Big Data Platform as a Service @ Netflix

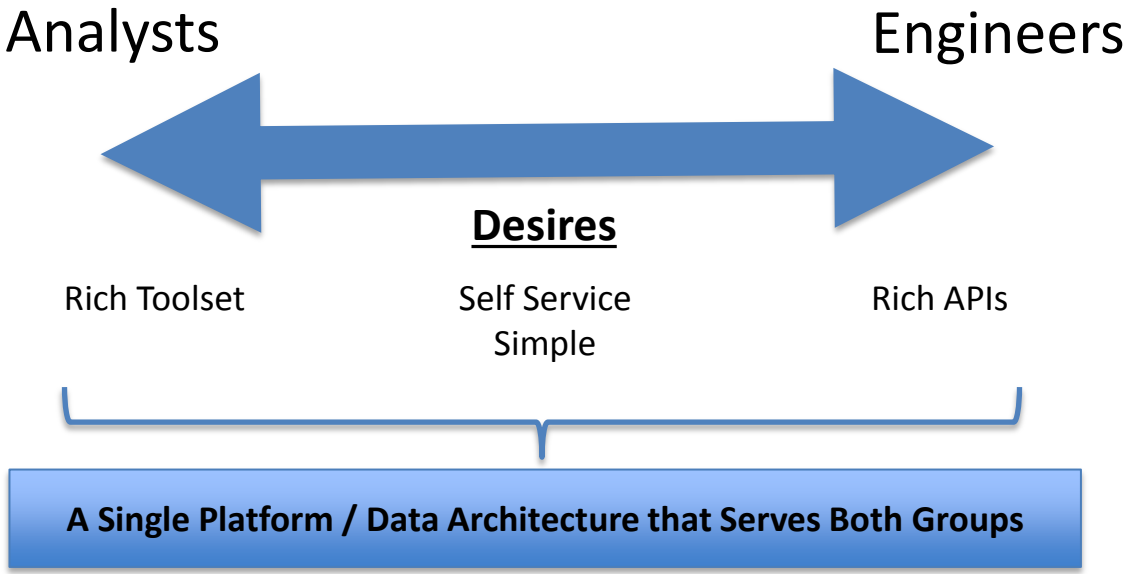
QCon SF

November 2013

Motivation

Data should be accessible, easy to discover,
and easy to process for everyone.

Big Data Users at Netflix



NETFLIX

Netflix Data Warehouse - Storage

S3 is the source of truth

- Decouples storage from processing.
- Persistent data; multiple/ transient Hadoop clusters

Data sources

- Event data from cloud services via Ursula/Honu
- Dimension data from Cassandra via Aegisthus

~100 billion events processed / day

Petabytes of data persisted and available to queries on S3.

NETFLIX

Netflix Data Platform - Processing

Long running clusters

sla and ad-hoc

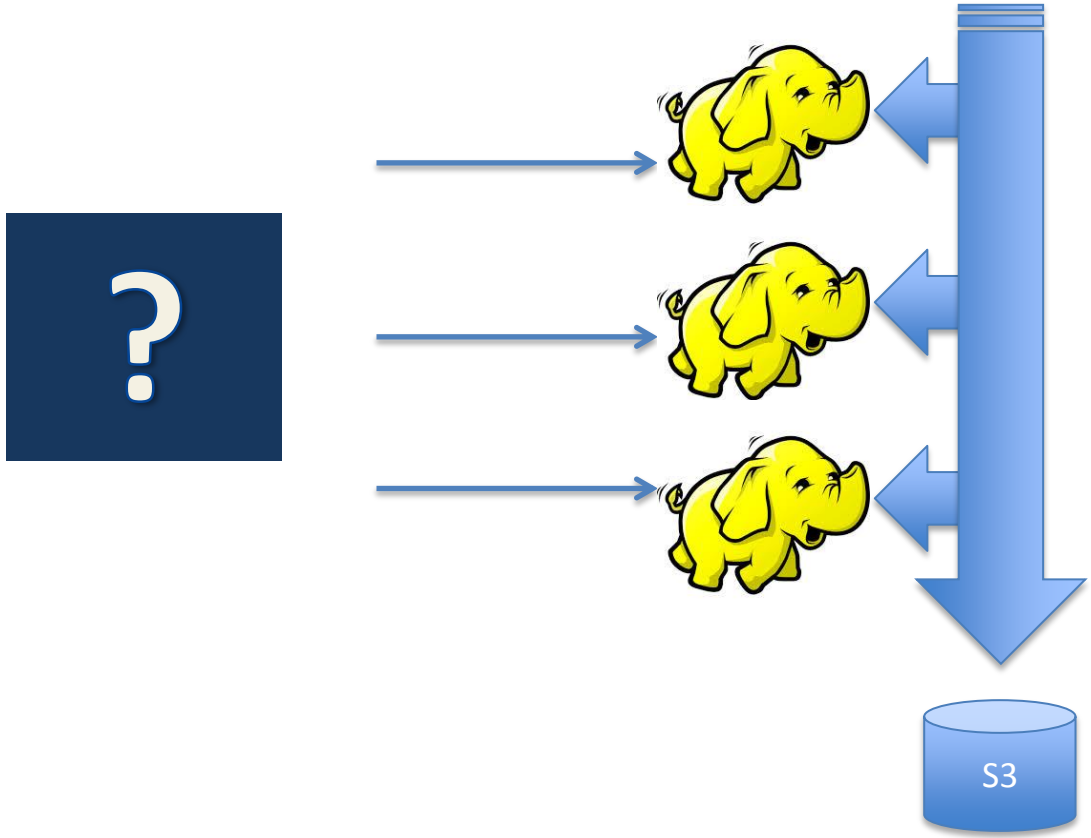
Supplemental nightly bonus clusters

For high priority ETL jobs

3,000+ instances in aggregate across the clusters

NETFLIX

Netflix Hadoop Platform as a Service

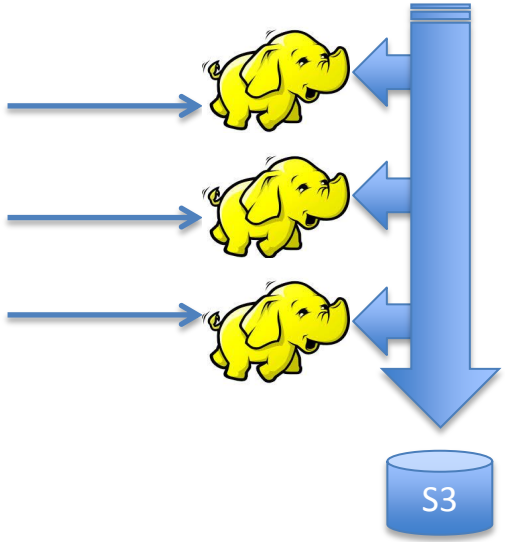


NETFLIX

Netflix Hadoop Platform as a Service

Complex backend infrastructures shouldn't yield complex interfaces to users.

?



Infrastructure desires:

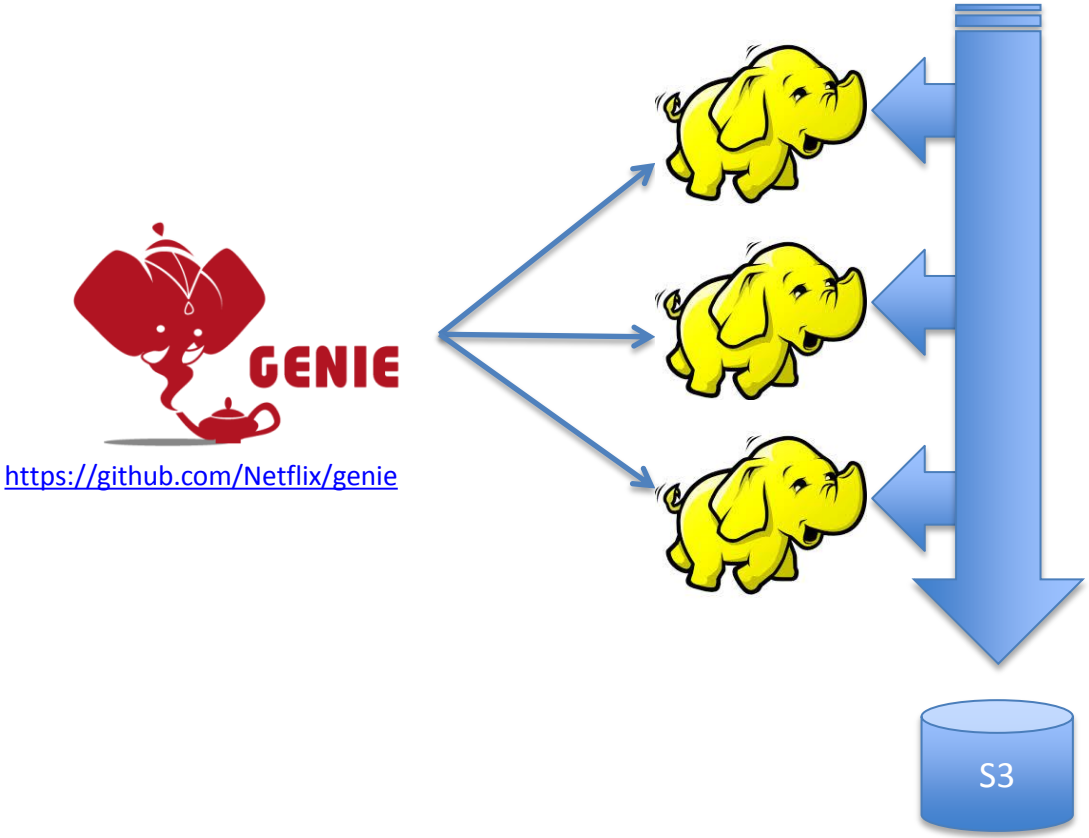
- Elastic, flexible, & non-disruptive scaling
- High degree of automation
- Loose coupling to platform customers

Users desire:

- Simplicity
- Adequate, scalable compute capacity
- Abstraction from physical details of backend clusters

NETFLIX

Netflix Hadoop Platform as a Service



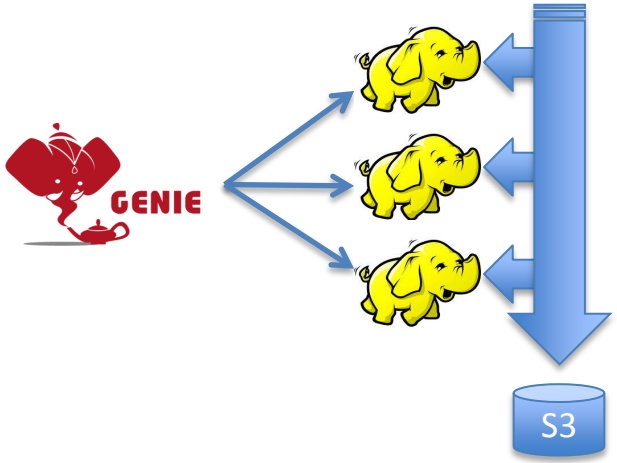
https://github.com/Netflix/genie

NETFLIX

Netflix Hadoop Platform as a Service

Genie service layer:

- Abstracts job submission to a complex backend into a simple REST interface.
- Cluster registration / job routing



Some Netflix Use Cases:

- “Red/Black” pushes
- Opportunistic provisioning of excess engineering capacity
- Easy experimentation with infrastructure

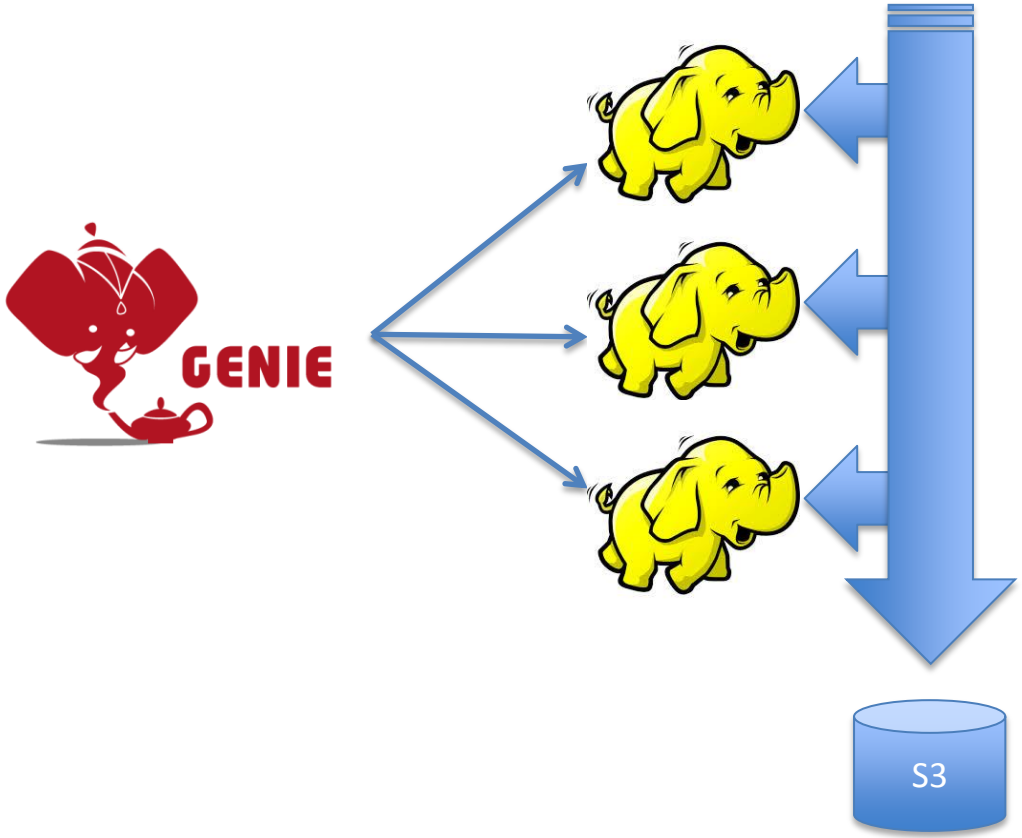
NETFLIX

Franklin

Data should be easy to use and discover.

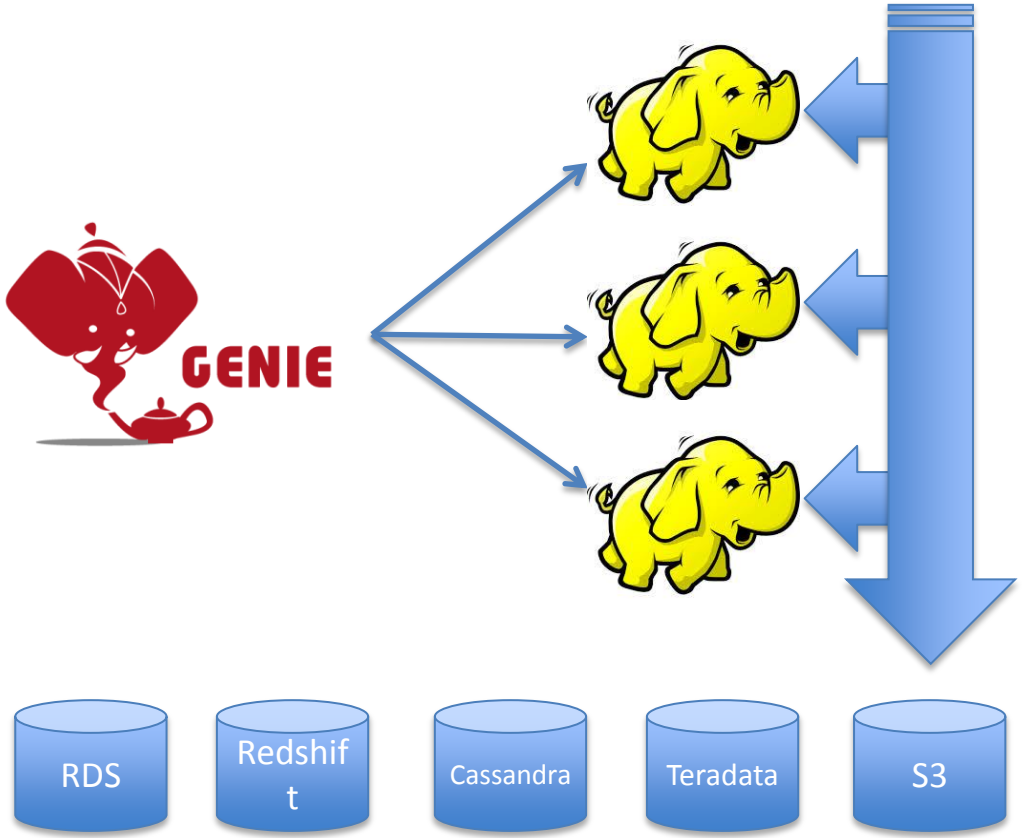
NETFLIX

Hadoop Platform as a Service



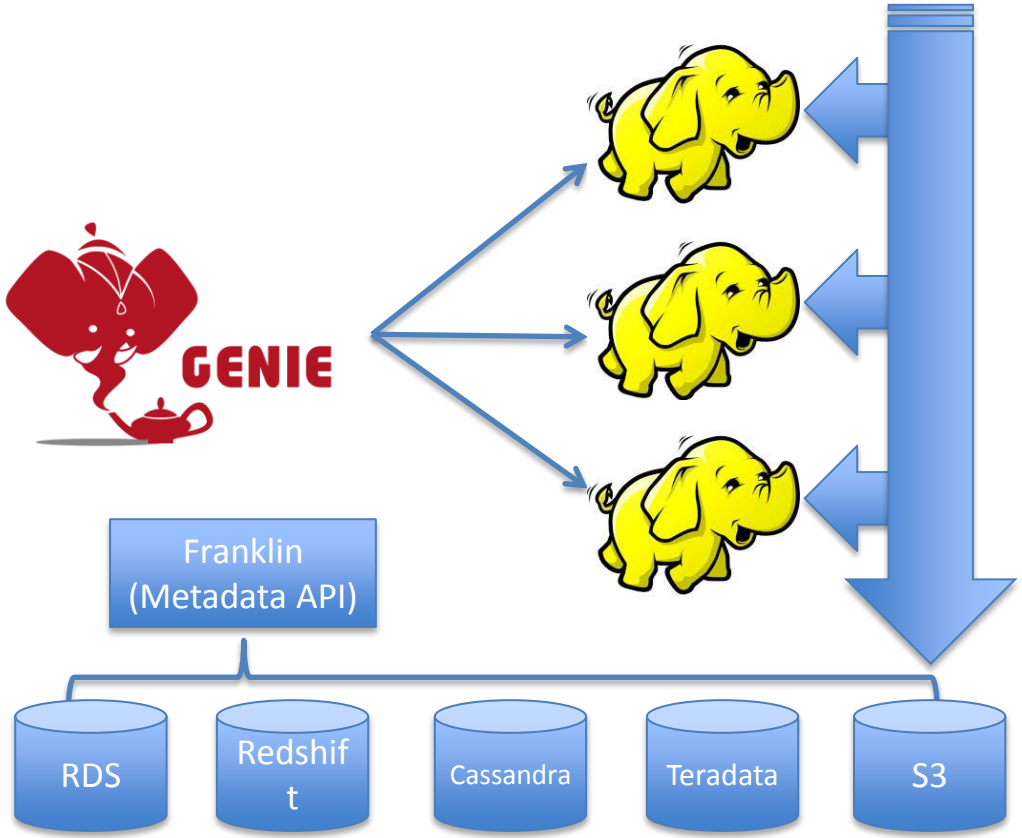
NETFLIX

Hadoop Platform as a Service



NETFLIX

Data Platform as a Service



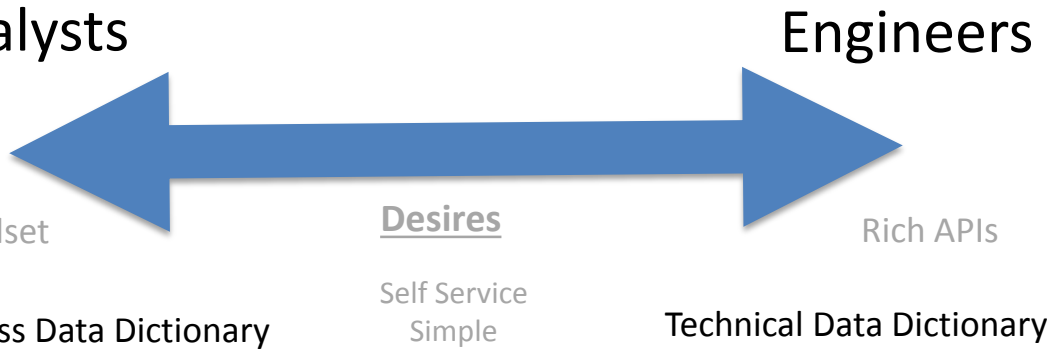
NETFLIX

Franklin

- Homogenizes the interface into multiple data stores.
  - Single API to program on
  - Single reference point for all data
- Abstracts name → (location, format, schema, …)
- Searchable catalog of all metadata of interest to analytics.
  - e.g. find all datasets with a field named ‘userid’

NETFLIX

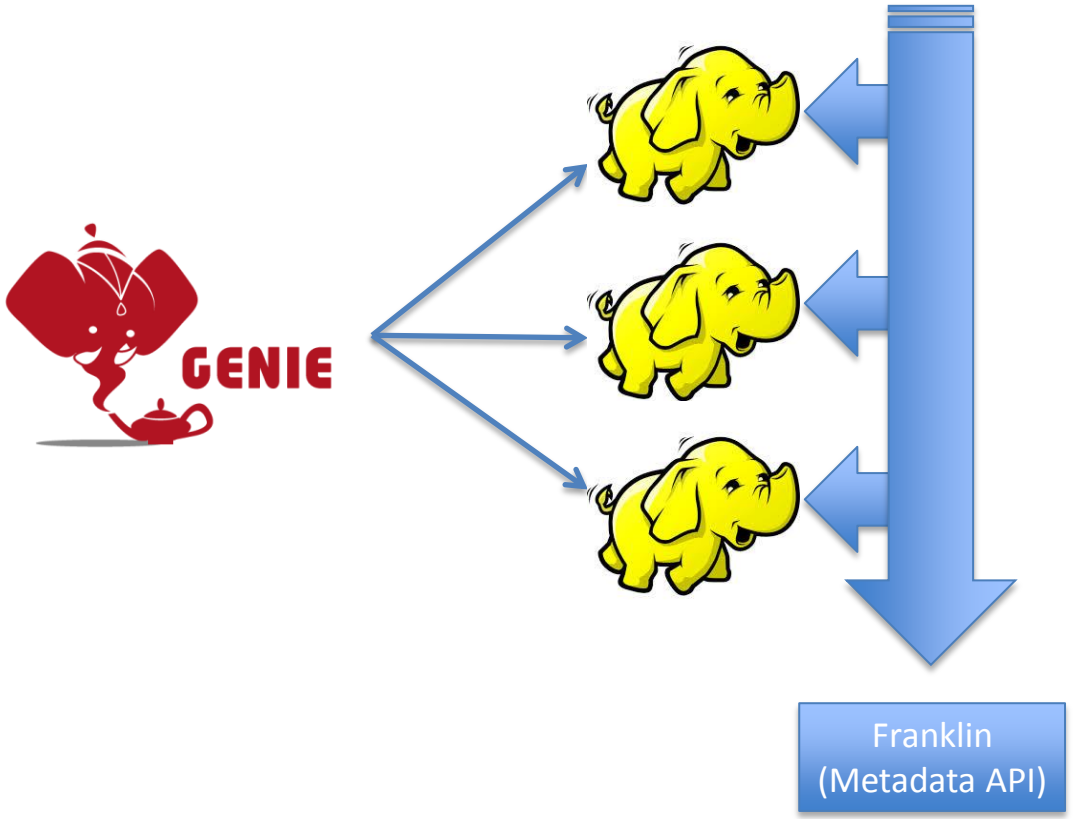
Franklin



Single Point of Reference for All Data
JavaScript GUI for one-off tasks

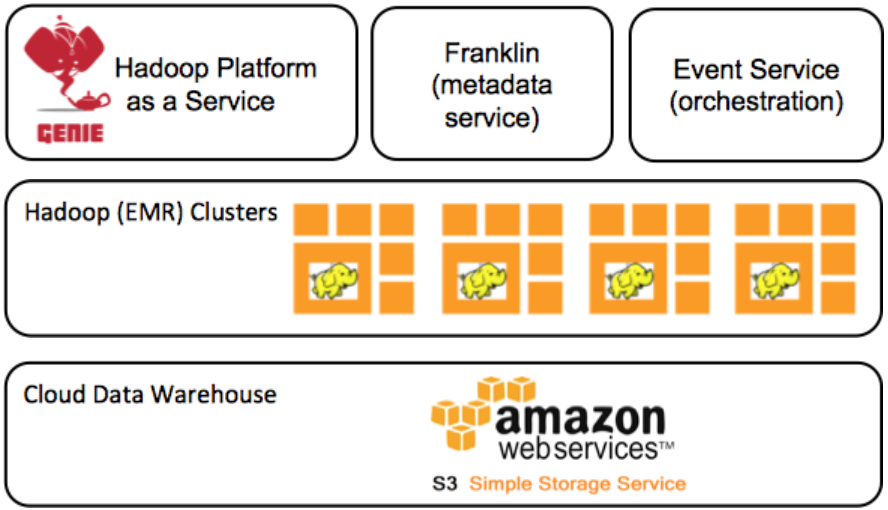
NETFLIX

Data Platform as a Service



NETFLIX

Netflix Data Platform – Primitive Service Layer



Primitive, decoupled services

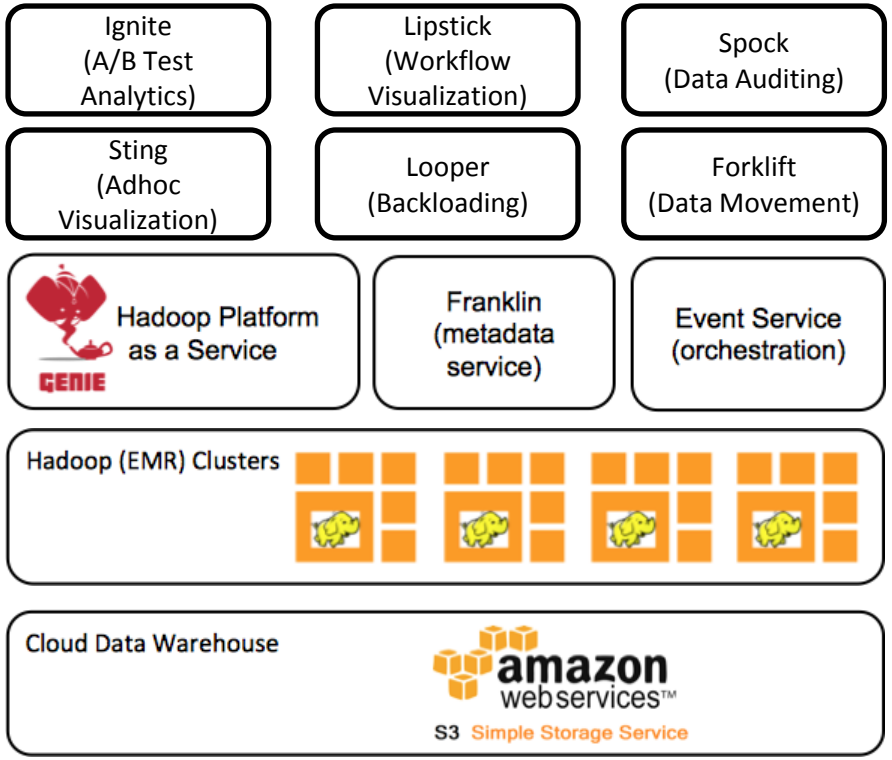
Building blocks for more complicated tools/services/apps

Serves 1000s of MapReduce Jobs / day

100+ jobs concurrently

NETFLIX

Netflix Data Platform – Tools



Heavily utilize services in the primitive layer.

Follow the same design philosophy as primitive apps:

- RESTful API
- Decoupled javascript interfaces

NETFLIX

Forklift

Data should be easy to move between sources.

NETFLIX

Why Forklift

- Frequent need to move around analytical data
  - Hive -> Reporting DBMS
  - DBMS -> S3
  - Hive -> R -> S3
- “Industrial” ETL
  - Highly automated
  - Reliable
  - Simple to use

NETFLIX

Forklift Design

- High level service leveraging
  - Genie
  - Franklin
- REST API
- GUI for adhoc requests
- Rich clients

```
Forklift().
 from(Hive('dse/title_dimension')).
 to(Teradata('dse/title_dimension')).
 executeOn(Genie())
```

NETFLIX

Sting

It should be fast and easy to explore and socialize data.

NETFLIX

Sting

- Caches results of Genie jobs in memory
- Sub second response to OLAP style operations
- Keeps datasets up to date
- Easy to use!

DSE Sting Graph Admin

Data environment: prod

Use prod data or test data

Hive Query

```
1  set hive.map.aggr=true;
2  set mapred.reduce.tasks=100;
3
4  select region_dateint,  region_hour, A.genre, sum(view_secs) / total_view_secs
5  from
6  (SELECT
7         analytic_genre_desc genre,
8         content_subtype_grain_desc subtype_grain,
9         content_subtype_desc subtype_desc,
10        content_type_desc content_type,
11        region_dateint,
12        region_hour,
13        standard_sanitized_duration_sec view_secs
14 FROM  dse.vhs_acct_device_ttl_sum vadts
15 join dse.ttl_title_d ttd on vadts.title_id = ttd.title_id
16 join dse.ttl_show_d tsd on ttd.show_title_id = tsd.show_title_id
```

Hive Query

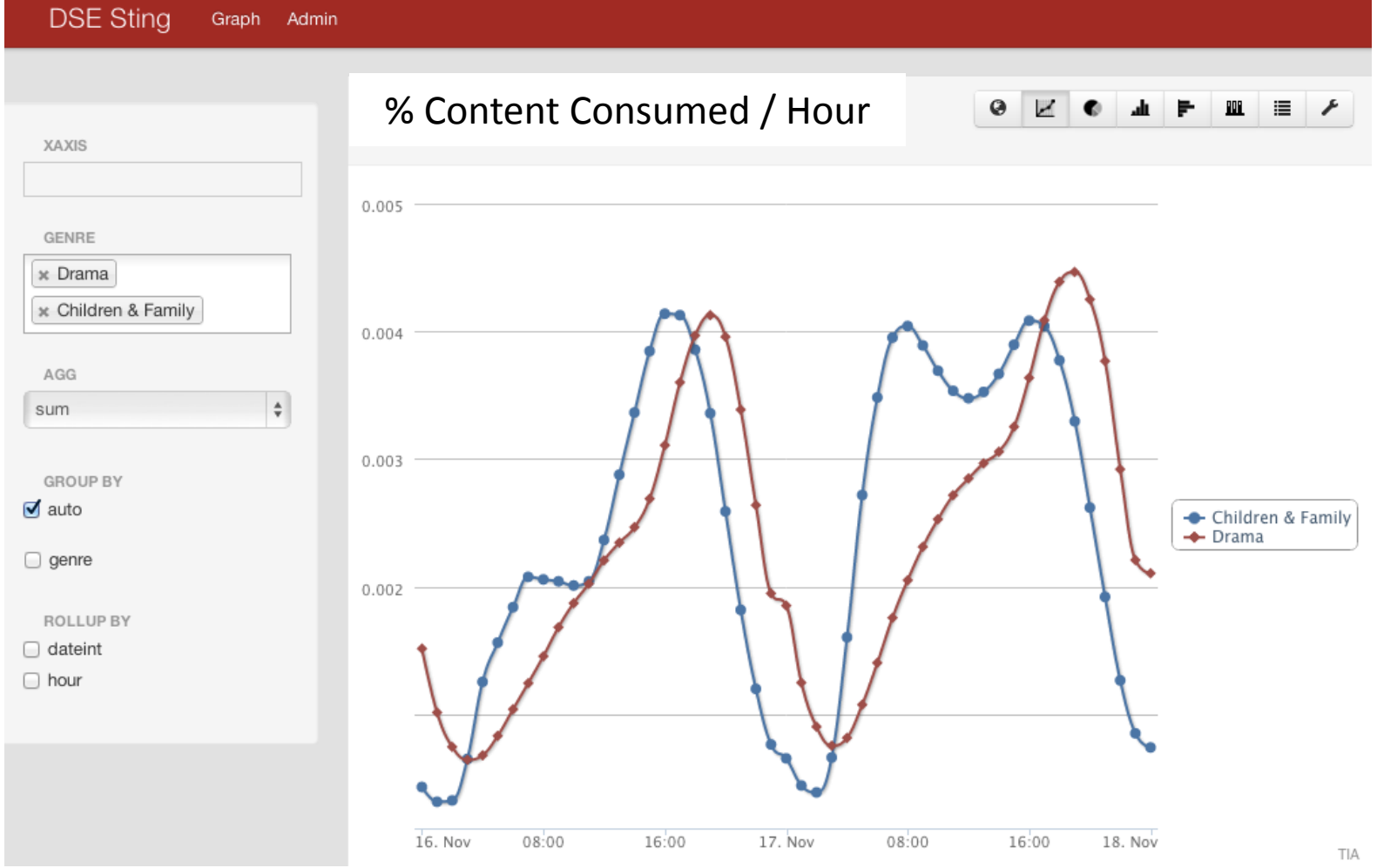
	field	type	default agg	default weight	data type	
✥	dateint	time				delete
✥	hour	time				delete
✥	genre	dimension				delete
✥	percentage	fact				delete

Schema should be entered in the order each field appears in the input row.

Schema

+ Add Field | Save Dataset

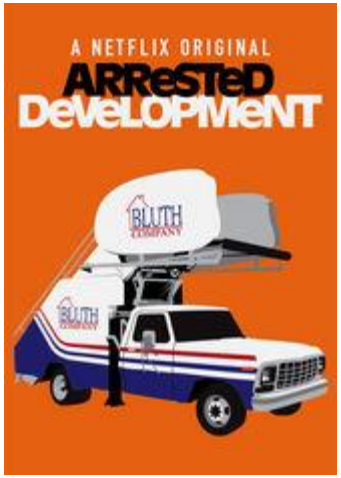
NETFLIX


DSE Sting
Graph
Admin
% Content Consumed / Hour
XAXIS
GENRE
Drama
Children & Family
AGG
sum
GROUP BY
auto
genre
ROLLUP BY
dateint
hour
0.005
0.004
0.003
0.002
16. Nov
08:00
16:00
17. Nov
08:00
16:00
18. Nov
Children & Family
Drama
TIA

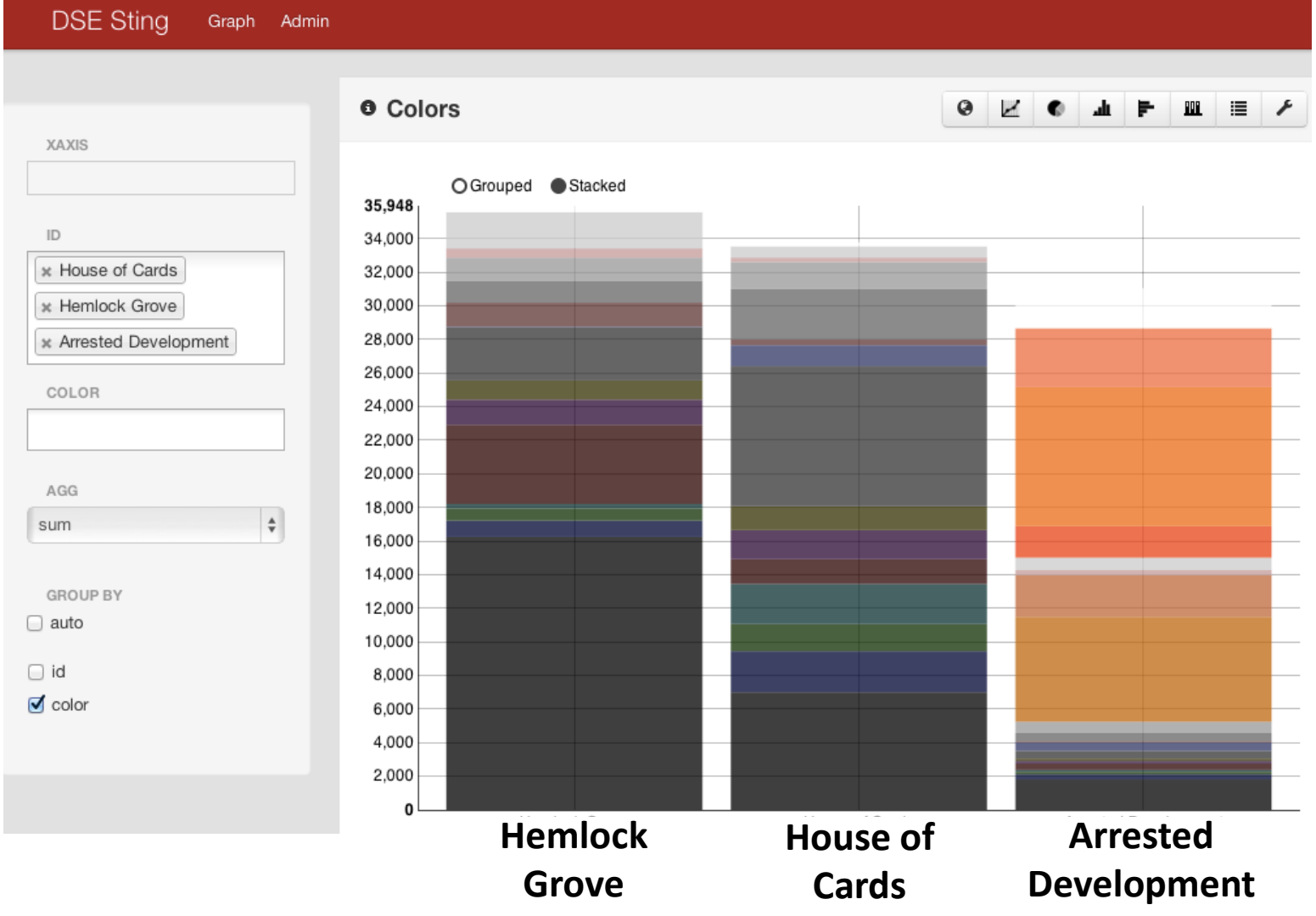
NETFLIX


NETFLIX ORIGINAL
HEMLOCK GROVE


HOUSE of CARDS
NETFLIX ORIGINAL


A NETFLIX ORIGINAL
ARRESTED DEVELOPMENT

NETFLIX


DSE Sting
Graph
Admin
XAXIS
ID
House of Cards
Hemlock Grove
Arrested Development
COLOR
AGG
sum
GROUP BY
auto
id
color
Colors
Grouped
Stacked
35,948
34,000
32,000
30,000
28,000
26,000
24,000
22,000
20,000
18,000
16,000
14,000
12,000
10,000
8,000
6,000
4,000
2,000
0
Hemlock Grove
House of Cards
Arrested Development

NETFLIX

Sting - backend

http://go/sting/graph/ContentPerHour/sum/percentage/?slicesize=20&sliceorder=desc


```
{
   "series":[
      {
         "data":[
                  [ 1381622400000, 0.0581890282854042 ],
                  [ 1381626000000, 0.0405827046800486 ],
                  [ 1382824800000, 0.0843322981735192 ],
                  [ 1382828400000, 0.0632406985680995 ]
         ],
         "name":"percentage",
         "xaxis":"_ds"
      }
   ]
}
```


NETFLIX

Lipstick

It should be easy to manipulate data and monitor progress of jobs on the infrastructure.

Pig and Hive at Netflix

- Hive
  - AdHoc queries
  - Lightweight aggregation
- Pig
  - Complex Dataflows / ETL
  - Data movement “glue” between complex operations

NETFLIX

What is Pig?

- A data flow language
- Simple to learn
  - Very few reserved words
  - Comparable to a SQL logical query plan
- Easy to extend and optimize
- Extendable via UDFs written in multiple languages
  - Java, Python, Ruby, Groovy, Javascript

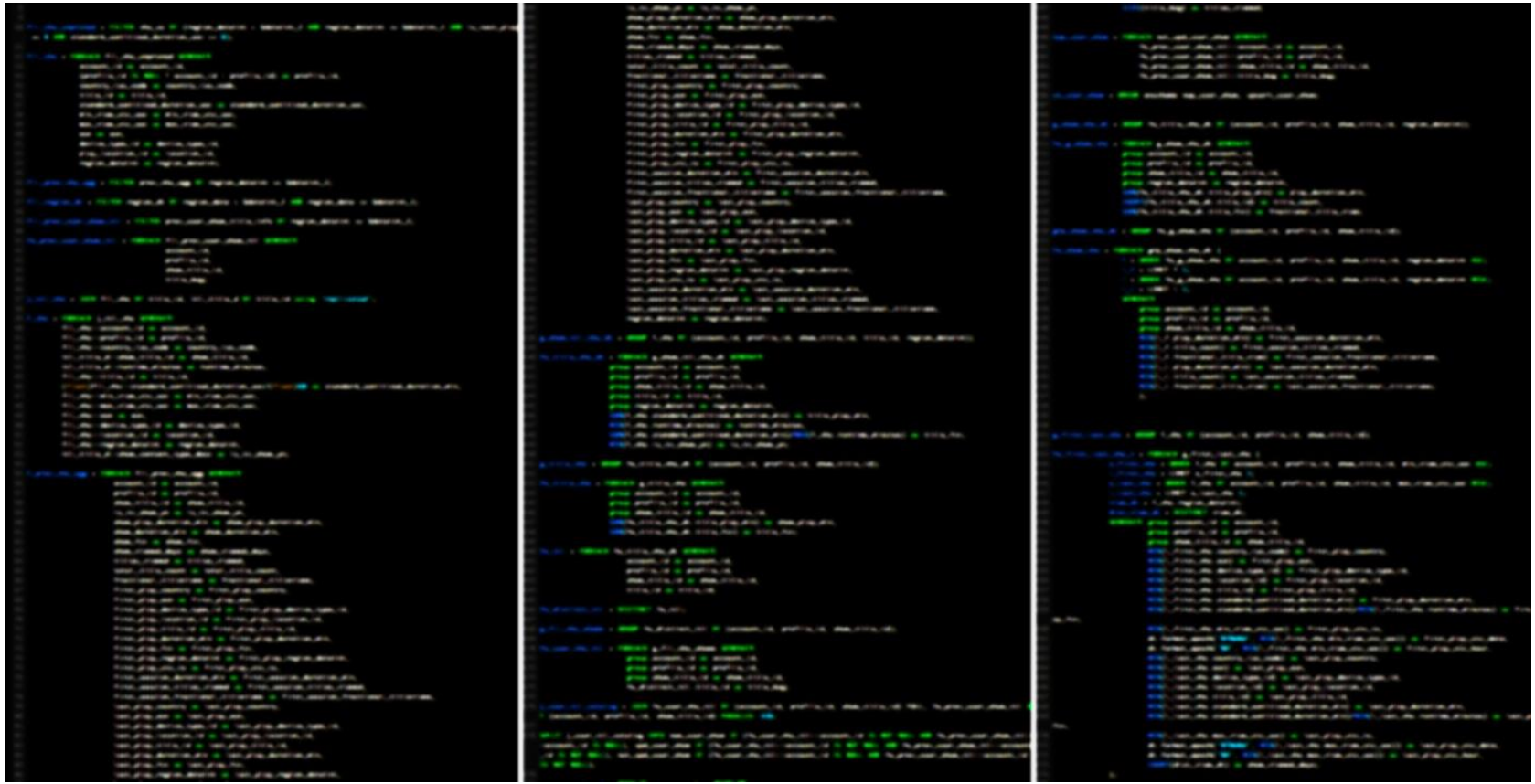
Sample Pig Script* (Word Count)

```
input_lines = LOAD '/tmp/my-copy-of-all-pages-on-internet' AS (line:chararray);

-- Extract words from each line and put them into a pig bag
-- datatype, then flatten the bag to get one word on each row
words = FOREACH input_lines GENERATE FLATTEN(TOKENIZE(line)) AS word;

-- filter out any words that are just white spaces
filtered_words = FILTER words BY word MATCHES '\\w+';

-- create a group for each word
word_groups = GROUP filtered_words BY word;

-- count the entries in each group
word_count = FOREACH word_groups GENERATE COUNT(filtered_words) AS count, group
AS word;

-- order the records by count
ordered_word_count = ORDER word_count BY count DESC;
STORE ordered_word_count INTO '/tmp/number-of-words-on-internet';
```

NETFLIX

* http://en.wikipedia.org/wiki/Pig_(programming_tool)#Example

A Typical Pig Script



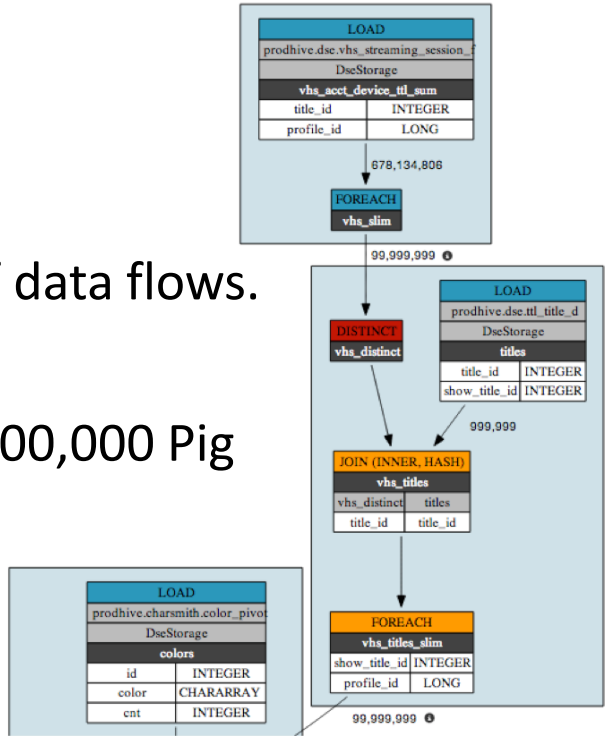
NETFLIX

Pig…

- Data flows are easy & flexible to express in text
  - Facilitates code reuse via UDFs and macros
  - Allows logical grouping of operations vs grouping by order of execution.
  - But errors are easy to make and overlook.
- Scripts can quickly get complicated
- Visualization quickly draws attention to:
  - Common errors
  - Execution order / logical flow
  - Optimization opportunities

Lipstick

- Generates graphical representations of data flows.
- Compatible with Apache Pig v11+
- Has been used to monitor more than 100,000 Pig jobs at Netflix

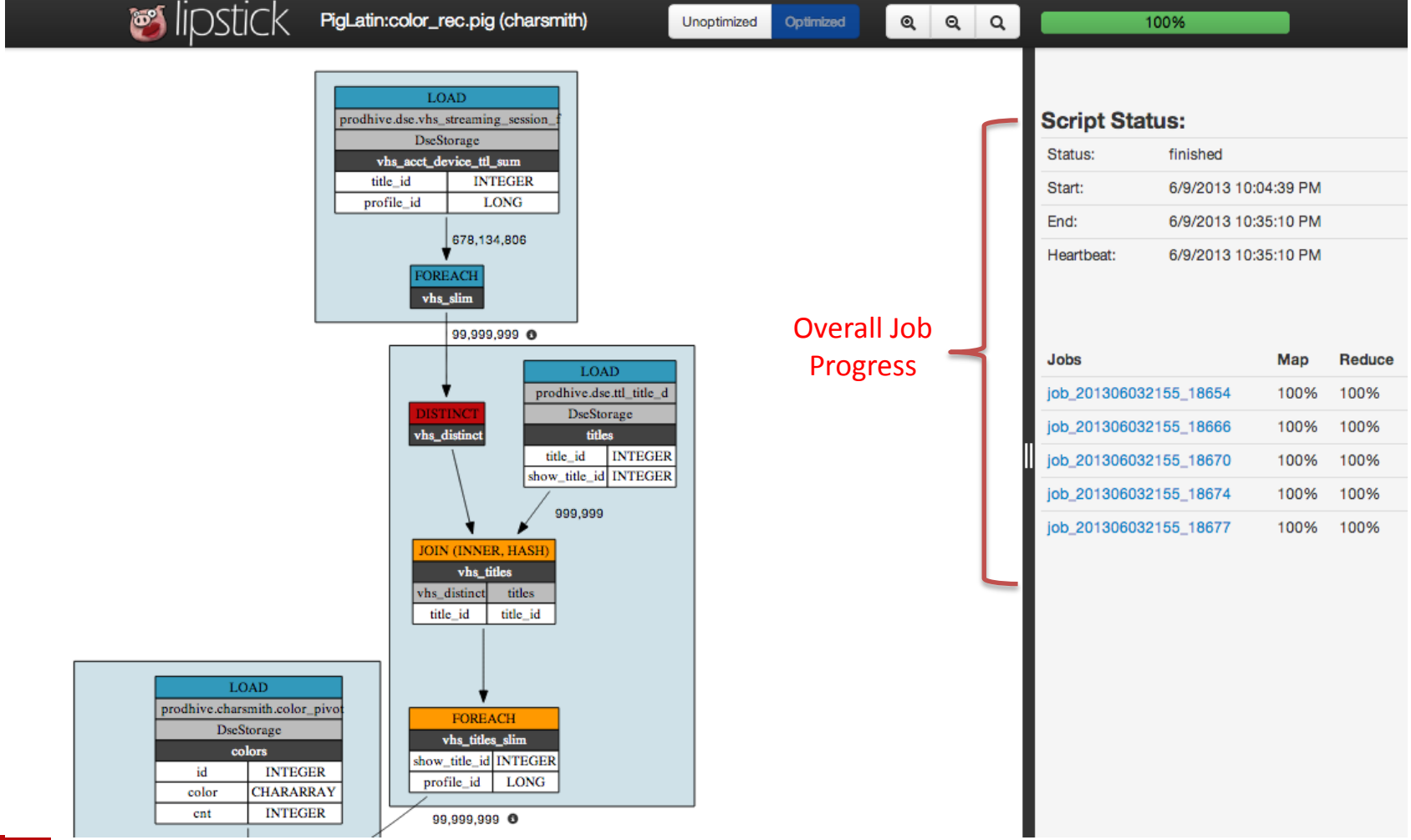


NETFLIX

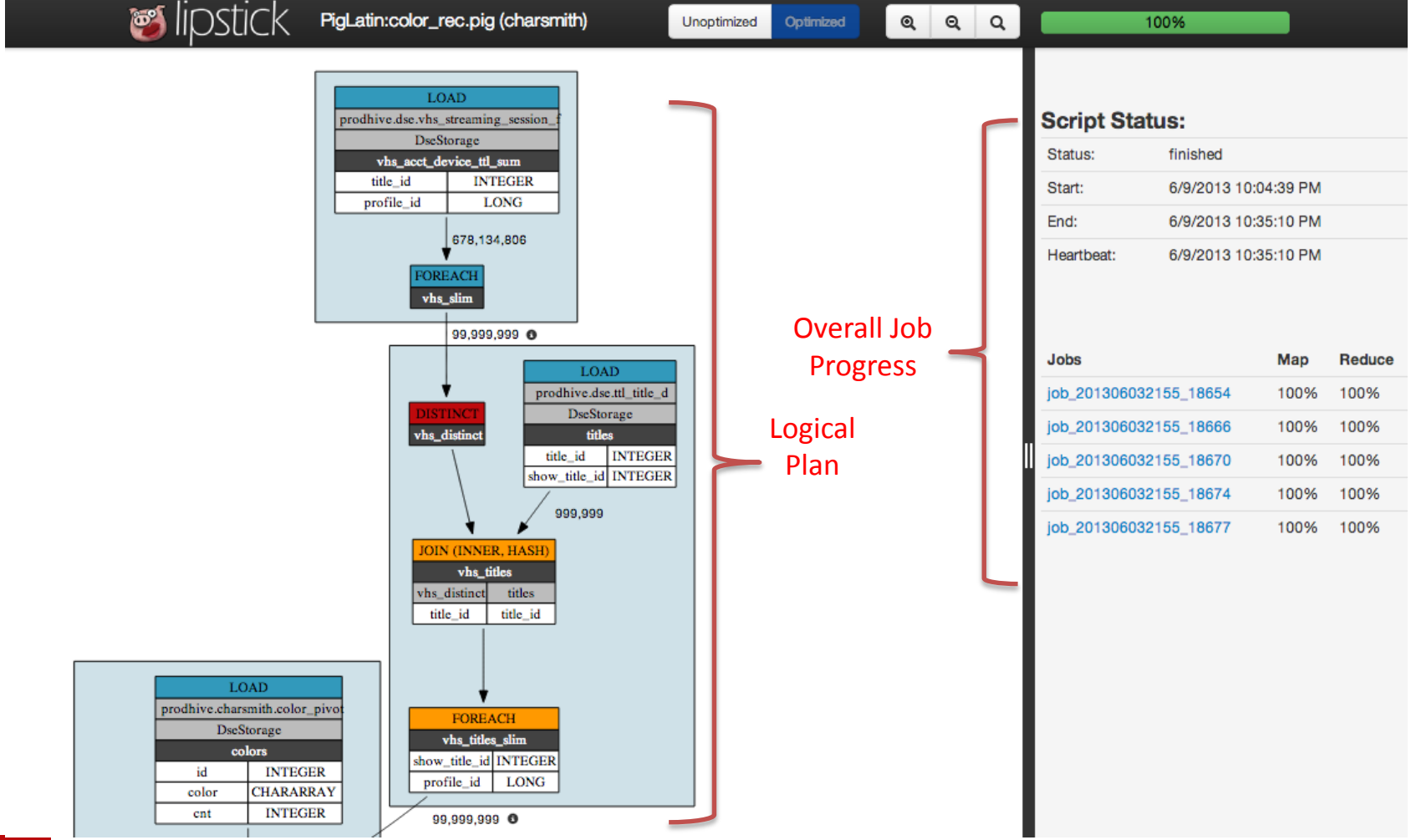
Lipstick

```
2013-06-07 22:53:23,365 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapRed
uceLayer.MapReduceLauncher - 25% complete
2013-06-07 22:53:24,234 [main] INFO  com.netflix.lipstick.pigtolipstick.BasicP2LClient -
Navigate to http://go/lipstick#job/10971d02-fe67-4f42-a5a0-6e246b870653 to view progress.
2013-06-07 22:53:36,398 [main] INFO  org.apache.pig.backend.hadoop.executionengine.mapRed
uceLayer.MapReduceLauncher - 50% complete
```

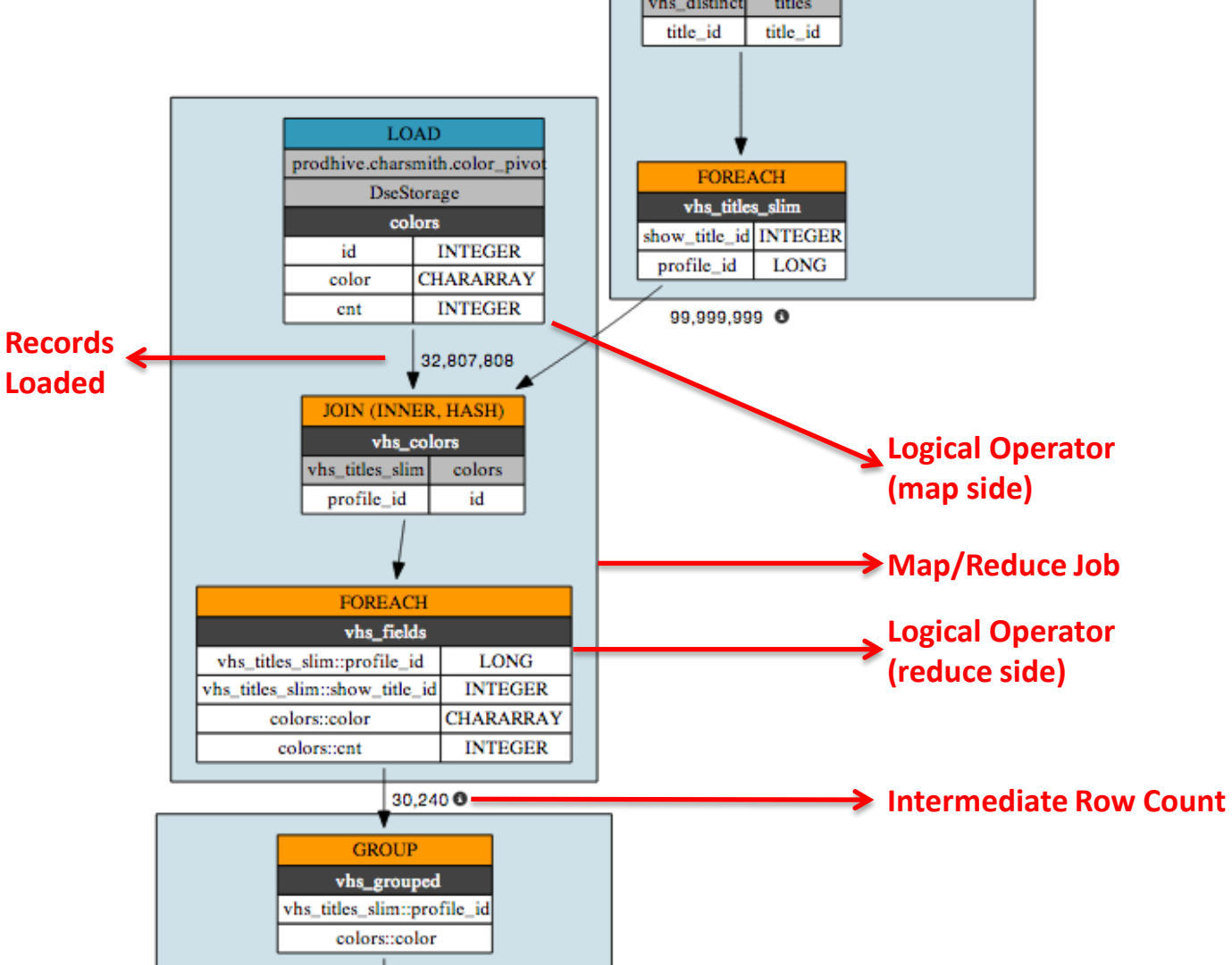
NETFLIX


lipstick
PigLatin:color_rec.pig (charsmith)
Unoptimized
Optimized
100%
LOAD
prodhive.dse.vhs_streaming_session_
DseStorage
vhs_acct_device_ttl_sum
title_id INTEGER
profile_id LONG
678,134,806
FOREACH
vhs_slim
99,999,999
LOAD
prodhive.dse.ttl_title_d
DseStorage
titles
title_id INTEGER
show_title_id INTEGER
DISTINCT
vhs_distinct
999,999
JOIN (INNER, HASH)
vhs_titles
vhs_distinct titles
title_id title_id
LOAD
prodhive.charsmith.color_pivot
DseStorage
colors
id INTEGER
color CHARARRAY
cnt INTEGER
FOREACH
vhs_titles_slim
show_title_id INTEGER
profile_id LONG
99,999,999
Overall Job Progress
Script Status:
Status: finished
Start: 6/9/2013 10:04:39 PM
End: 6/9/2013 10:35:10 PM
Heartbeat: 6/9/2013 10:35:10 PM
Jobs Map Reduce
job_201306032155_18654 100% 100%
job_201306032155_18666 100% 100%
job_201306032155_18670 100% 100%
job_201306032155_18674 100% 100%
job_201306032155_18677 100% 100%

NETFLIX


lipstick
PigLatin:color_rec.pig (charsmith)
Unoptimized
Optimized
100%
LOAD
prodhive.dse.vhs_streaming_session_
DseStorage
vhs_acct_device_ttl_sum
title_id INTEGER
profile_id LONG
678,134,806
FOREACH
vhs_slim
99,999,999
LOAD
prodhive.dse.ttl_title_d
DseStorage
titles
title_id INTEGER
show_title_id INTEGER
DISTINCT
vhs_distinct
999,999
JOIN (INNER, HASH)
vhs_titles
vhs_distinct titles
title_id title_id
LOAD
prodhive.charsmith.color_pivot
DseStorage
colors
id INTEGER
color CHARARRAY
cnt INTEGER
FOREACH
vhs_titles_slim
show_title_id INTEGER
profile_id LONG
99,999,999
Overall Job Progress
Logical Plan
Script Status:
Status: finished
Start: 6/9/2013 10:04:39 PM
End: 6/9/2013 10:35:10 PM
Heartbeat: 6/9/2013 10:35:10 PM
Jobs Map Reduce
job_201306032155_18654 100% 100%
job_201306032155_18666 100% 100%
job_201306032155_18670 100% 100%
job_201306032155_18674 100% 100%
job_201306032155_18677 100% 100%

NETFLIX


title_id
title_id
FOREACH
vhs_titles_slim
show_title_id INTEGER
profile_id LONG
99,999,999
LOAD
prodhive.charsmith.color_pivot
DseStorage
colors
id INTEGER
color CHARARRAY
cnt INTEGER
32,807,808
Records Loaded
JOIN (INNER, HASH)
vhs_colors
vhs_titles_slim colors
profile_id id
Logical Operator (map side)
Map/Reduce Job
FOREACH
vhs_fields
vhs_titles_slim::profile_id LONG
vhs_titles_slim::show_title_id INTEGER
colors::color CHARARRAY
colors::cnt INTEGER
Logical Operator (reduce side)
30,240
Intermediate Row Count
GROUP
vhs_grouped
vhs_titles_slim::profile_id
colors::color

NETFLIX

Lipstick for Fast Development

- During development:
  - Keep track of data flow
  - Spot common errors
  - Easily estimate and optimize complexity

NETFLIX

Lipstick for Job Monitoring

- During execution:
  - Graphically monitor execution status from a single console
  - Spot optimization opportunities
    - Map vs reduce side joins
    - Data skew
    - Better parallelism settings

NETFLIX

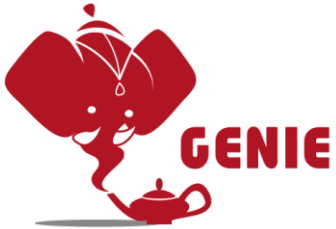
Lipstick for Support

- Empowers users to support themselves
  - Better operational visibility
  - Examine intermediate output of jobs
  - One stop shop for job information
- Facilitates communication between infrastructure / support teams and end users
  - Lipstick link contains all information needed to provide support.

NETFLIX

Big Data + NetflixOSS

- Check out all the NetflixOSS tools at http://netflix.github.io/



http://github.com/Netflix/genie

http://github.com/Netflix/Lipstick

NETFLIX

Thank you!

- Jeff Magnusson:

jmagnusson@netflix.com | http://www.linkedin.com/in/jmagnuss |@jeffmagnusson

Jobs: http://jobs.netflix.com
Netflix OSS: http://netflix.github.io
Tech Blog: http://techblog.netflix.com/

NETFLIX