

Node.js를 이용해서 Big Data 세계를 헤엄친 이야기

나와 같은 실수를 반복하지 마세요

SK Planet, Data Infrastructure

ByungJoon Lee

imjuni.github.io/playnode2015

이 페이지는 Reveal.js를 사용해서 만들었습니다.
위 주소를 접속하면 페이지를 바로 볼 수 있습니다
다만, 이미지는 삭제 되었을 수도 있어요 π_π (저작권 문제로 인해서)

무슨 일을 하고 있어요?

Big Data Visualization 시스템을 개발합니다

Big Data ♡?

Big Data란,

Wikipedia:

기존 데이터베이스 관리도구로 데이터를 수집, 저장, 관리, 분석할 수 있는 역량을 넘어서는 대량의 정형 또는 비정형 데이터 집합 및 이러한 데이터로부터 가치를 추출하고 결과를 분석하는 기술을 의미한다

약간 뜬구름 잡는 느낌, ..

딱히 정해져 있지 않아요,

대용량의 데이터를 다루고 저장하는 기술을 통칭하는 것 같습니다

Hadoop, Hive, HBase, Impala, Presto, Sentry, Kafka, Zookeeper etc ...

저희 팀에서는

하루 25억 건 이상의 데이터(로그)가 들어오고

6PB 정도의 데이터를 저장되어 있습니다

일일 사용량은 11TB정도 입니다

이런 대용량 데이터를 다루는 것을
Big Data라고
하는 것 같습니다

Big Data

- 기존 데이터베이스 관리도구로 수집, 저장, 관리, 분석할 수 있는 규모를 넘어서는 데이터를 다루는 것
- 관련 제품:
 - Hadoop, Hive, HBase, Impala, Presto, Sentry, Kafka, Zookeeper etc ..

참 쉽죠?

잘 받아서 잘 저장하고,

이런 작업도 아주 중요하지만

잘 보여 주는 것

매우 중요합니다!

Data Visualization 시스템이란,

잘 보여 주는 것

그래서 잘 보여주는

R Studio, MSTR, Hue와 같은 도구를

Data Visualization 도구라고 합니다

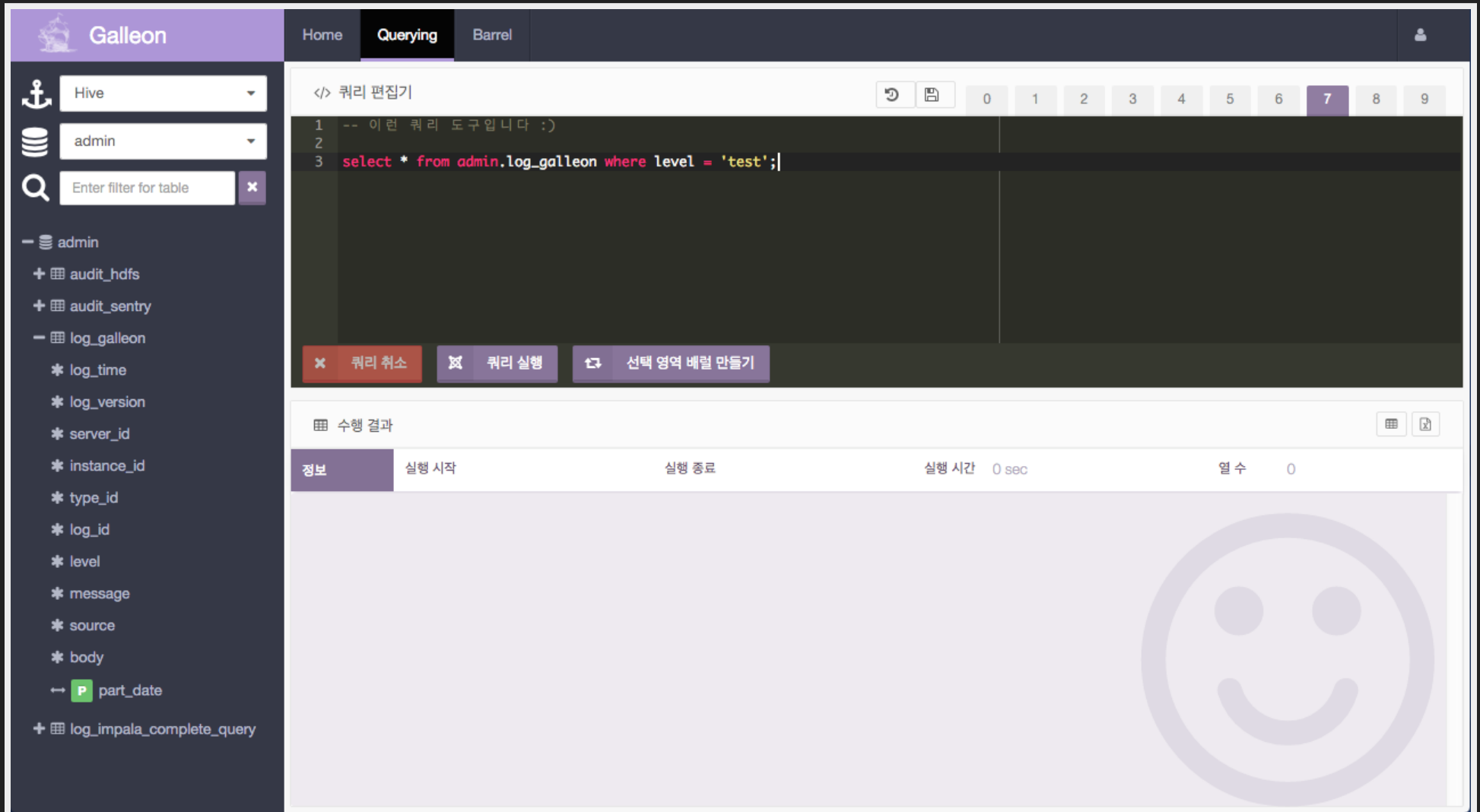
제가 만든 Data Visualization 도구는

보다 솔직하게 이야기를 하면,



그냥 쿼리를
실행하는 도구예요,...

이런 느낌??



The screenshot displays the Galleon web interface. The top navigation bar includes 'Home', 'Querying', and 'Barrel'. The left sidebar shows a database connection to 'Hive' as 'admin' and a list of tables including 'audit_hdfs', 'audit_sentry', 'log_galleon', and 'log_impala_complete_query'. The main area is titled '</> 쿼리 편집기' (Query Editor) and contains the following SQL query:

```
1 -- 이런 쿼리 도구입니다 :)
2
3 select * from admin.log_galleon where level = 'test';
```

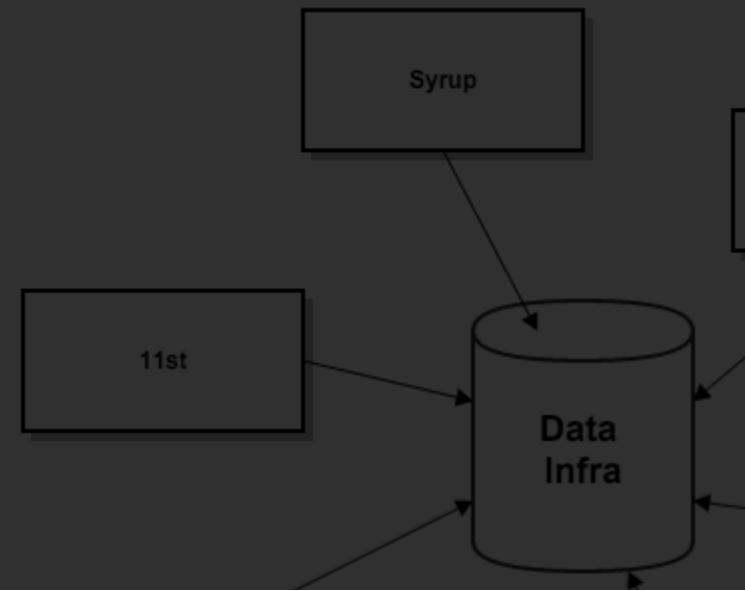
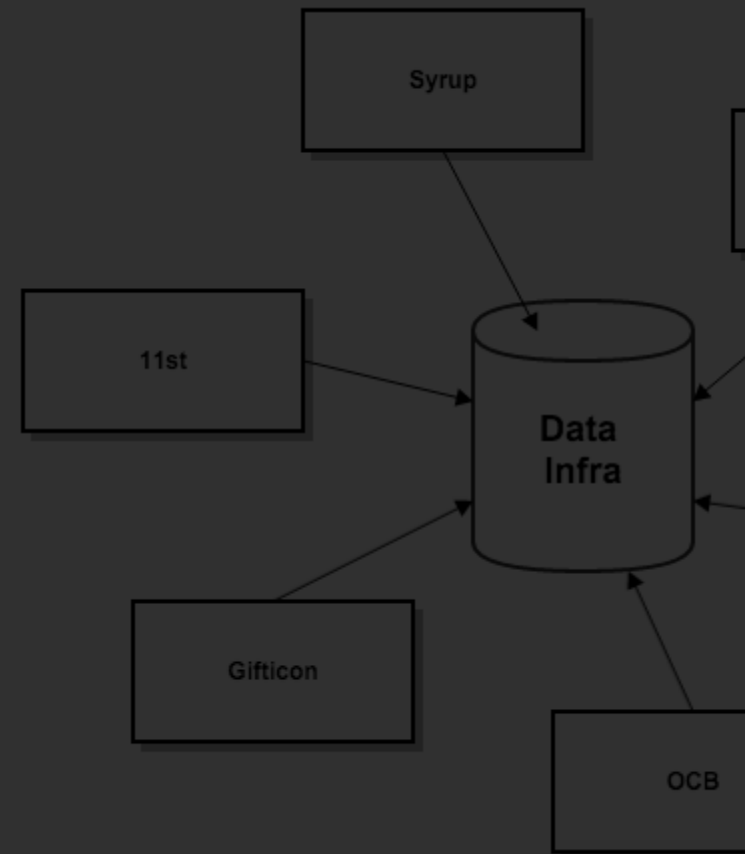
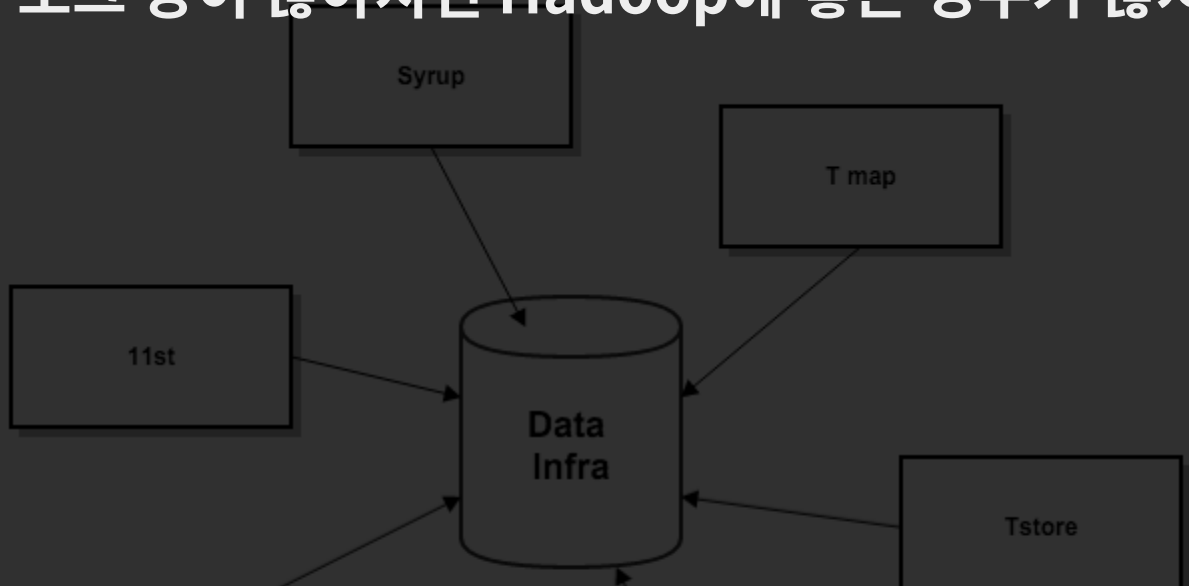
Below the editor are buttons for '쿼리 취소' (Cancel Query), '쿼리 실행' (Execute Query), and '선택 영역 베릴 만들기' (Make Selection Area Visible). The '수행 결과' (Query Results) section shows a table with columns: '정보' (Info), '실행 시작' (Execution Start), '실행 종료' (Execution End), '실행 시간' (Execution Time) at 0 sec, and '열수' (Column Count) at 0. The table body is empty, featuring a large smiley face watermark.



최근 로그가

아주 다양한 곳에서 유입되고,

로그 양이 많아지면 Hadoop에 쌓는 경우가 많지요



Hadoop에 쌓이면, Hive를 Hadoop에 붙이고,

이와 같은 도구를 이용해서 즉시 데이터를 조회할 필요성이 생깁니다

잘 쌓은 로그를 이용해서

UV, PV

쿼리 한 방에 뽑으면 얼마나 편할까요?

☆ Big Data에서!

그래서

이런 시스템이 필요합니다

궁서체입니다 ;)

근데 hue가
비슷한 일을 하
거든요

The screenshot shows the Hive Editor interface. The top navigation bar includes 'Hive Editor', 'Query Editors', 'Data Browsers', 'Workflows', '검색', 'Security', '파일 브라우저', 'Job Browser', and '2publiv'. The main area is divided into a left sidebar for '데이터베이스' (Database) and a central query editor. The query editor contains the SQL statement: `1 select * from sample`. Below the query editor are buttons for '실행' (Execute), '저장' (Save), '다른 이름으로 저장...' (Save with name...), '설정' (Settings), '또는 생성' (Or create), and '새 쿼리' (New query). The bottom section displays the results of the query in a table format, with columns for 'sample.code', 'sample.description', 'sample.total_emp', and 'sample.salary'.

	sample.code	sample.description	sample.total_emp	sample.salary
0	00-0000	All Occupations	135185230	42270
1	11-0000	Management occupations	6152650	100310
2	11-1011	Chief executives	301930	160440
3	11-1021	General and operations managers	1697690	107970
4	11-1031	Legislators	64650	37880
5	11-2011	Advertising and promotions managers	36100	94720
6	11-2021	Marketing managers	166790	118160
7	11-2022	Sales managers	333910	110390
8	11-2031	Public relations managers	51730	101220

모양도 비슷하죠?

개인정보보호법 전자금융거래법

A hand holding a Joker playing card against a background of a person's face. The card is tilted and shows the Joker character. The text '개인정보보호법' and '전자금융거래법' is overlaid in large, bold, red Korean characters.

잘 알려진 도구로는

Compliance 이슈를

해결하기 어렵습니다

망 분리, WEB & WAS & DB 3 tier 구성, 다운로드 제어, 계정 관련 제약, 접근 로그 보관 등
다양한 조건이 있습니다

Data Visualization

- 잘 쌓인 데이터에서 보물과도 같은 정보를 찾아낼 수 있도록 도와주는 도구
 - EDA(Exploratory data analysis)라는 어휘도 있다
- 기존 시스템은 잘 만들었지만 국내에서는 법적 제약으로 인해 사용이 어렵다

참 쉽죠?

그래서

그냥 만들기로 했습니다

저는 Java에 조금 알레르기가 있습니다

그래서 Java를 사용하지 않고 구현하는 방법을 고민했습니다

그 때

그(그녀)가

눈에 들어왔습니다 ♡

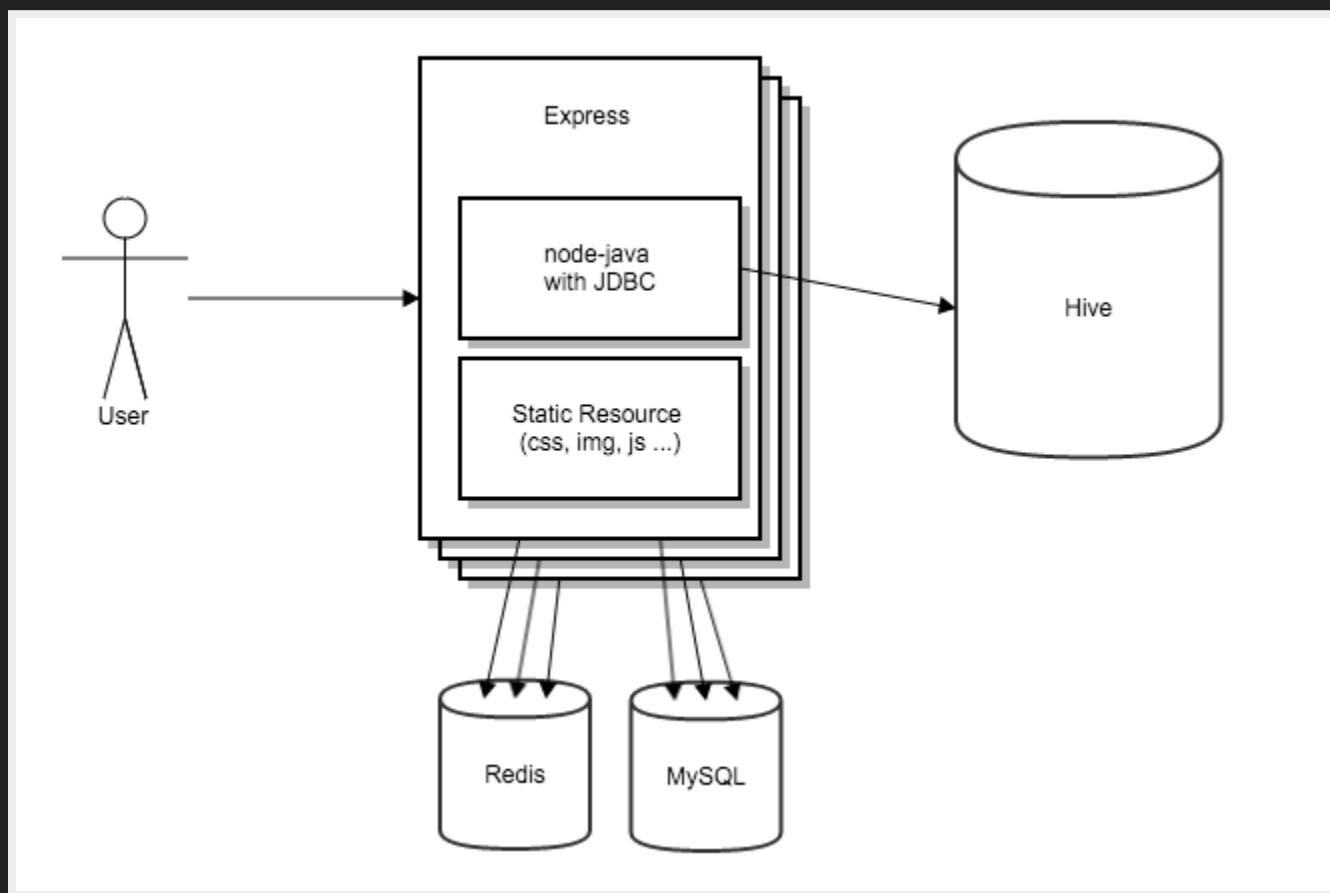


Node.js

쉬운 문법, 다양한 npm 패키지 등 모든 것
을 할 수 있을 것 같았습니다

그렇게 우리는 시작했습니다

처음에는 간단하게 시작해야죠?



- Express

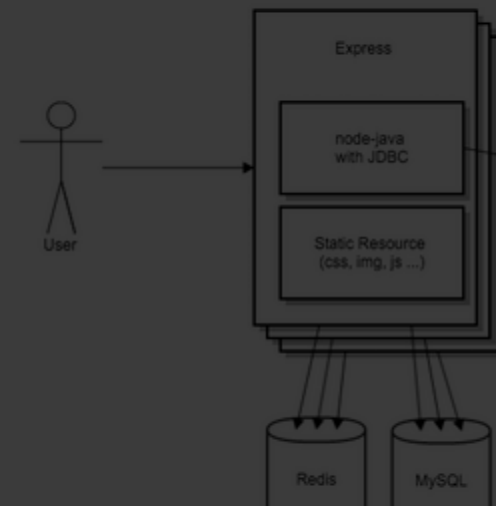
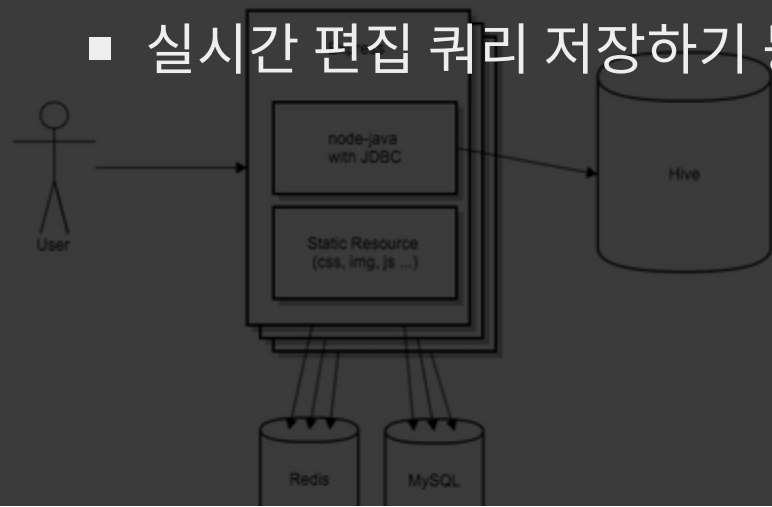
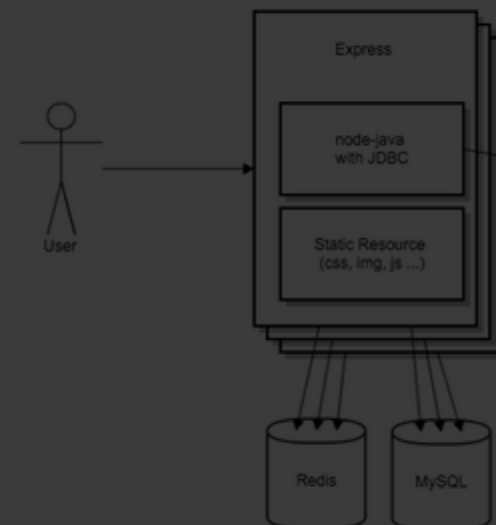
- WEB, WAS를 동시에 처리

- MySQL

- 영구히 저장되는 정보
- 즐겨찾기, 자주 쓰는 쿼리 저장하기 등

- Redis

- 세션 정보 저장하기
- 실시간 편집 쿼리 저장하기 등



오늘부터 우리는 1일이야!



그런데

Hive에 어떻게 접속하고, 어떻게 자료를 가져오지?

node-java

Node.js로 개발을 시도한 이유

Data Visualization을 위해서

Data retrieval 과정이 필수적이고

Data retrieval을 위해서

저희는 Hive에 반드시 접근해야 합니다

Hive에 접근하는 가장 쉬운 방법은 역시
JDBC (for Hive)

node-java를 사용하면

JDBC를 사용할 수 있습니다

node-java 좋아요?

Java 잘해요?

어울리지 않아요

**참 철 없던 시절,
별생각 없이 JDBC 함수를 실행했어요**

**JDBC 함수가 실행되는 동안
Express 서버가 멈춥니다**

node-java에서

함수 실행은

그냥 `while` 문으로 `loop`을
실행하는 것과 같습니다

처음에 Hive와 Impala를 사용했는데,

(소근소근)

impala가 초기 버전에서 invalidate
metadata를 하다가 종종

hang이 걸려서 멈출 때가 있었어요

node-java를 쓰다가

java method에서 hang이 걸리면

Node.js 전체가 hang이 걸립니다

이 문제를 해결하려면

Java 코드를 추가로 개발해서

JDBC 함수를 호출 할 때 thread를 하나 만든 다음 그 thread에서 JDBC function call을 하고

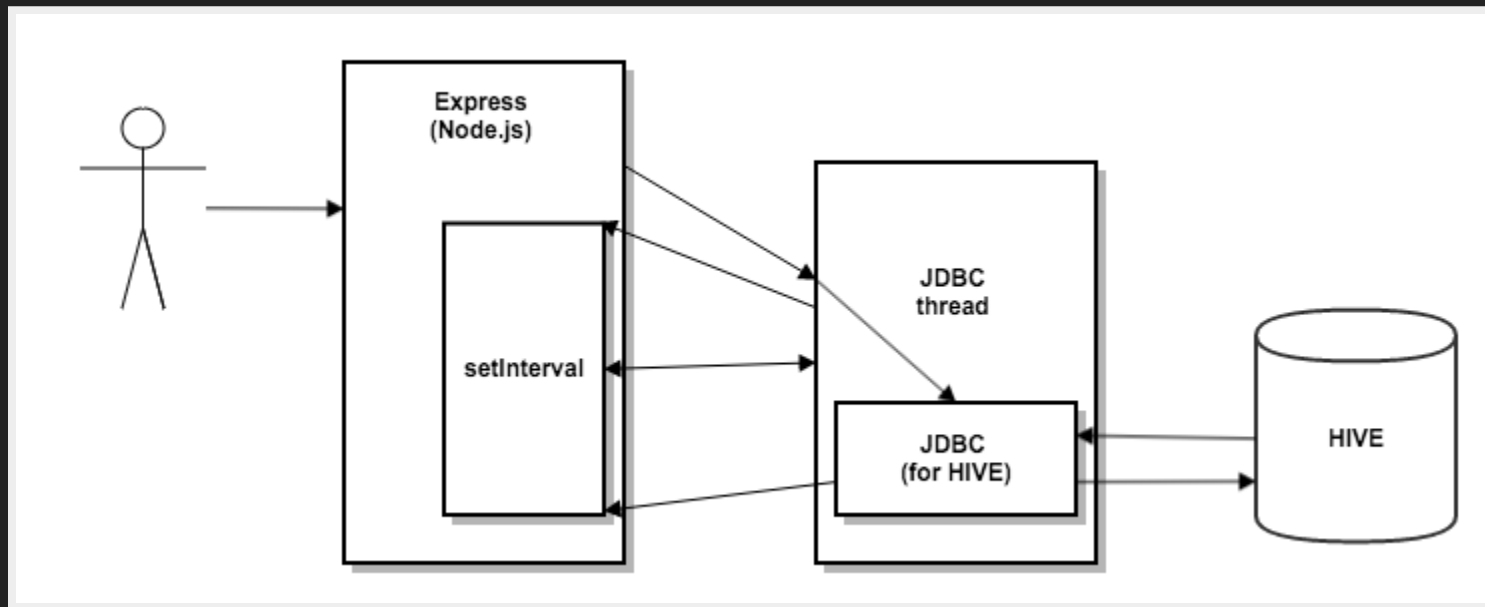
비동기 방식처럼 바로 결과를 돌려 주고,

Node.js에서는 setInterval 등으로 java code를 지속적으로 호출 하면서 thread status를 확인하고,

실행이 완료되면 결과를 가져와야 합니다

만약 실행 시간이 일정 시간 이상 걸리면 취소를 하고 timeout을 내는 코드도 추가해야 합니다, java로...

그림으로 보면,



와,.....,..... [씨-]



망했어요..

**node-java는 이 한 가지 이유
만으로 사용하기 어렵습니다**

2015년, 11월 현재 node-java는 개선을 거듭하여

Async, Sync Java Method Call 외 Promise Java Method Call
기능이 추가되었으며 Generator + Promise를 이용하여 테스트
한 결과 비동기 호출이 가능해졌습니다

이 점, 참고 부탁드립니다!

그런데,

설상가상으로 사용자는 JDBC에 없는 요구사항을 요청합니다

JDBC 함수가 실행되는 동안

Hive 로그나 Impala 로그 등 실행되는 동안 상태를 확인하고 싶어합니다



모든 것을 해결할 수 있는 방법이 있습니다

node-java 군, 넌 해고다

node-java

- Node.js에서 java jar, class 파일을 실행할 수 있도록 해주는 package
- C++, JNI, JDK (for JVM) 사용
- 2015년 11월 기준 괜찮은 선택, 다만
 - 서버 빌드 여부 확인
 - JDK & JVM 확인: jar 파일이 Java 8 으로 개발 되었는가? 등
 - 사용할 때는 반드시 **Promise** 를 이용하여 비동기로 실행

참 쉽죠?

node-thrift

**Thrift는 Hive에서 제공하는 또 다른 접근
방법입니다**

JDBC (for Hive)

디컴파일을 하면 Thrift 프로토콜 스택으로 JDBC Interface를 구현했습니다

그러므로 Thrift를 사용하는 것은 가장 근본적인 Hive 접근 방법입니다

Thrift

<https://thrift.apache.org/>

Thrift는

개발 환경, 언어와 독립된 프로토콜 명세입니다

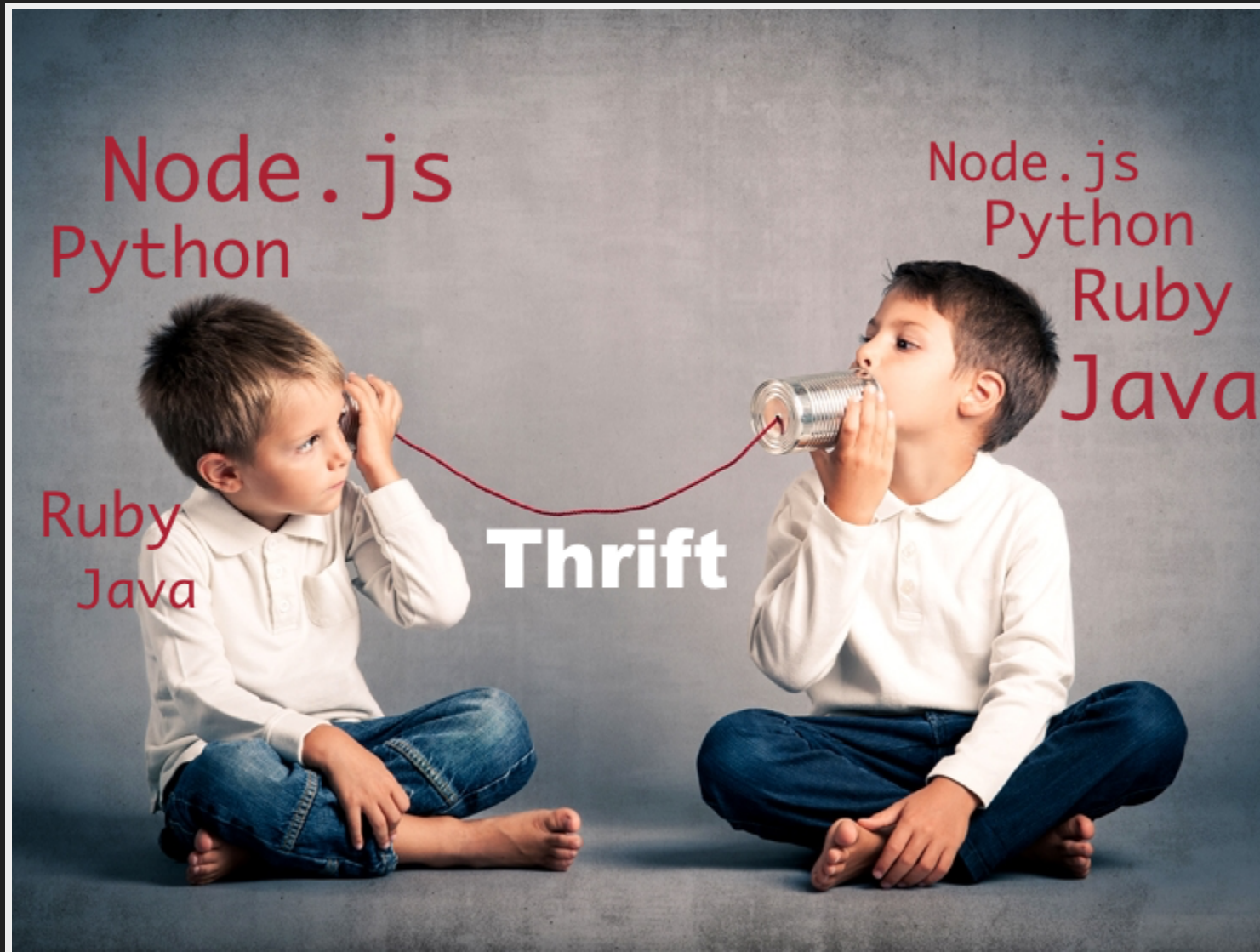
그래서 Thrift를 이용하면 다양한 언어로

프로토콜 구현체를 생성할 수 있습니다

Actionscript 3.0, c_glib, C++, CSharp, D, Dart, Delphi, Go, Graphviz, Haxe Framework, Haskell, Java,

Javascript, Node.js, OCaml, Perl, PHP, Python, Ruby

이런 느낌으로,



**Node.js와 Javascript가
있으니, 문제 없습니다**

Hive 서버에 포함된 Thrift idl을

Thrift로 컴파일하면 Javascript 구현체를 만들 수 있습니다

```
sh> brew install thrift  
sh> thrift -r --gen js:node /apache-hive-1.1.0-src/service/it/TCLIService
```

gen-nodejs가 뿡하고 생기고,

gen-nodejs와 node-thrift를 이용해서 Hive에 접근한 뒤

- 쿼리도 실행하고,
- 쿼리 캔슬도 하고,
- 로그도 가져올 수 있습니다

로그에 대해서 깨알 첨언을 하자면

Apache Hive 1.0 아래 버전을 사용하면 getLog가 없습니다
1.0 이후에 getLog가 추가 되었습니다

만약 Hive 버전이 1.0 아래라면?

CDH에서 배포하는 Hive를 사용하세요~

gen-nodejs 아래, getLog가 있습니다

Apache Hive 1.0 이상에서 getLog

VS

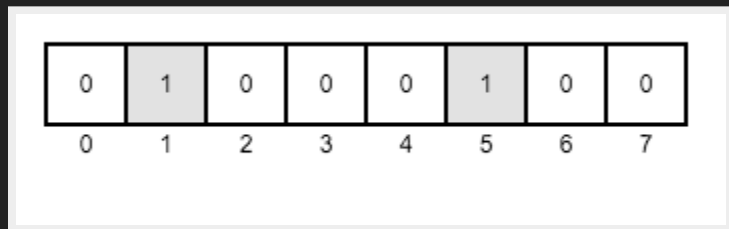
CDH Hive 0.13.0 이하에서 getLog

getLog로 받은 buffer를 해석하는 방식이 조금 달라요,

깨알 첨언 둘,

NULL field를 처리하는 방법이 따로 있습니다

되돌려 받은 buffer 내에, nulls 값이 있고 이 nulls 값은 bit vector 형태로 null 값 인지 아닌지를 확인할 수 있는 값이 들어있습니다



코드로 표현하면,

```
HS2Util.prototype.getIsNull = function getIsNull (buf, index) {  
  var bufIdx = (index !== 0) ? Math.floor(index / 8) : 0;  
  var pos = index % 8;  
  
  return !(buf[bufIdx] & this.BITMASK[pos]);  
};
```


이 모든 것이 귀찮다면,

<https://github.com/imjuni/jshs2>

JSHS2는

이 귀찮은 작업을 모두 해주었지만,

인증을 구현하지 않아서

HIVE auth mechanism을 NOSASL로 설정해야 합니다

kerberos, plain sasl 등과 같은 인증 수단을 말합니다

node-thrift

- Thrift 다른 언어, 플랫폼에서 공통적으로 사용할 수 있는 프로토콜
- getLog를 사용하기 위해서는 Hive 버전 확인
- NULL 필드를 다루기 위해서는 bit vector 사용
- 귀찮으면 **JSHS2**
 - 인증은 NOSASL

와,.. 이제 Hive에 접속할 수 있게 되었어요

와,.. 이제 Hive에 접속할 수 있게 되었어요

이제, 개발할 수 있겠네요

이제, 개발할 수 있겠네요



쿼리를 실행할 때,

Connection

Connection less

vs

Persistent Connection

Impala

초기 Impala에 직접 접속할 때,

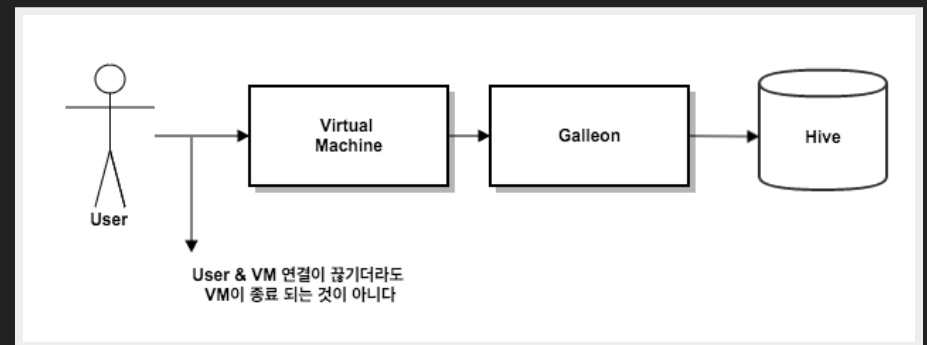
Impala가 충분히 성장하지 않아 Persistent Connection 방식으로
연결을 유지하면, Connection이 완전히 정리되지 않고 leak이 되는 문제가 있었다

Compliance Issue

개인정보보호법 때문에 사용자는 가상머신에서 접속하는데,

보통 가상머신을 종료하지 않고 연결을 끊기만 하기 때문에 쿼리를 실행하고 결과를 보다가

가상머신과 연결을 끊으면 서버에서는 Connection이 계속 남아서 leak이 된다



Hue - Persistent Connection

VS

Galleon - Connection less

제가 만든 것이 Galleon 입니다 ;)

Connection less

반드시 Full-Fetch 과정이 필요

Full-Fetch!

쿼리 결과를 모두 Browser에 저장해야 한다!



내가 직접 Grid Display Engine을 만들어도 되지만,

잘 만든 것을 가져다 사용하기로 결정

Grid

이름	장점	단점
SlickGrid.js	<ul style="list-style-type: none"> • 수백만 건의 데이터도 빠르게 처리 가능 	<ul style="list-style-type: none"> • 안 이쁘.. • 여러 줄 출력 불가능
Handsontable 0.10.5	<ul style="list-style-type: none"> • 30~40 만 건까지는 합리적인 성능 • 여러 줄 출력 가능 • Excel 닮은 꼴 	<ul style="list-style-type: none"> • 비교적 성가신 버그가 많다 • 버전이 올라서 고쳐질 확률이 없다
Handsontable latest	<ul style="list-style-type: none"> • 5~10 만 건까지는 합리적인 성능 • 여러 줄 출력 가능 • Excel 닮은 꼴 • 많은 버그가 수정 됨 	<ul style="list-style-type: none"> • 명색이 Big Data인데, 10만 건은 너무 작다 • 30 만건만 실행해도...
DataTable.js	<ul style="list-style-type: none"> • 다양한 부가 기능 (search, paging 등) 	<ul style="list-style-type: none"> • Infinity Row 기능을 켜면 느리다

Grid

이름	장점	단점
SlickGrid.js	<ul style="list-style-type: none"> • 수백만 건의 데이터도 빠르게 처리 가능 	<ul style="list-style-type: none"> • 안 이쁘.. • 여러 줄 출력 불가능
Handsontable 0.10.5	<ul style="list-style-type: none"> • 30~40 만 건까지는 합리적인 성능 • 여러 줄 출력 가능 • Excel 닮은 꼴 	<ul style="list-style-type: none"> • 비교적 성가신 버그가 많다 • 버전이 올라서 고쳐질 확률이 없다
Handsontable latest	<ul style="list-style-type: none"> • 5~10 만 건까지는 합리적인 성능 • 여러 줄 출력 가능 • Excel 닮은 꼴 • 많은 버그가 수정 됨 	<ul style="list-style-type: none"> • 명색이 Big Data인데, 10만 건은 너무 작다 • 30 만건만 실행해도...
DataTable.js	<ul style="list-style-type: none"> • 다양한 부가 기능 (search, paging 등) 	<ul style="list-style-type: none"> • Infinity Row 기능을 켜면 느리다

Full-Fetch

데이터 전송은,

Socket.io

Node.js Killer package

이러한 과정을 거쳐서!

Galleon

고생스럽지만,

순수 Node.js로

Hive에 접속, 쿼리 실행 등의

개발은 좋은 경험

그런데, 나를

분노

하게 한 것들은 없는가?

개인정보보호법

파일이 **파일시스템에 스키는 것** 조차 불허

모두 메모리에서 처리

다량의 데이터가 Node.js에서 머무르고,
다량의 loop를 수행하며 Excel XML을 생성하고,
Node.js에 부적합

전체 요약

- Node.js도 Big Data Storage에 접근할 수 있다
 - 되도록 Thrift 추천
 - 최근 Presto는 http protocol을 이용하니 node-java 사용하지 않고 개발 가능
- node-java는 나쁘지 않은 선택이지만 확인할 것이 많다
 - 서버 배포 여부
 - JVM, JDK 버전 등
- Grid
 - 역시 Handsontable이 좋지만, 최신 버전은 다량의 데이터를 처리하기에는 무리가 있다
- Node.js와 Hive 사이에 Connection을 유지하기 위해서는 많은 노력이 필요

JSHS2 많이 써주시고, 많이 도와주세요

그래서, 앞으로는?

OpenSource

JSHS2 말고, Galleon 전체를

JSHS2

SASL-PLAIN 지원, kerberos 지원, LDAP 지원

이 부분은 중요도가 조금 떨어진다고 판단하여, 추후 지원할 예정

감사합니다!!

Q & A

