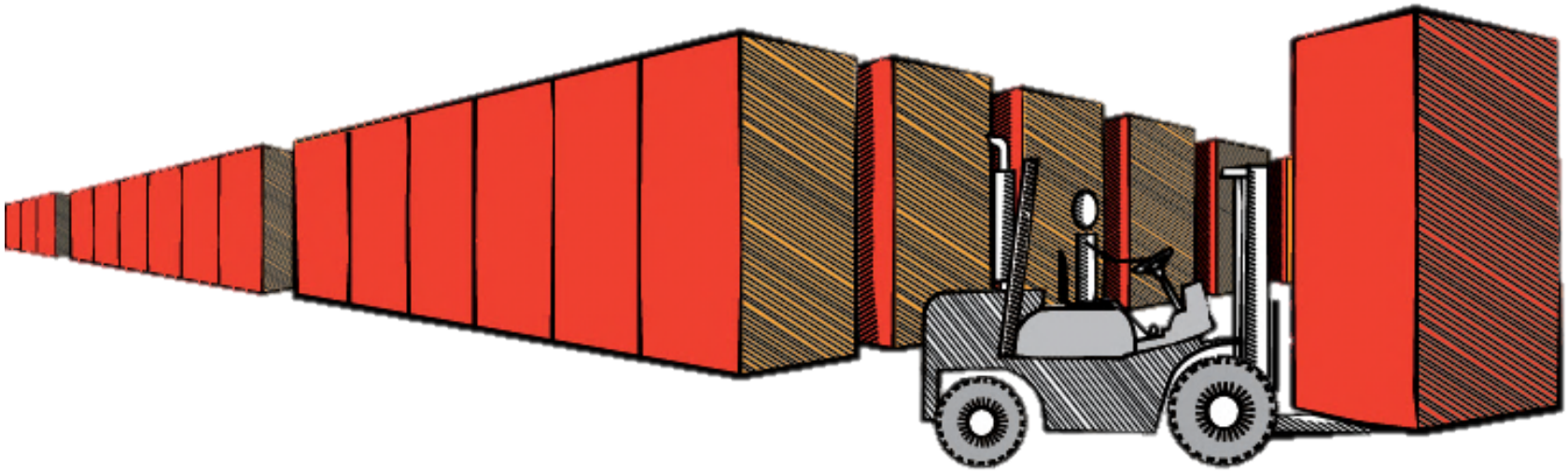
AWS re:Invent

Getting Maximum Performance from Amazon Redshift: High Concurrency, Fast Data Loads, and Complex Queries

November 13, 2013

amazon web services

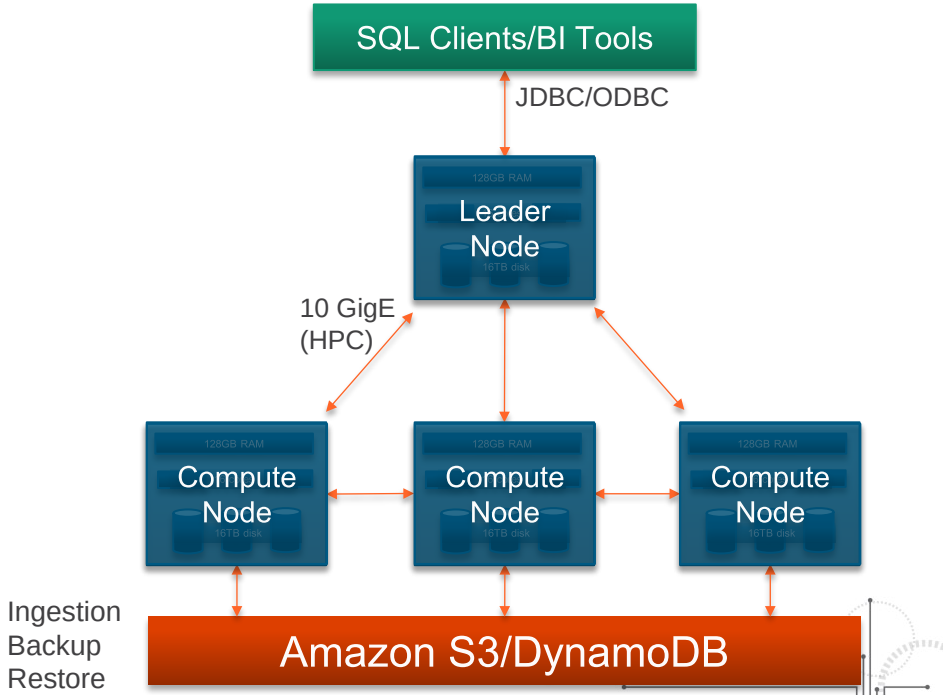
Amazon Redshift



Fast, simple, petabyte-scale data warehousing for less than $1,000/TB/Year

Amazon Redshift architecture

- Leader Node
  - SQL endpoint
  - Stores metadata
  - Coordinates query execution
- Compute Nodes
  - Local, columnar storage
  - Execute queries in parallel
  - Load, backup, restore via Amazon S3
  - Parallel load from Amazon DynamoDB
- Single node version available



Amazon Redshift is priced to let you analyze all your data

	Price Per Hour for HS1.XL Single Node	Effective Hourly Price per TB	Effective Annual Price per TB
On-Demand	$ 0.850	$ 0.425	$ 3,723
1 Year Reservation	$ 0.500	$ 0.250	$ 2,190
3 Year Reservation	$ 0.228	$ 0.114	$ 999

Simple Pricing

Number of Nodes x Cost per Hour

No charge for Leader Node

No upfront costs

Pay as you go

AWS re:Invent

Getting Maximum Performance from Amazon Redshift: High Concurrency

Ben Myles, Desk.com (@benmyles)

November 13, 2013

amazon web services

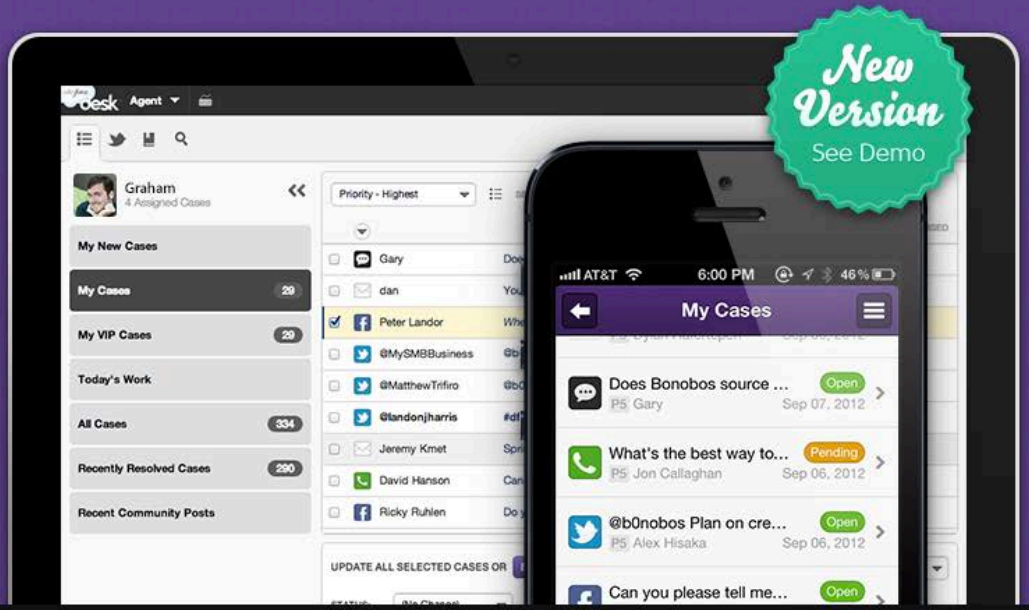
salesforce desk

FEATURES PRICING CUSTOMERS RESOURCES HELP LOGIN GET STARTED FOR FREE

Delight your customers with awesome Customer Support

TRY IT FOR FREE TODAY No credit card required.

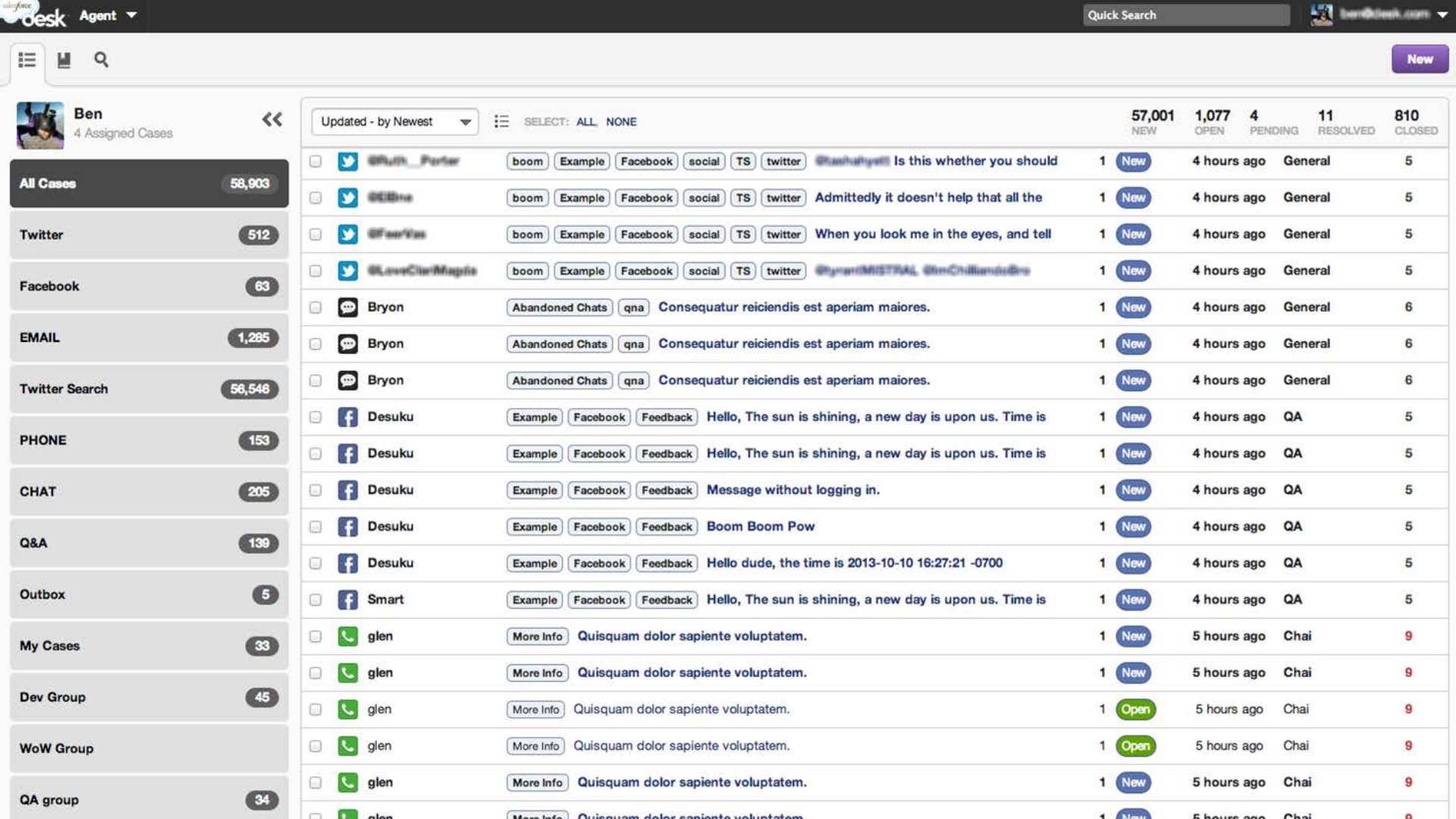
GET STARTED ▸ REQUEST A DEMO



"With everything that we do, Desk.com makes us better."

Square

Fab. Square rdio Eventbrite TED yelp PANDORA


desk
Agent
Quick Search
New
Ben
4 Assigned Cases
All Cases 58,903
Twitter 512
Facebook 63
EMAIL 1,285
Twitter Search 56,546
PHONE 153
CHAT 205
Q&A 139
Outbox 5
My Cases 33
Dev Group 45
WoW Group
QA group 34
Updated - by Newest
SELECT: ALL NONE
57,001 NEW
1,077 OPEN
4 PENDING
11 RESOLVED
810 CLOSED
boom Example Facebook social TS twitter Is this whether you should 1 New 4 hours ago General 5
boom Example Facebook social TS twitter Admittedly it doesn't help that all the 1 New 4 hours ago General 5
boom Example Facebook social TS twitter When you look me in the eyes, and tell 1 New 4 hours ago General 5
boom Example Facebook social TS twitter 1 New 4 hours ago General 5
Bryon Abandoned Chats qna Consequatur reiciendis est aperiam maiores. 1 New 4 hours ago General 6
Bryon Abandoned Chats qna Consequatur reiciendis est aperiam maiores. 1 New 4 hours ago General 6
Bryon Abandoned Chats qna Consequatur reiciendis est aperiam maiores. 1 New 4 hours ago General 6
Desuku Example Facebook Feedback Hello, The sun is shining, a new day is upon us. Time is 1 New 4 hours ago QA 5
Desuku Example Facebook Feedback Hello, The sun is shining, a new day is upon us. Time is 1 New 4 hours ago QA 5
Desuku Example Facebook Feedback Message without logging in. 1 New 4 hours ago QA 5
Desuku Example Facebook Feedback Boom Boom Pow 1 New 4 hours ago QA 5
Desuku Example Facebook Feedback Hello dude, the time is 2013-10-10 16:27:21 -0700 1 New 4 hours ago QA 5
Smart Example Facebook Feedback Hello, The sun is shining, a new day is upon us. Time is 1 New 4 hours ago QA 5
glen More Info Quisquam dolor sapiente voluptatem. 1 New 5 hours ago Chai 9
glen More Info Quisquam dolor sapiente voluptatem. 1 New 5 hours ago Chai 9
glen More Info Quisquam dolor sapiente voluptatem. 1 Open 5 hours ago Chai 9
glen More Info Quisquam dolor sapiente voluptatem. 1 Open 5 hours ago Chai 9
glen More Info Quisquam dolor sapiente voluptatem. 1 New 5 hours ago Chai 9

desk Agent

Quick Search

CASE | CUSTOMER | COMPANY | INFO LINKS

SUBJECT: Nihil non eum optio hic.

STATUS: Open

PRIORITY: 4

GROUP: Development

AGENT: (Unassigned)

DESCRIPTION: Asperiores ad sed nobis eum beatae. Expedita id consequatur voluptates enim aut quia. Dolore atque voluptatibus est. Illo non totam corrupti commodi nobis sit. Nisi magni aut velit.

LABELS

New Email

DIRECT CASE LINK

https://reporting.deskqa7.com/agent

0 ATTACHMENTS Add

9 OTHER OPEN CASES FOR THIS CUSTOMER View All

EMAIL | CUSTOMER HISTORY (10) | COMPANY HISTORY

Nihil non eum optio hic.

To: (Empty)

From:

Asperiores ad sed nobis eum beatae. Expedita id consequatur voluptates enim aut quia. Dolore atque voluptatibus est. Illo non totam corrupti c magni aut velit.

HIDE

You can format your message with Textile Syntax (formatting help here)

ADD CC ADD BCC ADD ATTACHMENT LESS

Add

SUBJECT: Re: Nihil non eum optio hic.

TO

CC

BCC

ATTACHMENTS

APPLY MACRO | Close Tab | Update

Business Insights

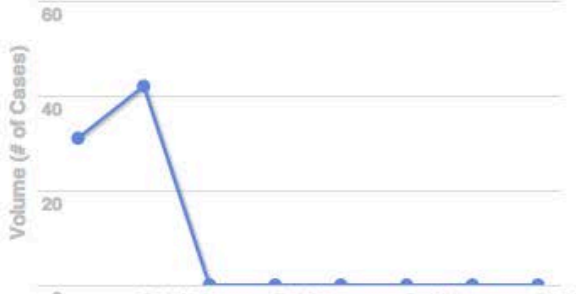
- **Dashboard**
- Cases
- Performance

OCT 17, 2013 - OCT 24, 2013

1st Contact Resolution	Case Reopen Rate	Average Handle Time	Time to 1st Response	Time to 1st Resolution
0%	0%	19.4min	12.9d	15.8hrs
high: 0% start: 0%	high: 0% start: 0%	high: 115.1 start: 0.1	high: 12.9 start: 0	high: 172.9 start: 172.9
low: 0% end: 0%	low: 0% end: 0%	low: 0 end: 0.1	low: 0 end: 0	low: 0 end: 0

Case Volume Overview

by New Cases



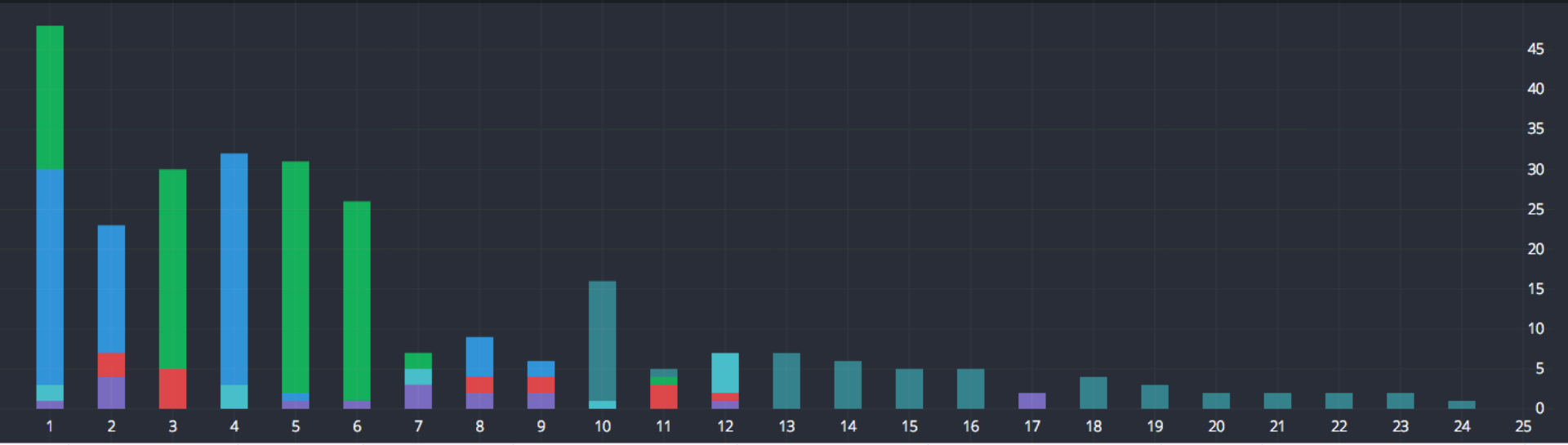
Response Time vs. Resolution Time

Calendar Hours | Business Hours | Compare All Hours

Agents Report

09/20/2013 - 09/24/2013

Active Segment: Channels | Chat | Twitter | Email | Q&A | Phone | Facebook

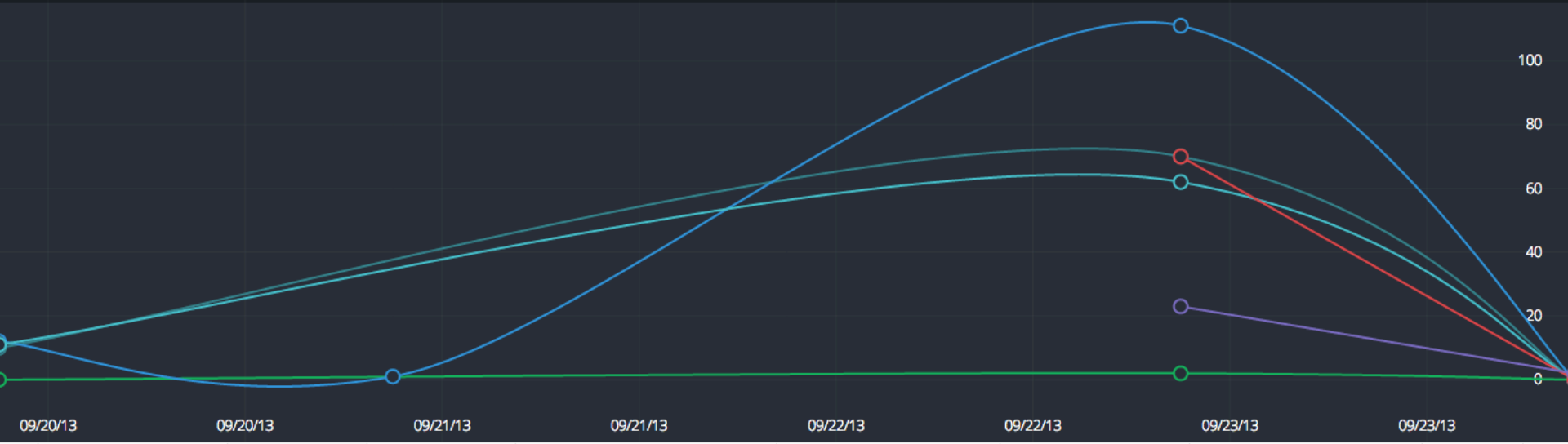


Agents Report Name	0:26:49 Time Online	444 Resolved Cases	0.2 Agent Replies Per Resolve	0:00:10 Average Handle Time	279:15:33 Average Time to Resolution	23.5% FCR Rate	0.0% % Cases Resolved with a Macro
1 Eric	0:10:13	84	0.8	0:00:32	792:35:53	80.0%	0.0%
2 Chai	8:40:25	53	0.3	0:00:04	469:49:28	50.0%	0.0%
3 Jeremy	0:07:11	46	0.0	0:00:00	393:58:54	0.0%	0.0%

Overview Report

09/20/2013 - 09/24/2013

Active Segment: Channels | Chat | Twitter | Email | Q&A | Phone | Facebook



Overview Report Date/Time	374 Created Cases	698 Resolved Cases	240:55:16 Average Time to First Response	0.1 Agent Replies Per Resolve	0:00:10 Average Handle Time	296:37:49 Average Time to Resolution	29.2% FCR Rate
Sun, 09/22/2013	338	433	651:03:41	0.0	0:00:34	573:51:53	66.7%
Thu, 09/19/2013	33	11	0:00:00	0.5	0:00:00	71:30:46	0.0%
Mon, 09/23/2013	2	254	312:37:26	0.0	0:00:05	541:08:39	50.0%

Our Data

- Primarily raw events
  - E.g., ‘case closed’, ‘case received’, ‘case resolved’
- Nearly 3B event rows
- About 200K+ new events per hour

Amazon Redshift Environment

- **Read** Cluster: 2+1 Node 8XL
- **Write** Cluser: 2+1 Node XL
- < 1TB disk space used (compression ftw!)
- High CPU and I/O utilization

Amazon Redshift Performance

- User-facing portal

Requirements:

- All queries *must* execute in < 30 seconds
- **Target is < 5 seconds**

Amazon Redshift Performance

Technique 1: Less Processing at Query Time

- Generate *events from events* to simplify queries
- Example: case_first_open from case_open
- *However*, increases resources consumed and time taken by hourly data updates

Amazon Redshift Performance

```
SELECT
  DATE_TRUNC('day', e.created_at) AS t,
  CASE WHEN event_type =                1 THEN
      COUNT(*) ELSE NULL END AS case_opens,
  CASE WHEN event_type = /*case_first_open*/2 THEN
      COUNT(*) ELSE NULL END AS case_first_opens
FROM events e
WHERE ...
```

Amazon Redshift Performance

Technique 2: Read + Write Cluster

- Originally had single cluster
- Each hourly data update used significant CPU + I/O and ran for ~30mins
- Huge impact on query performance

Amazon Redshift Performance

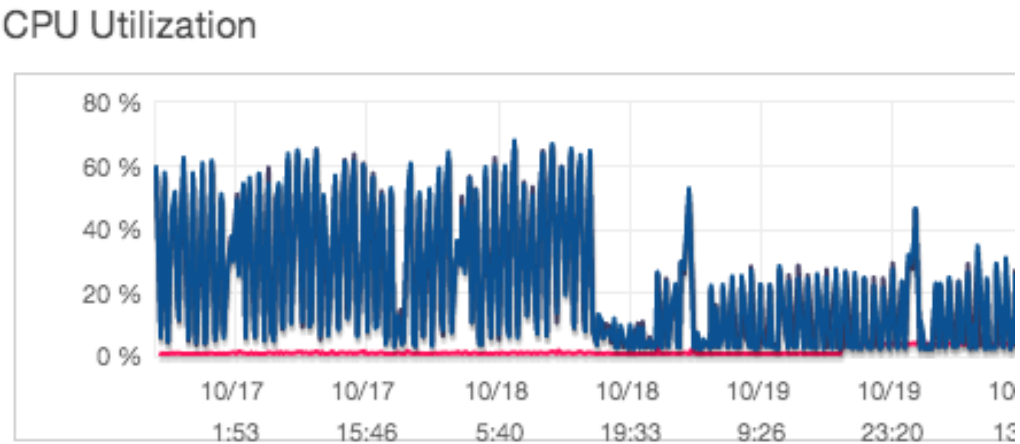
Technique 2: Read + Write Cluster

- Added a ‘Write’ cluster just for processing data updates

MySQL -> S3 ->
‘Write’ Cluster -> S3 -> ‘Read’ Cluster

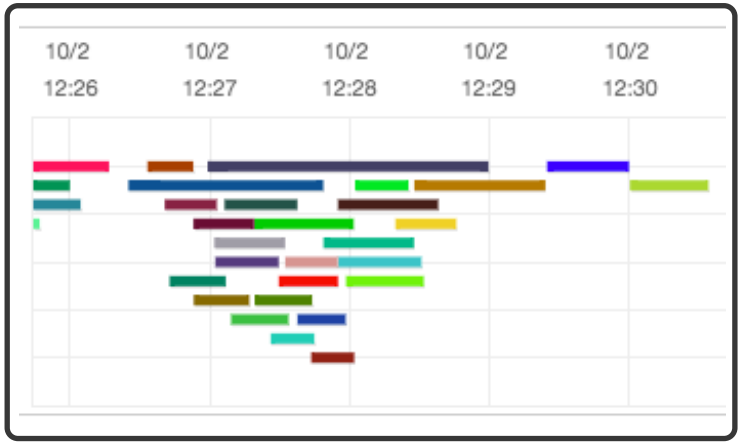
Amazon Redshift Performance

Technique 2: Read + Write Cluster



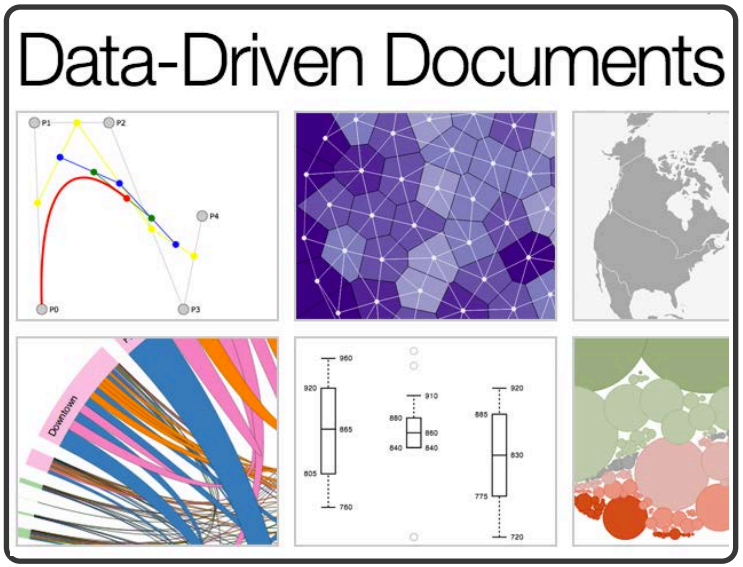
Concurrency Challenges

- Queries execute when end-users load reports
- Max of 15 concurrent queries in Amazon Redshift
- Single user rapidly hitting refresh could have big impact

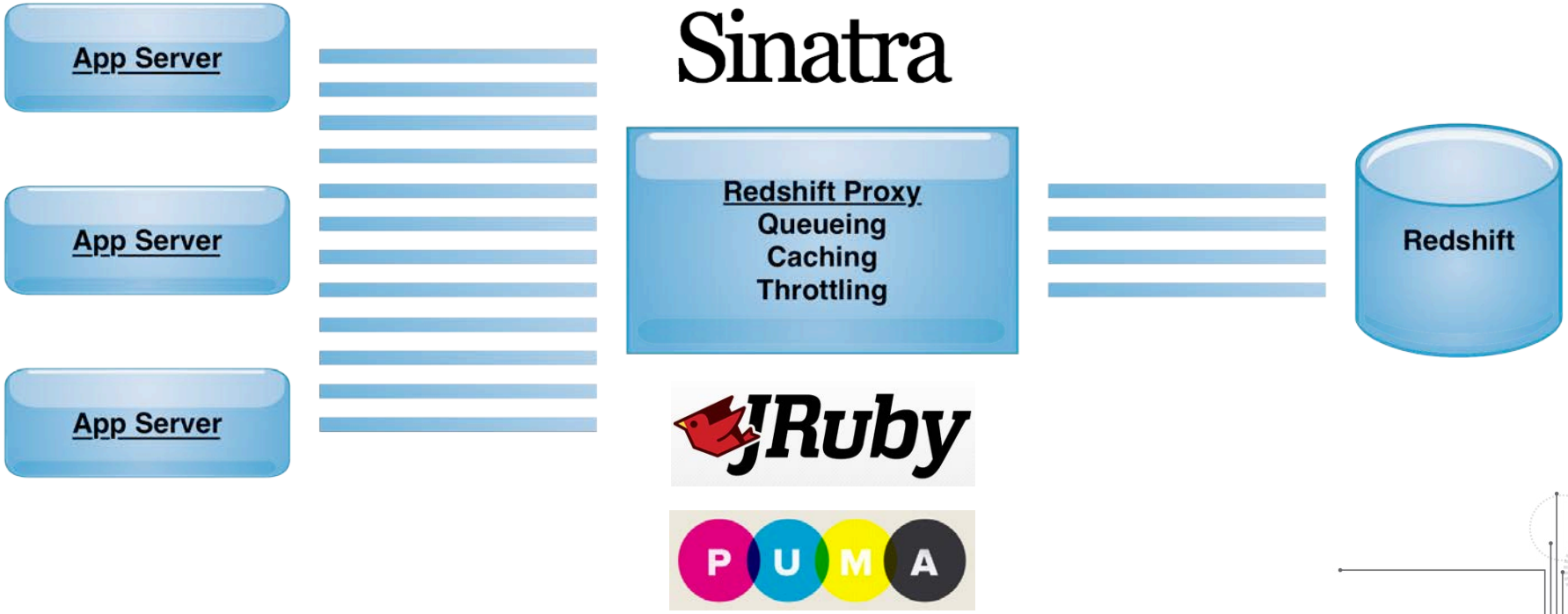


Solution: Larger Datasets, Fewer Queries

- D3 allows binding MBs of data in browser
- Paging, sorting, filtering is done client-side – instantly



Solution: Amazon Redshift Proxy



Amazon Redshift Proxy: Queueing

- Queue requests outside Amazon Redshift
- Minimize (or control) contention
- Abandoned requests don’t ever get to Amazon Redshift
- Future: provide ETAs and other queue statistics

Amazon Redshift Proxy: Caching

- Data only updates once per hour
- Cache all reports (JSON) for duration of hour
- ***Every*** cache hit is a big win
- Just use memcached

Amazon Redshift Proxy: Throttling

- We can rate limit reports on per-customer basis
- Ensures single customer cannot monopolize resources

Amazon Redshift Proxy: Example

```
curl http://localhost:9292/query
       -H 'Accept: application/json'
       -H 'Content-Type: application/json'
       -d '{"sql":"SELECT * FROM events LIMIT
?","params":[2]}'
```

Amazon Redshift Proxy: Example

```
"header": [
    ["site_id", "Int"],
    ["event_type","Int"],
    ["created_at", "Timestamp”],
    ...
],
"rows": [
    [1, 1, "2013-07-05T19:01:32-07:00”, ...],
    [1, 1, "2013-07-05T19:01:37-07:00”, ...]
]
```

Summary

- Amazon Redshift allowed us to rapidly build and ship our next-gen customer-facing reporting portal
- It’s always getting better – improvements and new features nearly every week
- Awesome support and guidance from AWS team

AWS re:Invent

Getting Maximum Performance from Amazon Redshift: Fast Data Loads

Niek Sanders, HasOffers

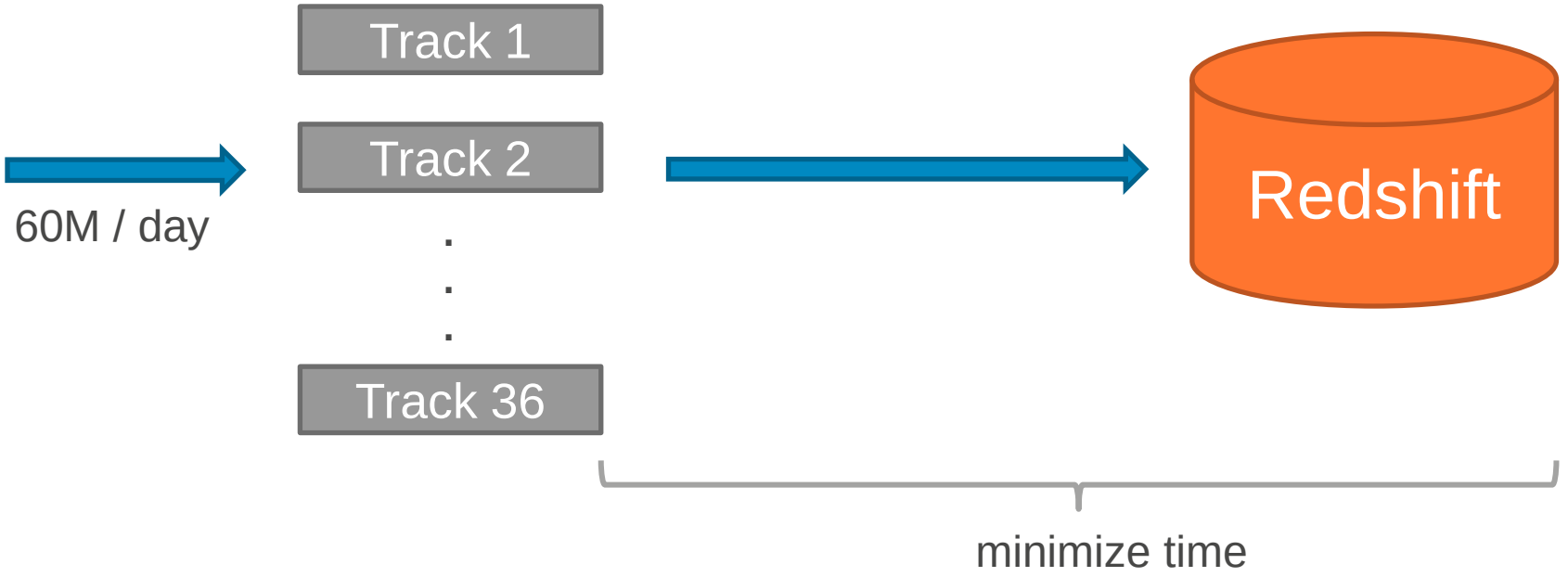
November 13, 2013

amazon web services

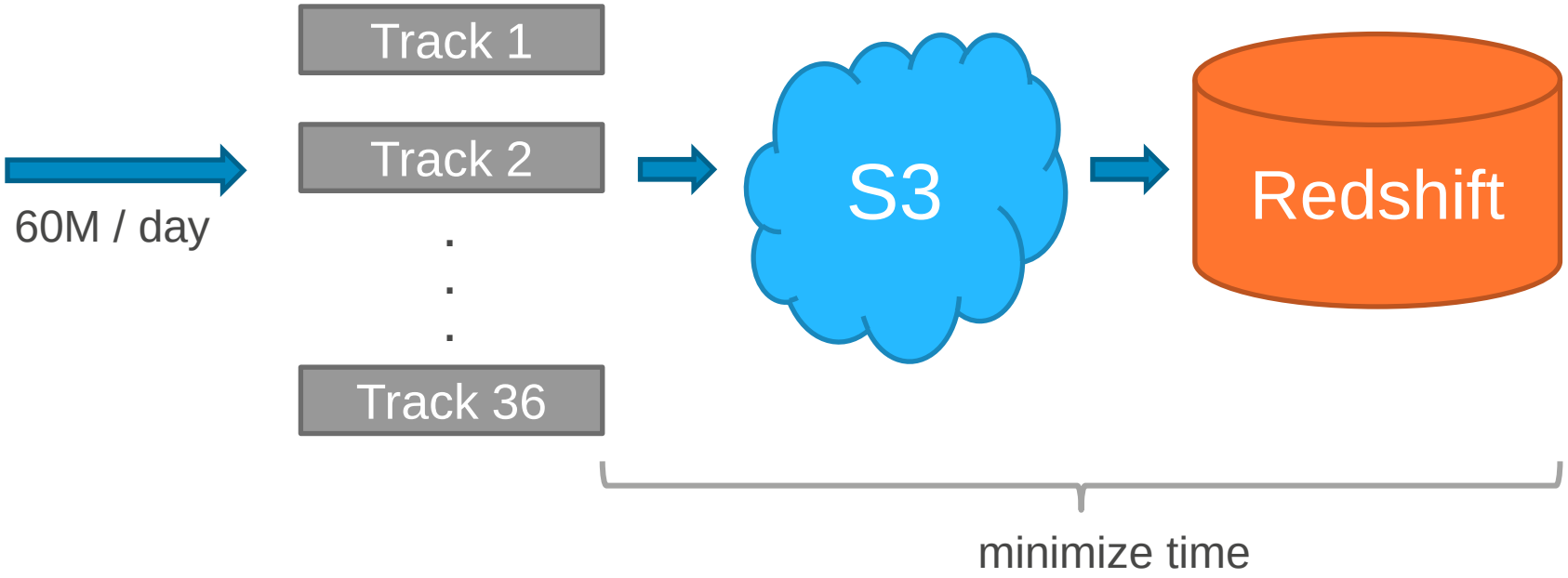
- Attribution for web & mobile marketing
- Tons of data
- Ad-hoc analysis
- Near real-time expectation

Live(ish) Amazon Redshift Data Loading

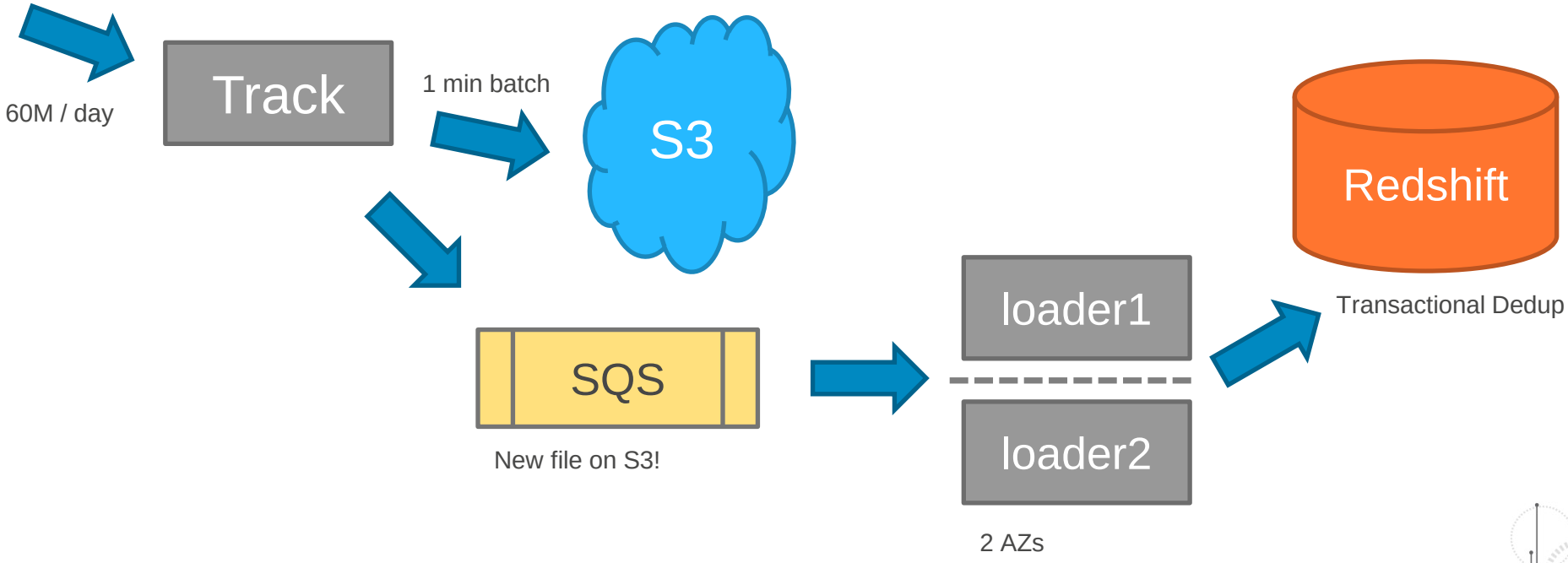
Objective



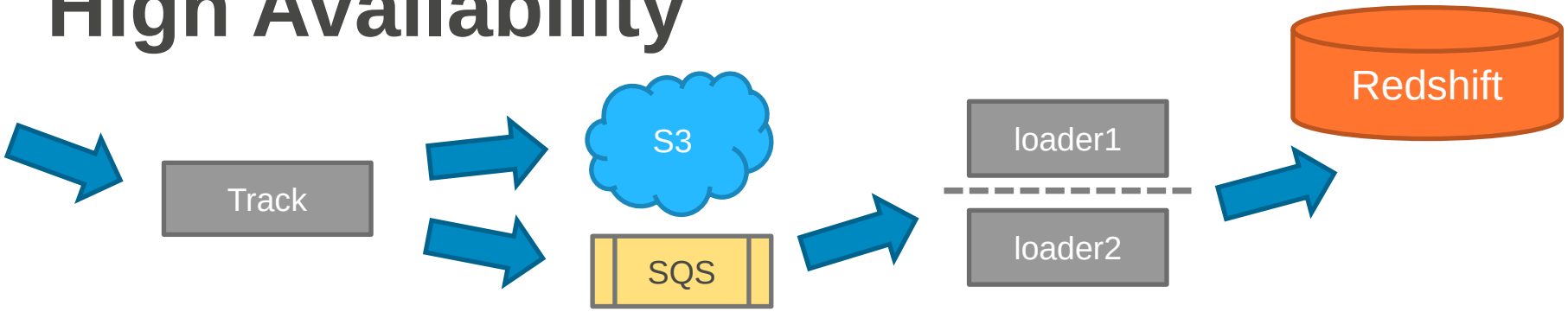
Requires Amazon S3, DML Inserts Slow



Core Idea


60M / day
Track
1 min batch
S3
Redshift
Transactional Dedup
SQS
New file on S3!
loader1
loader2
2 AZs

High Availability



- 2 loaders, separate AZs
- Shared-nothing on loaders
- Amazon SQS re-queue if loader dies
- Idempotent, transactional de-duplicator

Four Big Ideas

1. Micro-batching and Amazon S3
2. Amazon SQS for work queue
3. Bulk Amazon S3 COPY
4. Transactional de-duplicator

Four Big Ideas

1. ~~Micro-batching and Amazon S3~~
2. ~~Amazon SQS for work queue~~
3. Bulk Amazon S3 COPY 
4. Transactional de-duplicator

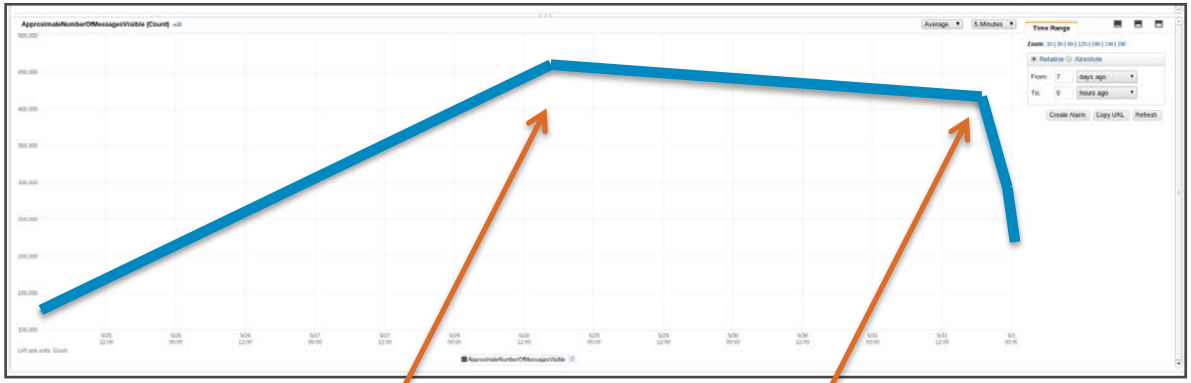
Amazon S3 COPY

```
COPY clicks FROM 's3://foo/2014/01/' ...
```

Single file or path prefix.

Bulk Amazon S3 COPY

Amazon SQS Work Queue

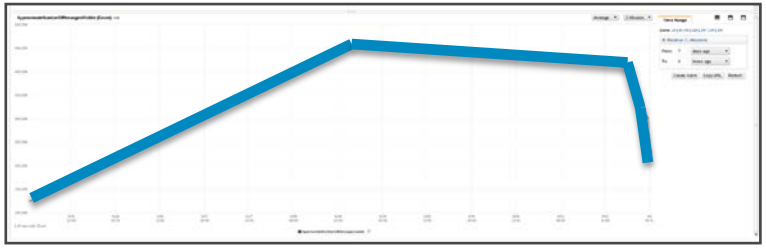


File Loads

Prefix Loads

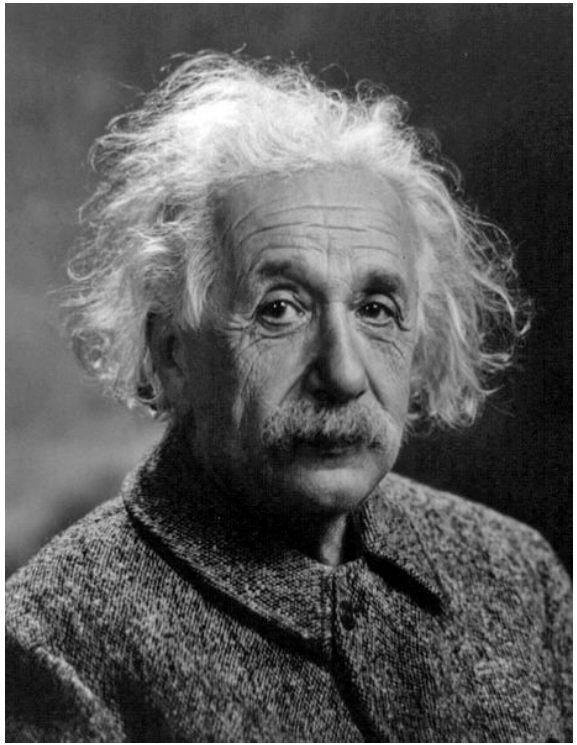
Bulk Amazon S3 COPY

- Bulk loads = 10x speed



- Cluster size, parallelism
  - Roughly equal file size
  - XL nodes → 2 files per node
  - 8XL nodes→ 16 files per node

Bulk Amazon S3 COPY: Common Prefixes



s3://foo/customerA/2013/03/clicks81.gz
s3://foo/customerB/2013/03/clicks93.gz
s3://foo/customerB/2013/03/clicks30.gz
s3://foo/customerC/2013/03/clicks01.gz

s3://foo/common_prefix/

Bulk Amazon S3 COPY: Common Prefixes

New!

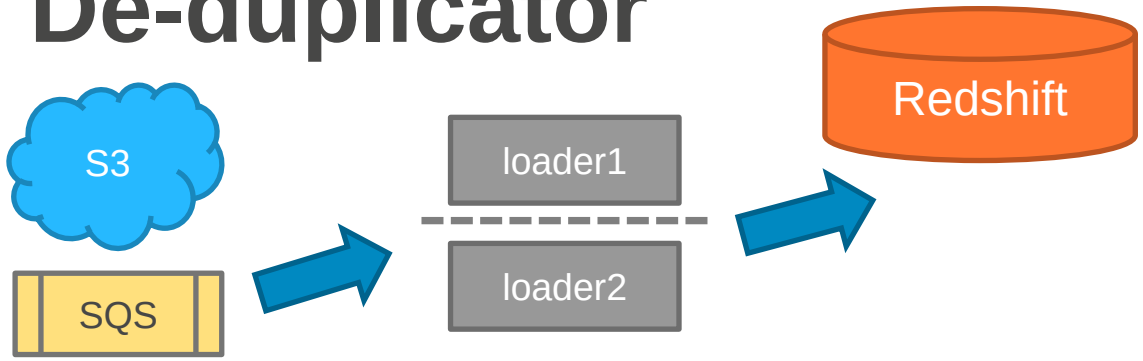
```
COPY clicks
FROM ‘s3://foo/files.manifest’
MANIFEST…
```

List of files

Four Big Ideas

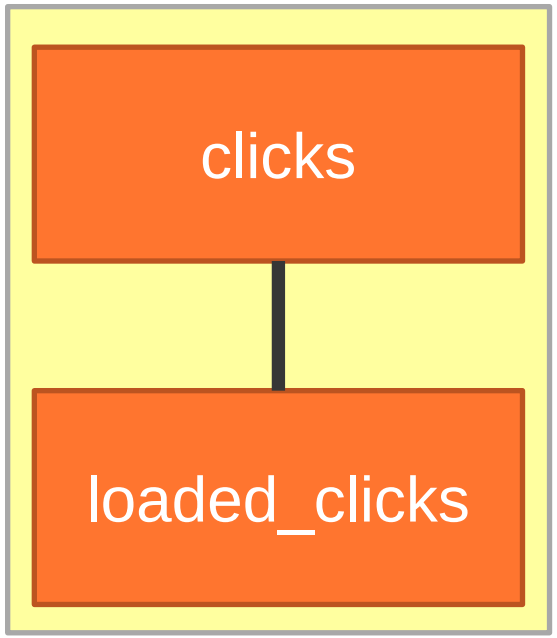
1. ~~Micro-batching and Amazon S3~~
2. ~~Amazon SQS for work queue~~
3. ~~Bulk Amazon S3 COPY~~
4. Transactional de-duplicator

Transactional De-duplicator



- Prevent duplicate loads
- Redshift transactions
- Idempotent load flow

Transactional Deduplicator

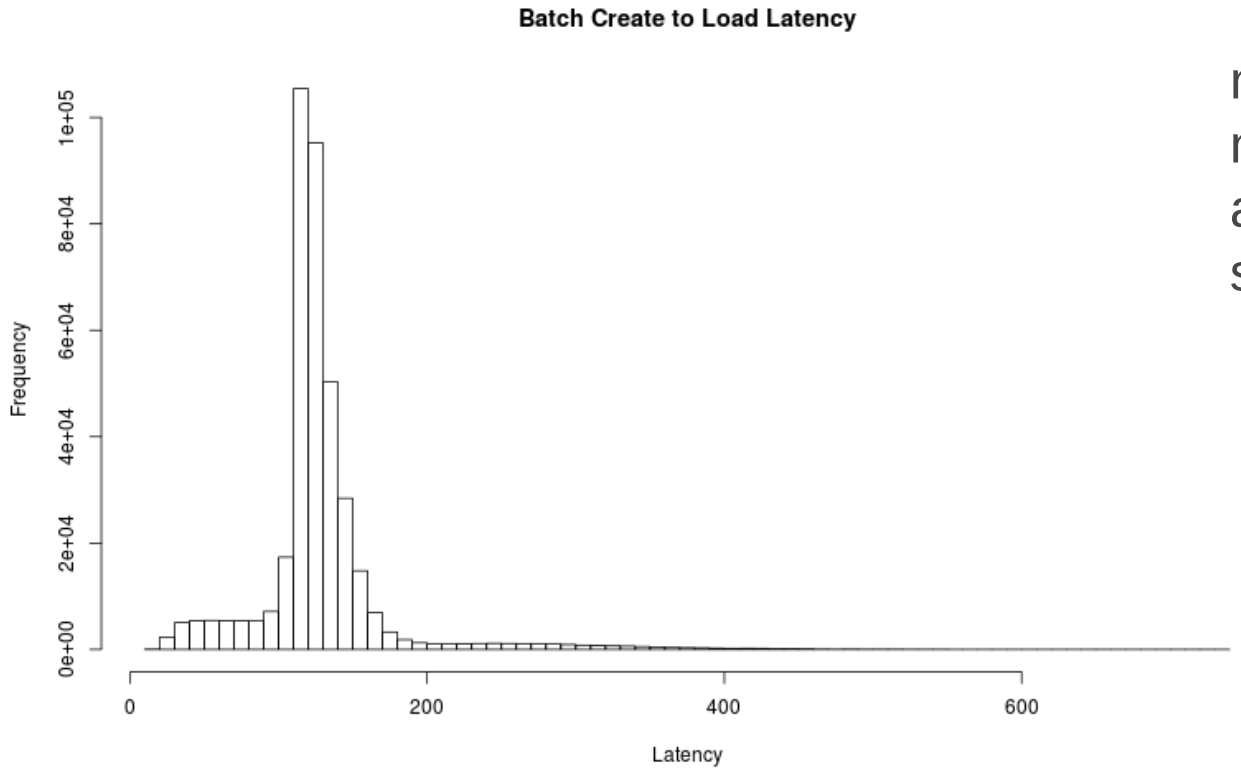


- Batch in loaded_clicks? Stop
- Begin txn
  - COPY
  - Batch in loaded_clicks? Rollback
  - Insert batch to loaded_clicks
- Commit

Four Big Ideas

1. Micro-batching and Amazon S3
2. Amazon SQS for work queue
3. Bulk Amazon S3 COPY
4. Transactional de-duplicator

Results



min = 15
max = 733
avg = 129 sec
stddev = 47

has offers™

Thank you!

niek@hasoffers.com

AWS re:Invent

Getting Maximum Performance from Amazon Redshift: Complex Queries

Timon Karnezos, Aggregate Knowledge

November 13, 2013

amazon web services

Study: Facebook Leads to 24% Sales Boost

Social network brings new
cost, per Aggregate Knowl

channel to see a user before they convert. That last-click attribution, for better or worse, is what helped Google establish a multibillion dollar advertising business. “Where Facebook gets last-touch credit is where it’s the only place on the planet that reached that user,” Jakubowski said. Facebook tends to factor in a couple days or weeks before a user converts, he noted. Smallwood emphasized that marketers shouldn’t look at Facebook in silo but consider an entire media mix and attribute success through multi-touch attribution.

Meet the new boss

Multi-touch Attribution

Examples and real world applications [edit]

Data shows that a large percentage of users using a certain eCommerce platform found it by searching for “Thai food” on Google Food” page and then logged off without placing an order. Looking at each of these events as separate data points does not repre However, viewing these data points as a representation of overall user behavior enables one to interpolate how and why users ac

Behavioral analytics looks at all site traffic and page views as a timeline of connected events that did not lead to orders. Since r disconnect between what they are searching for on Google and what the “Asian Food” page displays. Knowing this, a quick look prominently and thus people do not think it is actually offered, even though it is.

Behavioral analytics is becoming increasingly popular in commercial environments. Amazon.com is a leader in using behavioral

Same as the old boss

Behavioral Analytics

Examples [edit source]

Market basket analysis might tell a retailer that customers often purchase shampoo and conditioner together, so putting both items on promotion at the same time would not create a significant increase in profit, while a promotion involving just one of the items would likely drive sales of the other.

Market basket analysis may provide the retailer with information to understand the purchase behavior of a buyer. This information will enable the retailer to understand the buyer's needs and rewrite the store's layout accordingly, develop cross-promotional programs, or even capture new buyers (much like the cross-selling concept). An apocryphal early illustrative example for this was when one super market chain discovered in its analysis that customers that bought diapers often bought beer as well, have

Same as the old boss

Market Basket Analysis

We know how to do this *in SQL*!*

** SQL:2003*

Here it is

```
SELECT record_date, user_id, action,
       site, revenue,
       SUM(1) OVER
              (PARTITION BY user_id
               ORDER BY record_date ASC)
       AS position
FROM user_activities;
```

So why is MTA *hard*?

“Web Scale”

Queries	Data
➢ 30 queries	➢ ~10^9 daily impressions
➢ 1700 lines of SQL	➢ ~10^7 daily conversions
➢ 20+ logical phases	➢ ~10^4 daily sites
➢ ***GBs of output***	➢ ***x 90 days***

per *report*

So, how do we deliver *complex reports* over *“web scale” data*?

(Pssst. The answer’s Amazon Redshift. Thanks, AWS.)

Write (good) queries

Organize the data

Optimize for the humans

Write (good) queries

Remember: SQL is *code*

Software engineering rigor applies to SQL

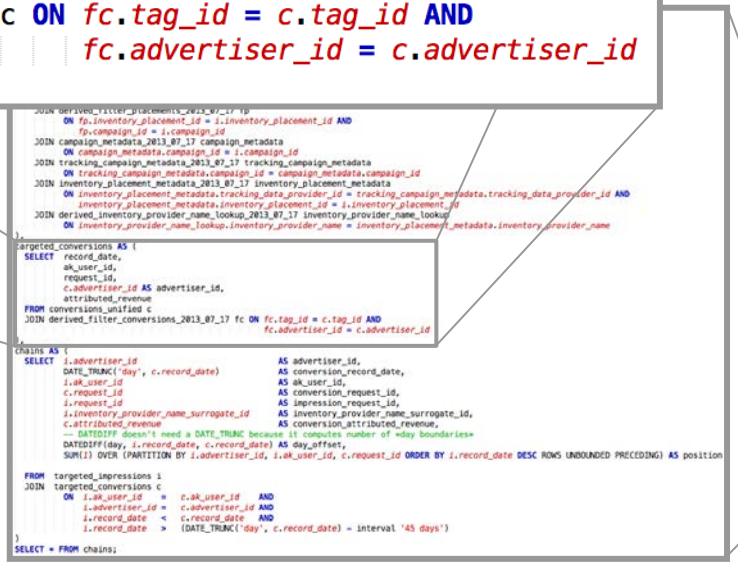
Factored

Concise

Tested

Common Table Expression

```
targeted_conversions AS (
  SELECT  record_date,
          ak_user_id,
          request_id,
          c.advertiser_id AS advertiser_id,
          attributed_revenue
  FROM conversions_unified c
  JOIN derived_filter_conversions_2013_07_17 fc ON fc.tag_id = c.tag_id AND
                                                   fc.advertiser_id = c.advertiser_id
),
```



Factored

Concise

Tested

Window functions

```
-- Position in timeline
SUM(1) OVER (PARTITION BY user_id
             ORDER BY record_date DESC ROWS UNBOUNDED PRECEDING)

-- Event count in timeline
SUM(1) OVER (PARTITION BY user_id
             ORDER BY record_date DESC BETWEEN UNBOUNDED PRECEDING AND
                                      UNBOUNDED FOLLOWING)
-- Transition matrix of sites
LAG(site_name) OVER (PARTITION BY user_id
                     ORDER BY record_date DESC)

-- Unique sites in timeline, up to now
COUNT(DISTINCT site_name) OVER (PARTITION BY user_id
                                ORDER BY record_date DESC
                                ROWS UNBOUNDED PRECEDING)
```

Window functions

Scalable, combinable

Compact but expressive

Simple to reason about

Organize the data

Leverage Amazon Redshift's MPP roots

Fast, columnar scans, IO

Fast sort and load

Effective when work is distributable

Leverage Amazon Redshift's MPP roots

Sort into multiple representations

Materialize shared views

Hash-partition by user_id

Optimize for the humans

Operations should not be the bottleneck

Develop without fear

Trade time for money

Scale with impunity

Operations should not be the bottleneck

Fast Amazon S3 = scratch space for cheap

Linear query scaling = GTM quicker

Dashboard Ops = dev/QA envs, marts, clusters with just a click

But, be frugal

Quantify and control costs

Test across different hardware, clusters

Shut down clusters often

Buy productivity, not bragging rights

Thank you!

References

http://bit.ly/rs_ak

http://www.adweek.com/news/technology/study-facebook-leads-24-sales-boost-146716

http://en.wikipedia.org/wiki/Behavioral_analytics

http://en.wikipedia.org/wiki/Market_basket_analysis

Amazon Redshift Customers at re:Invent

DAT306 - How Amazon.com is Leveraging Amazon Redshift

Thursday 11/14: 3pm in Murano 3303

DAT205 - Amazon Redshift in Action: Enterprise, Big Data, SaaS Use Cases

Friday 11/15: 9am in Lido 3006

AWS re:Invent

Please give us your feedback on this presentation

DAT305

As a thank you, we will select prize winners daily for completed surveys!

amazon web services