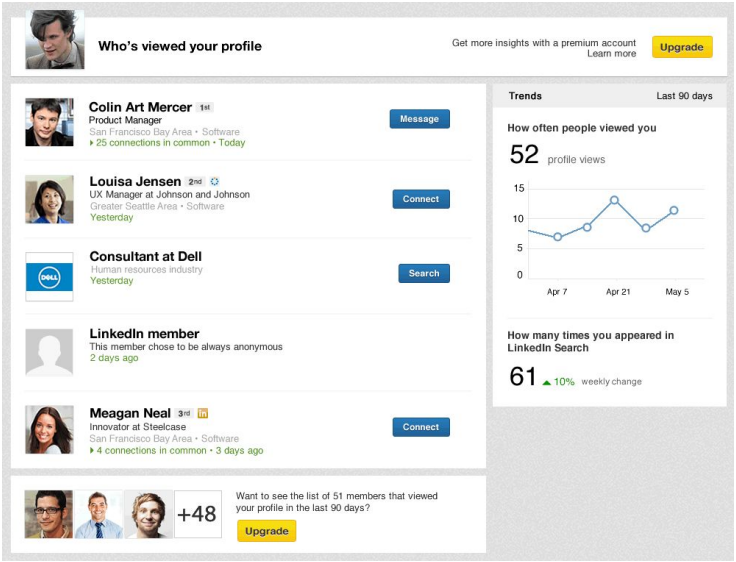
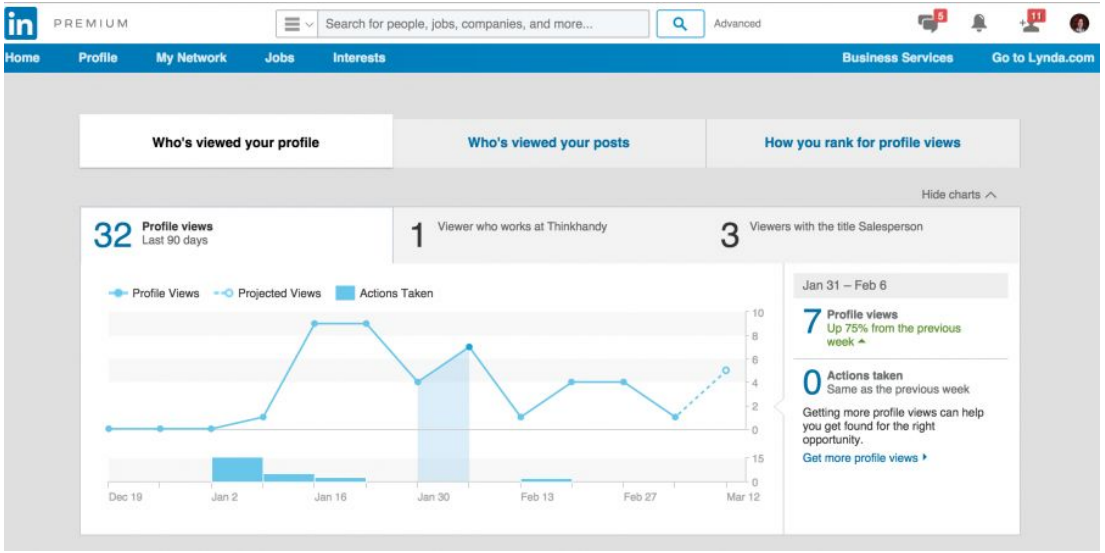
Pinot: Realtime OLAP for 530 Million Users

Presented By: Sushant Raikar

Linkedin


Who's viewed your profile
Get more insights with a premium account
Learn more
Upgrade
Colin Art Mercer 1st
Product Manager
San Francisco Bay Area • Software
25 connections in common • Today
Message
Louisa Jensen 2nd
UX Manager at Johnson and Johnson
Greater Seattle Area • Software
Yesterday
Connect
Consultant at Dell
Human resources industry
Yesterday
Search
LinkedIn member
This member chose to be always anonymous
2 days ago
Meagan Neal 3rd
Innovator at Steelcase
San Francisco Bay Area • Software
4 connections in common • 3 days ago
Connect
+48
Want to see the list of 51 members that viewed your profile in the last 90 days?
Upgrade
Trends
Last 90 days
How often people viewed you
52 profile views
15
10
5
0
Apr 7
Apr 21
May 5
How many times you appeared in LinkedIn Search
61 ▲ 10% weekly change


PREMIUM
Search for people, jobs, companies, and more...
Advanced
Home
Profile
My Network
Jobs
Interests
Business Services
Go to Lynda.com
Who's viewed your profile
Who's viewed your posts
How you rank for profile views
Hide charts
32 Profile views
Last 90 days
1 Viewer who works at Thinkhandy
3 Viewers with the title Salesperson
Profile Views
Projected Views
Actions Taken
10
8
6
4
2
0
15
0
Dec 19
Jan 2
Jan 16
Jan 30
Feb 13
Feb 27
Mar 12
Jan 31 – Feb 6
7 Profile views
Up 75% from the previous week
0 Actions taken
Same as the previous week
Getting more profile views can help you get found for the right opportunity.
Get more profile views

Challenge

100 QPS -> 10000 QPS

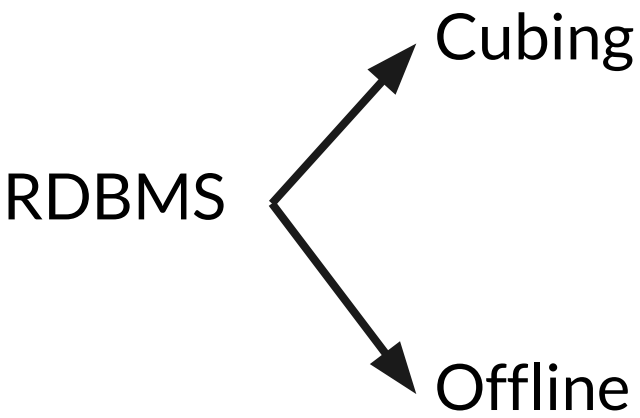
Pinot

- Single system for all OLAP needs
- Ability to serve upto 50000QPS
- Flexible
- Near-Realtime data

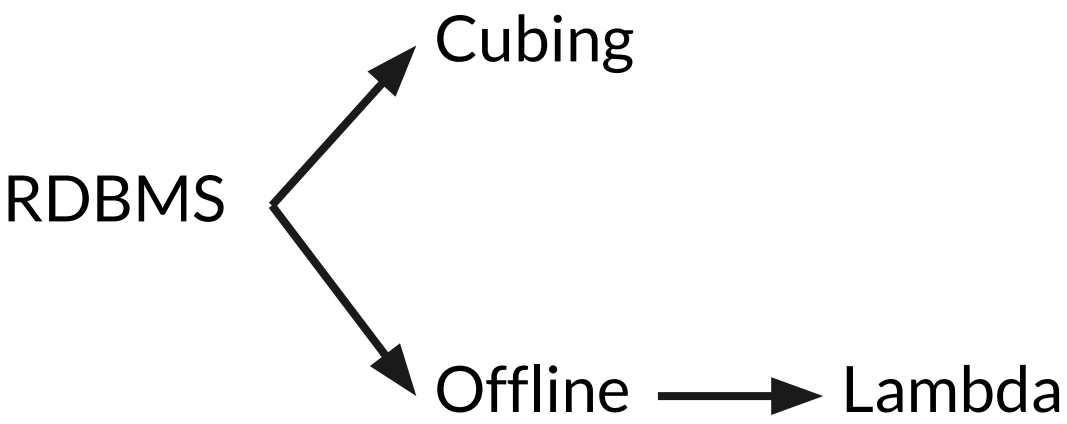
History

RDBMS

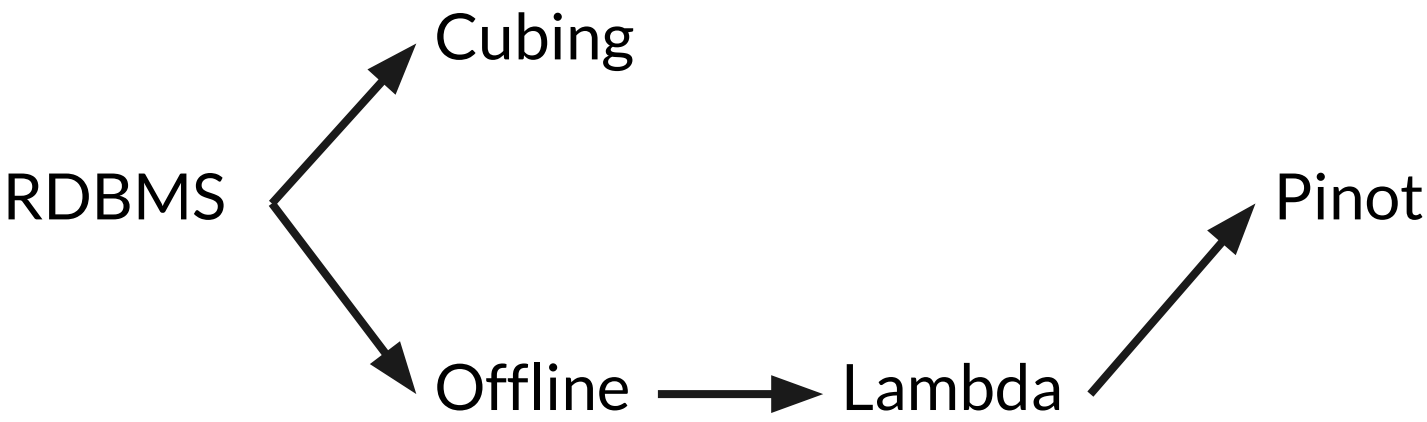
History



History



History



Data in Pinot

Doc Id	Model	Year	Price	tags
1	Toyota	1997	7500	moon-roof, reliable
2	Toyota	2001	7500	mp3, compact
3	Mazda	2005	1500	Manual, compact
4	Honda	2010	8000	leather, cool
5	Honda	2011	9000	automatic
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
10000	bmw	.	25000	moon-roof, reliable

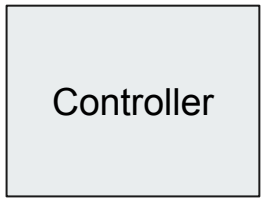
RAW DATA

INDEX GENERATION

Doc Id	Model	Year	Price	tags
1	Toyota	1997	7500	moon-roof, reliable
2	Toyota	2001	7500	mp3, compact
3	Mazda	2005	1500	Manual, compact
4	Honda	2010	8000	leather, cool
5	Honda	2011	9000	automatic
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
10000	bmw	.	25000	moon-roof, reliable

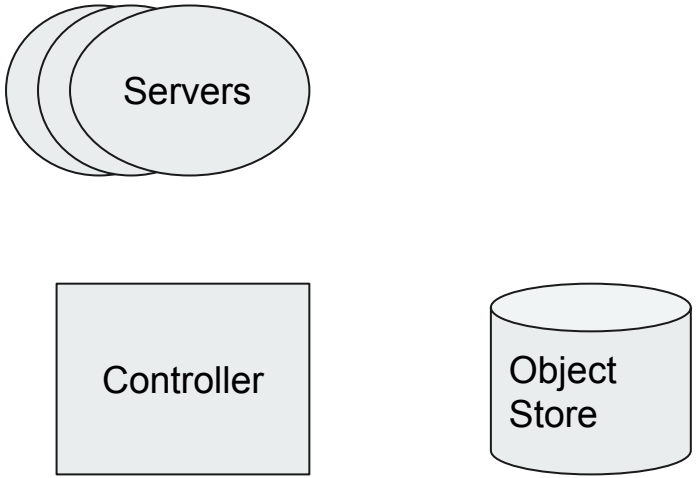
COLUMNAR DATA

Architecture

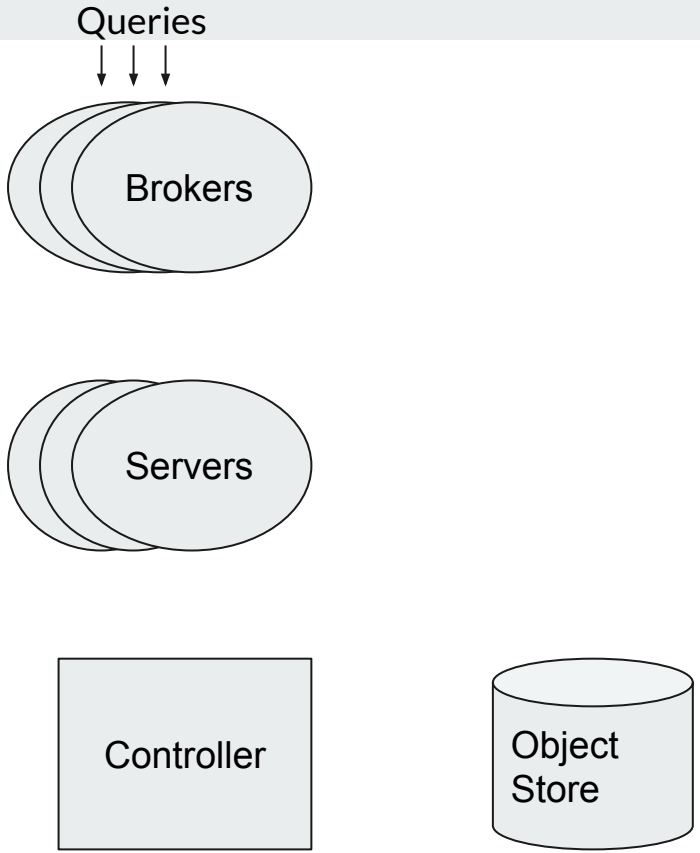


Architecture

Architecture

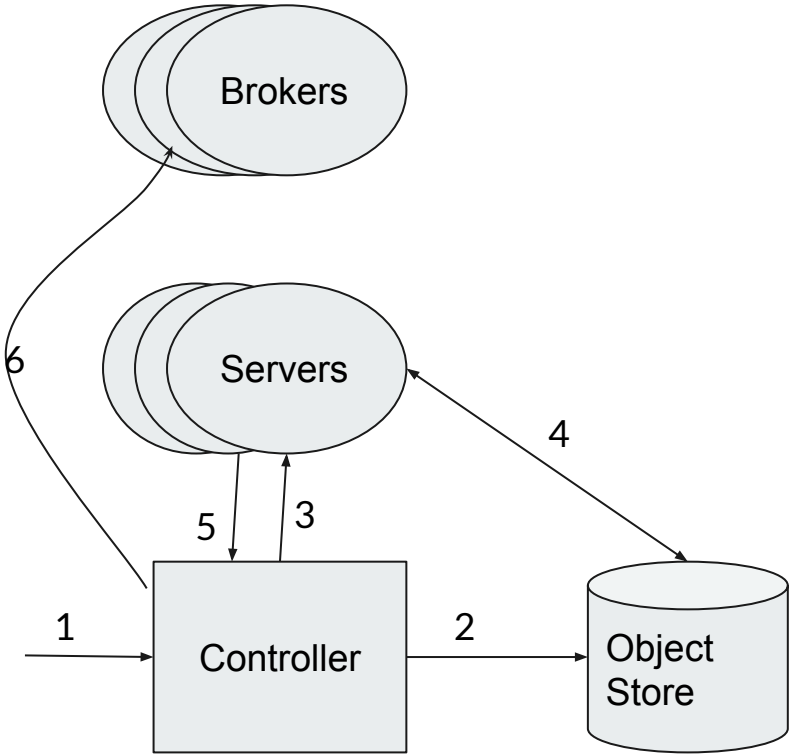


Architecture



Uploading Data

1. HTTP POST Data Segment to Controller
2. Controller uploads it to persistent object store
3. Controller asks servers to download segment
4. Server downloads data from Object Store
5. Tells Controller that it finished downloading
6. Brokers update routing Tables

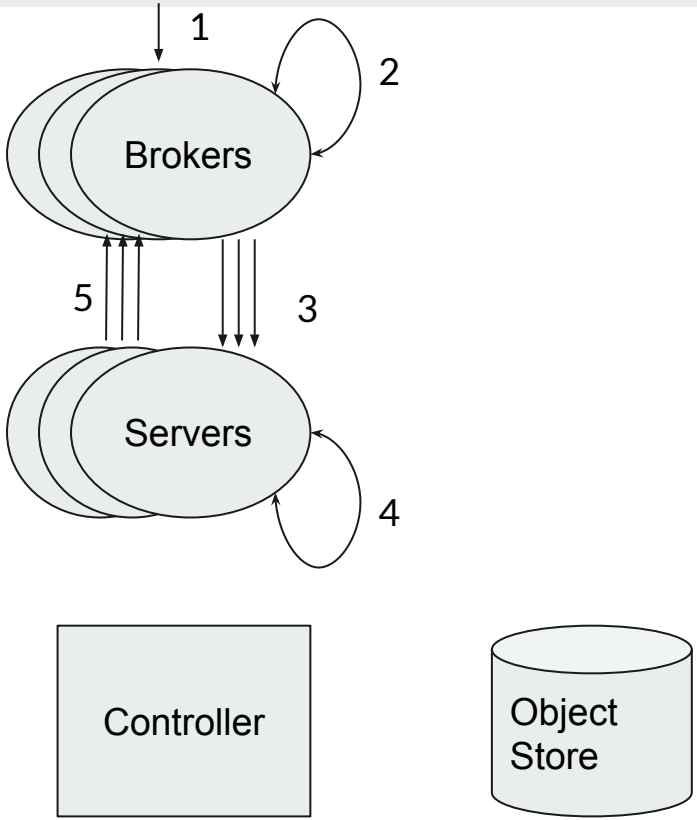


Running Queries

1. Queries arrive at broker get analyzed
2. Routing table is picked
3. Multiple servers are contacted
4. Servers generate logical and physical query plans and execute the query.
5. Broker merges data

Routing is important, especially for large clusters

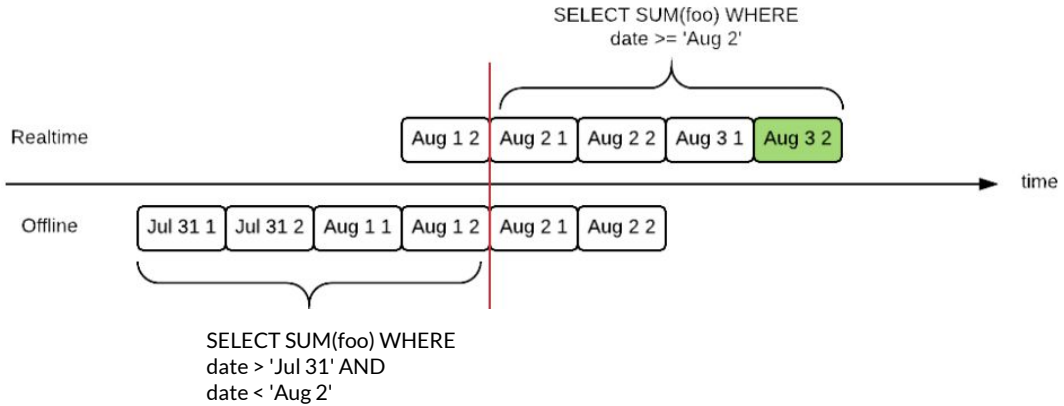
- Multiple routing tables for same table
- Randomly select one routing table for a query
- Load balancing



Hybrid Queries

To handle near real time queries

- "Online" Server
  - Consumes from Kafka
  - Stops based on time or number
  - Contacts controller for consensus
  - Forms indexes similar to offline
- Broker splits query
  - Based on Time
  - Into offline and online
  - Merges result

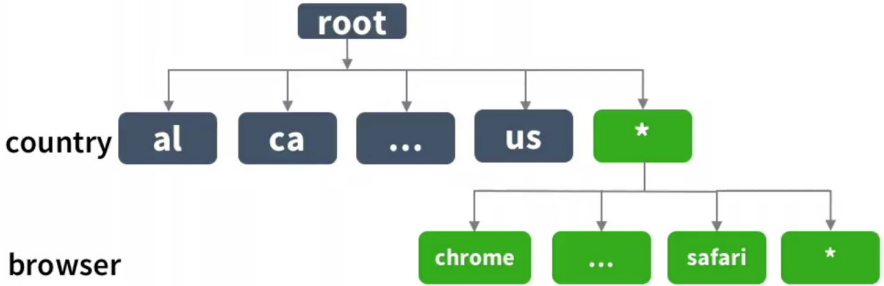



```
SELECT SUM(foo)
FROM T
 WHERE date > 'Jul 31'
```

Optimization: Star Tree Index

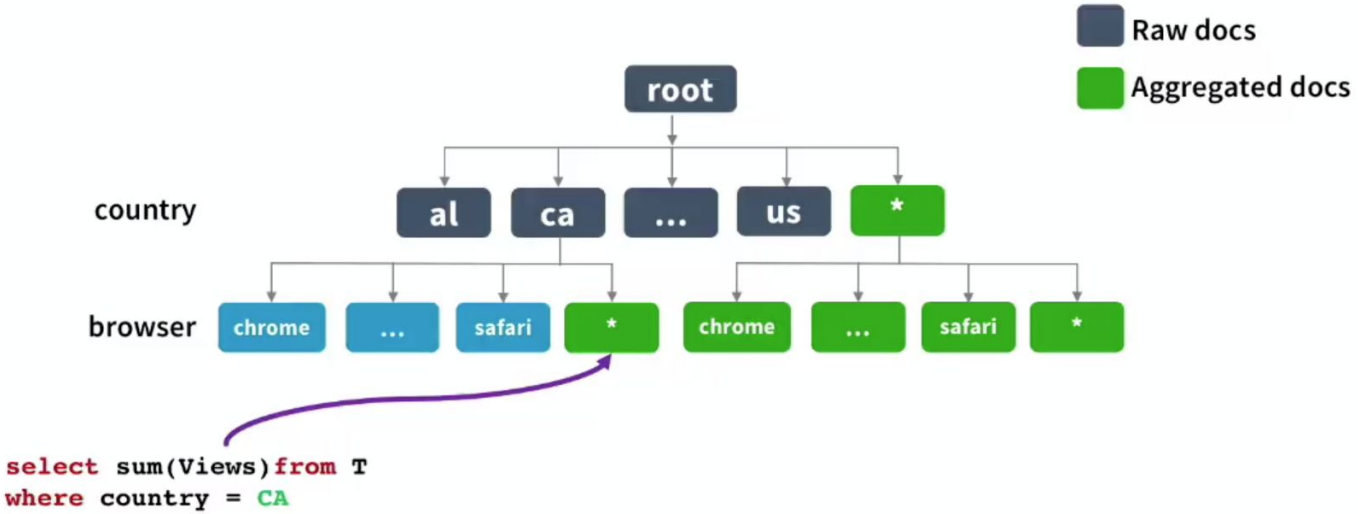
docId	country	browser	… other dimensions	View
0	al	ie		10
1	ca	safari		10
2	…	…		…
…	us	chrome		10
…	us	chrome		10
…	us	ie		10
N	us	safari		10

Raw records



1. Multidimensional sort
2. Split on the column and create a node for each value (+raw doc range)
3. Create star node (contains aggregated docs after removing the split column)
4. Apply 1,2,3 for each node recursively until the number of records in node < ***SplitThreshold***

Optimization: Star Tree Query

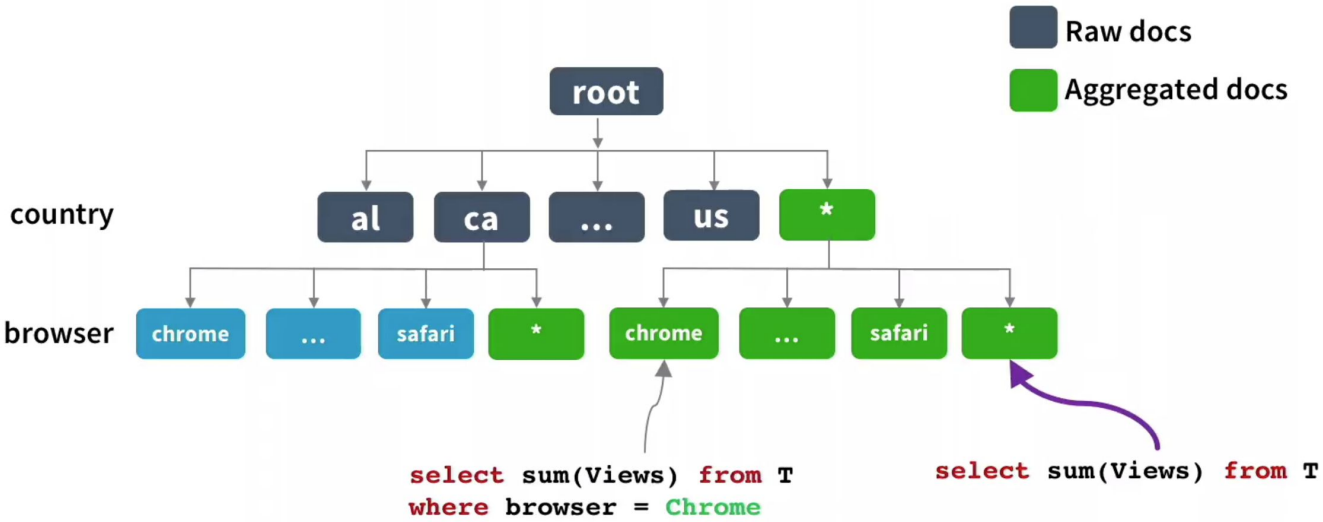



```
select sum(X)
from T
where d1=v1 and d2=v2 and …
```

➡

Any query pattern will scan
less than *SplitThreshold* records

Optimization: Star Tree Query




```
select sum(X)
from T
where d1=v1 and d2=v2 and …
```

➡

Any query pattern will scan less than *SplitThreshold* records

Results

- Single infrastructure for all OLAP needs
- Support for user facing analytical queries
  - 50000 Queries per Second
- Query Flexibility
  - High throughput, low complexity
  - Low throughput, complex queries
- Pluggable Indexing
  - Star Tree Index

Discussion

- Limitations of Pinot?
- How does Pinot differ from in-memory OLAP like Hyper or Hekaton?
- Pinot is becoming Apache, with support from companies like Uber, Slack. What does Linkedin gain or lose from this move?
- Is Pinot basically druid with better optimizations?
- Comments about the paper