels. The 'Most Popular Projects' section includes Mozilla Firefox, Apache HTTP Server, MySQL, Apache Subversion, PHP, and Linux Kernel. The 'Most Active Projects' section includes MediaWiki, Google V8 JavaScript Engine, Simulation of Urban MObility, Systems Biology Markup Language (SBML), DPorts, and Mozilla Firefox. The 'Most Active Contributors' section includes Translation updater bot, Scaramanga, Andrea Scarpino, Peter Simons, patacongo, and xaizek. A speaker icon is in the bottom right corner.

Join Now

Claim your contributions

Manage your project's data

Highlight your use of FOSS

Join Now

What's New

Ohloh has changed

It's now Open HUB

OH

Most Popular Projects

- Mozilla Firefox 13090 users
- Apache HTTP Server 9345 users
- MySQL 9073 users
- Apache Subversion 8698 users
- PHP 7674 users
- Linux Kernel 6982 users

Most Active Projects

- MediaWiki 22422 commits
- Google V8 JavaScript Engine 21984 commits
- Simulation of Urban MObility 8774 commits
- Systems Biology Markup Language (SBML) 8382 commits
- DPorts 6153 commits
- Mozilla Firefox

Most Active Contributors

- Translation updater bot 1379 commits
- Scaramanga 1366 commits
- Andrea Scarpino 1010 commits
- Peter Simons 889 commits
- patacongo 752 commits
- xaizek 736 commits