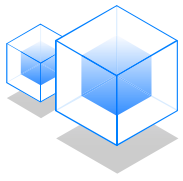


**BERLIN
BUZZWORDS
2019** JUNE 16-18

Accelerate big data analytics with Apache Kylin

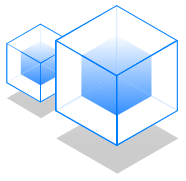
Shaofeng Shi, 史少锋

Apache Kylin committer & PMC



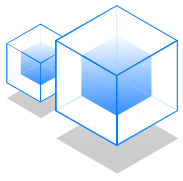
About me

- Shaofeng Shi , 史少锋
shaofengshi@apache.org,
shaofeng.shi@kyligence.io
- Apache Kylin committer & PMC, joined Kylin project since 2014 in eBay;
- Now chief architect at Kyligence Inc.



Agenda

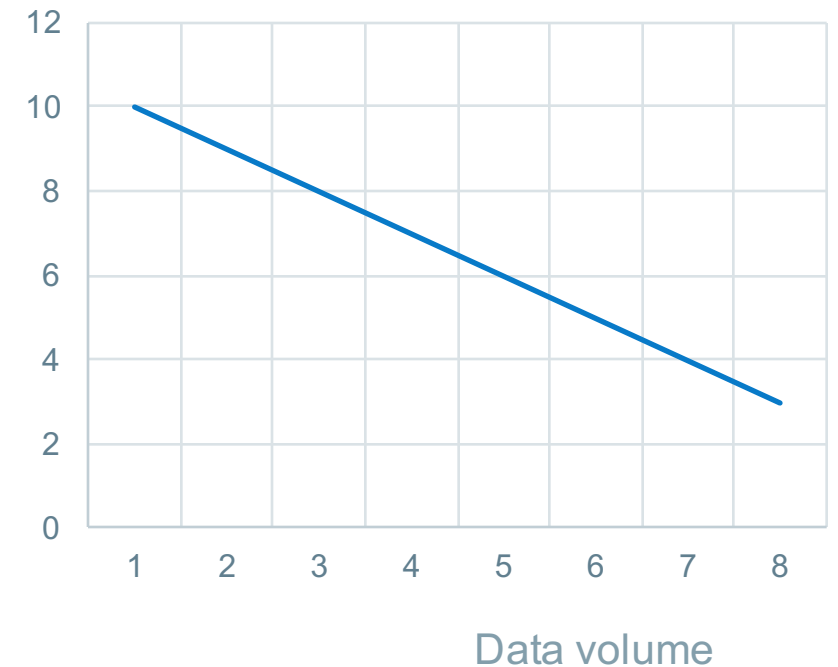
- Apache Kylin background
- Why OLAP cube is needed for big data
- How Kylin build/persist/query cube on Hadoop
- Performance benchmark
- Use cases

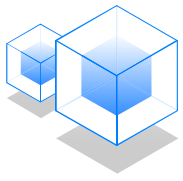


Background

- Huge amount of data be collected today.
 - Transactions, server/app logs, mobile/IoT, etc;
 - Hadoop is the de facto standard for big data.
- Hadoop was not designed for low-latency SQL query.
 - MapReduce, Hive, Pig, Spark... are for batch processing;
- Query performance decreases dramatically as data volume increases. User experience is bad.
- Challenge: how to keep high performance data analytics as data grows?

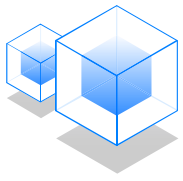
Query performance





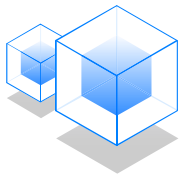
MPP and SQL on Hadoop

- Massive Parallel Processing (MPP) and SQL on Hadoop technologies tries to solve the problem.
 - Amazon Redshift, Greenplum, Presto, Impala, Spark SQL, etc;
- MPP accelerate queries in the following ways:
 - Distribute data into multiple nodes, processing in parallel;
 - Optimize I/O with column-oriented storage, compression encoding, etc;
 - In-memory processing;

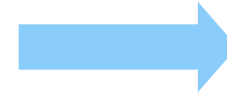
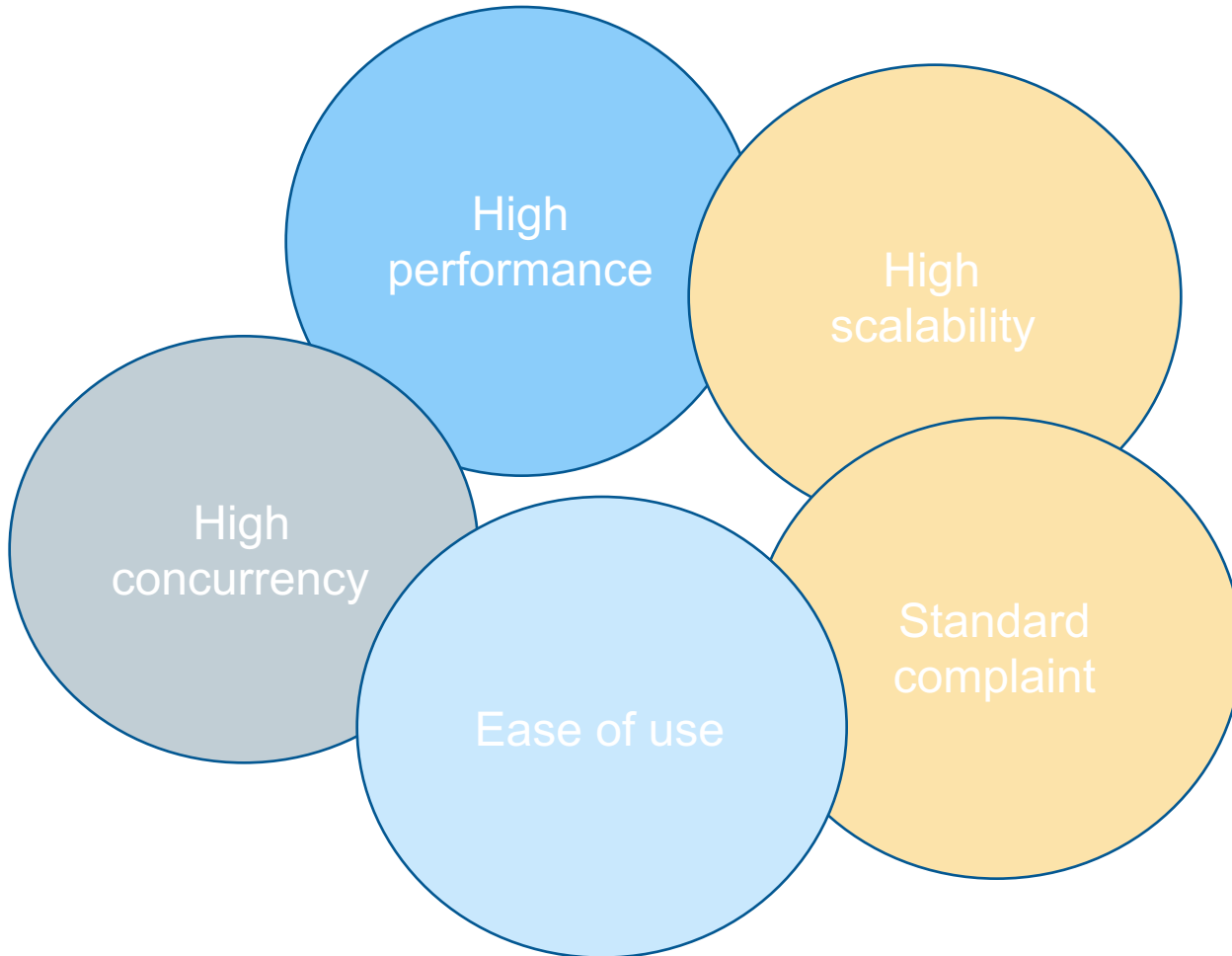


MPP's limitation

- Performance
 - Handling massive data costs much resource; When resource is insufficient, performance is downgraded. Latency from tens of seconds to tens of minutes.
- Concurrency
 - CPU/memory/network intensive, hard to serve multiple concurrent users;
- Scalability
 - Data need be redistributed when scale out, taking hours;
 - Master node becomes a bottleneck as cluster size increase. Cluster size is limited to 1-2 hundreds;

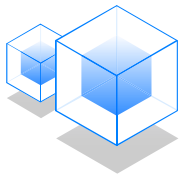


A real OLAP solution for big data



+



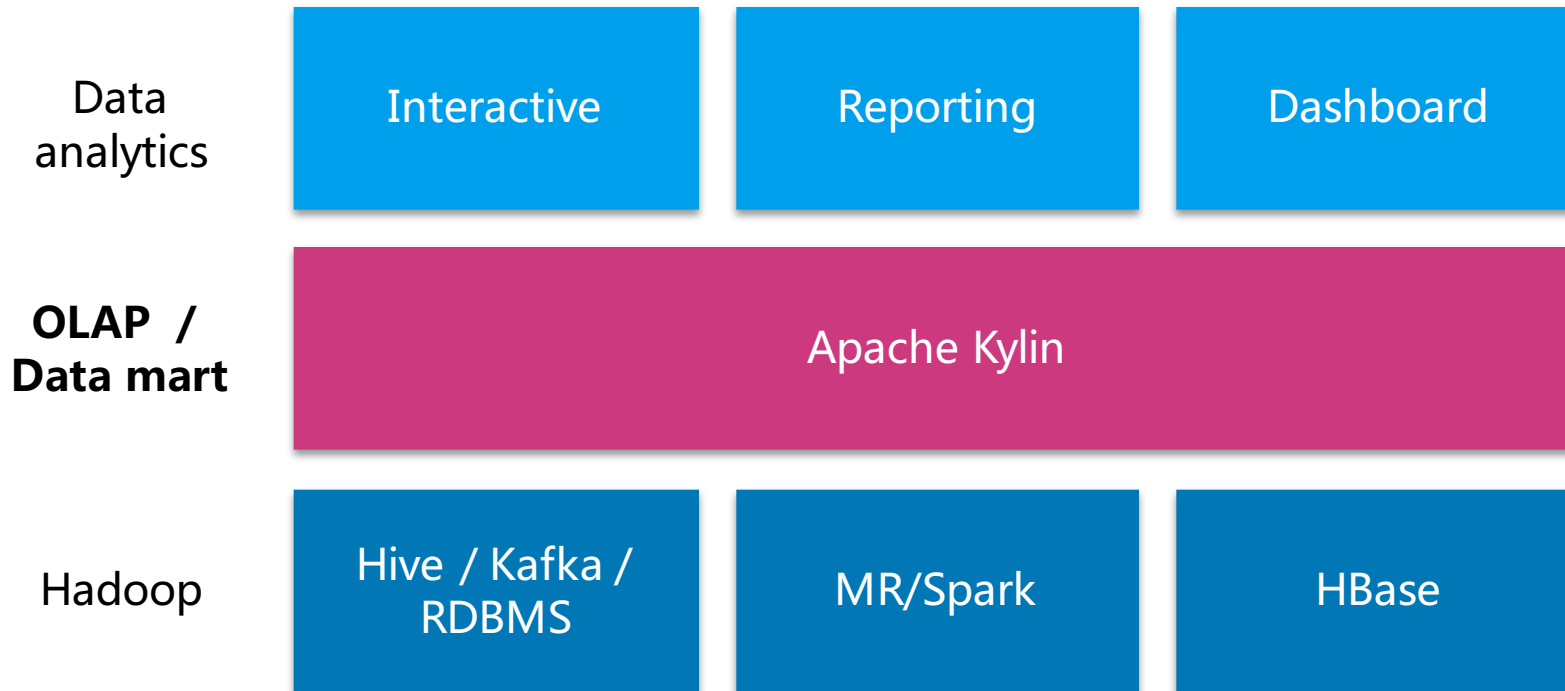


What is Apache Kylin

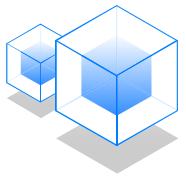
Apache Kylin: Extreme OLAP Engine for Big Data



<https://kylin.apache.org>

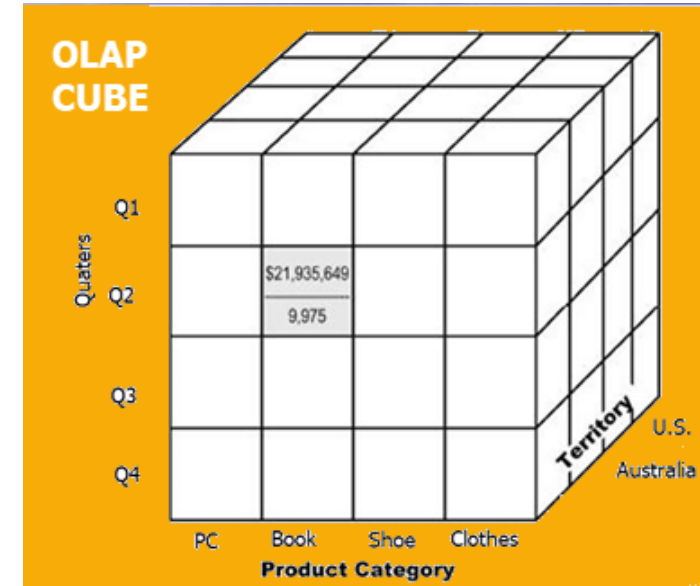


- Build OLAP cube on Hadoop
- Support TB to PB data
- Sub-second query latency
- ANSI-SQL
- JDBC / ODBC / REST API
- BI integration
- Web GUI
- LDAP/SSO

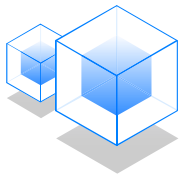


OLAP cube

- OLAP cube is a data structure optimized for very quick data analysis.
- OLAP cube has been adopted by traditional Data Warehouses for decades.
- For analyst, the cube concept is easier to be accepted than others.
- Why not use cube for big data OLAP?

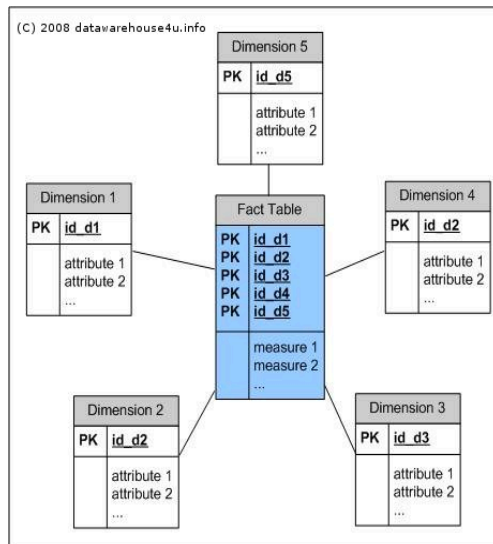


Picture is from
<https://www.guru99.com/online-analytical-processing.html#4>

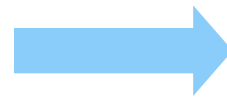


Benefits from cube for big data

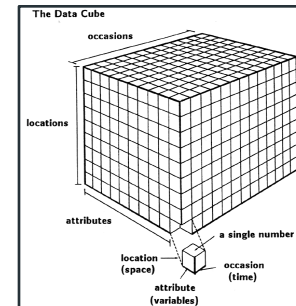
- ✓ **Performance:** querying cube can be 1000x faster than querying raw data;
- ✓ **Ease of use:** cube is topic oriented, analysts can self-serve;
- ✓ **Cost efficiency:** build once, use many times. Much computing resources can be saved.
- ✓ **Feasibility:** with Hadoop, building cube for a large dataset becomes easy.



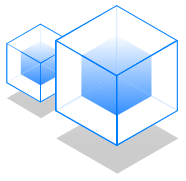
Source data



Classification,
aggregation, and
sorting



OLAP Cube



Apache Kylin is built with mainstream technologies

— Offline data flow
— Online data flow

 Data Analyst, BI Tools, Web App...

4. Send SQL query

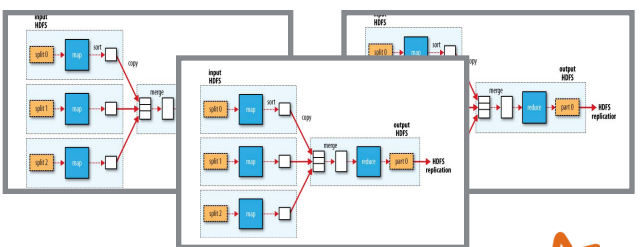


 **Apache Kylin**

5. Parse, optimize & rewrite query

6. Scan & filter cube

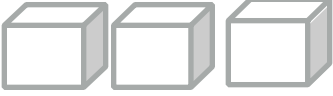

 
 

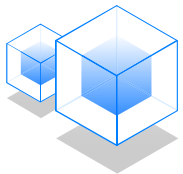
1. Extract data to Hadoop

2. Build the cube

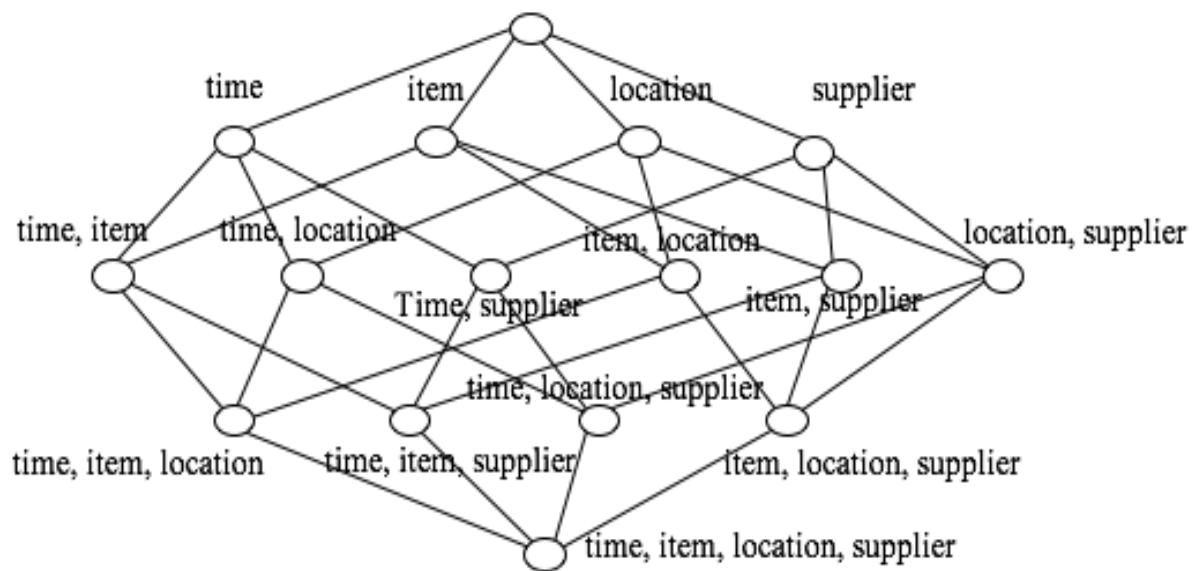
3. Load



Kylin can build all cuboids your want

- One cube has 2^N cuboids; (N = dimension number)
- Cuboid = Materialized view;
- All cuboids are consistent (updated atomically);
- User can define partial cube with rules.



0-D(apex) cuboid

1-D cuboids

2-D cuboids

3-D cuboids

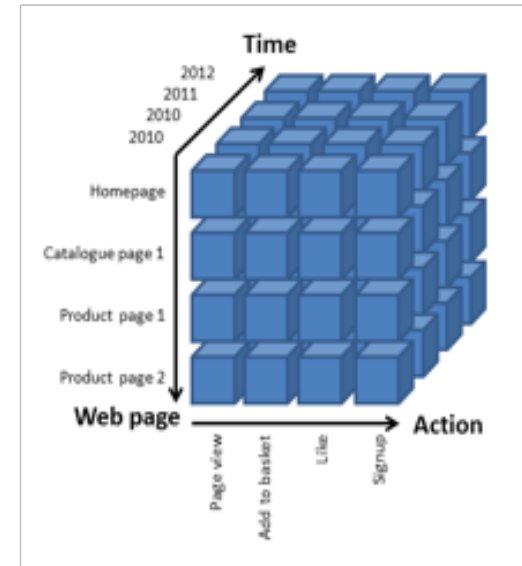
4-D(base) cuboid

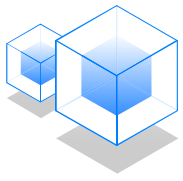
• **Base vs. aggregate cells; ancestor vs. descendant cells; parent vs. child cells**

1. (9/15, milk, Urbana, Dairy_land) - <time, item, location, supplier>
2. (9/15, milk, Urbana, *) - <time, item, location>
3. (*, milk, Urbana, *) - <item, location>
4. (*, milk, Chicago, *) - <item, location>
5. (*, milk, *, *) - <item>

OLAP Cube

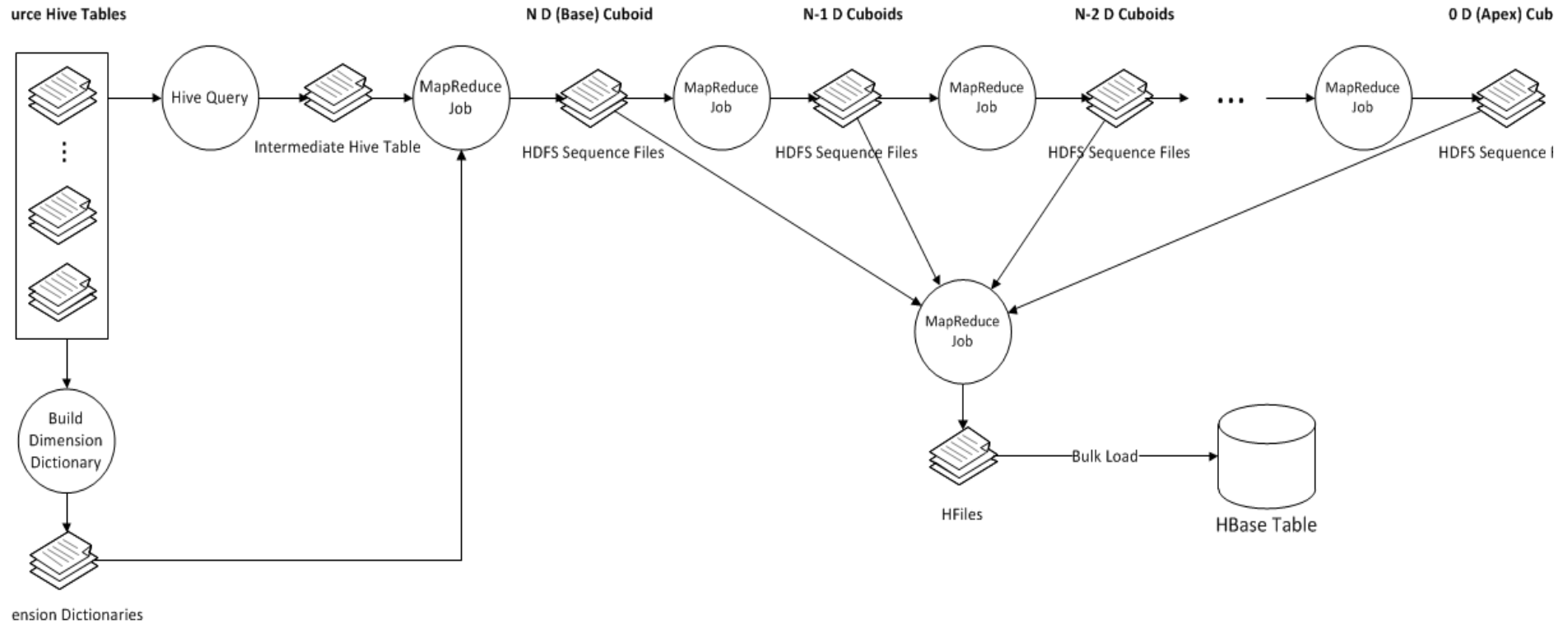
- Cuboid = one combination of dimensions
- Cube = all combination of dimensions (all cuboids)

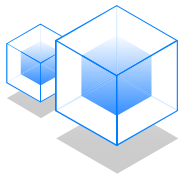




How Kylin build cube on Hadoop

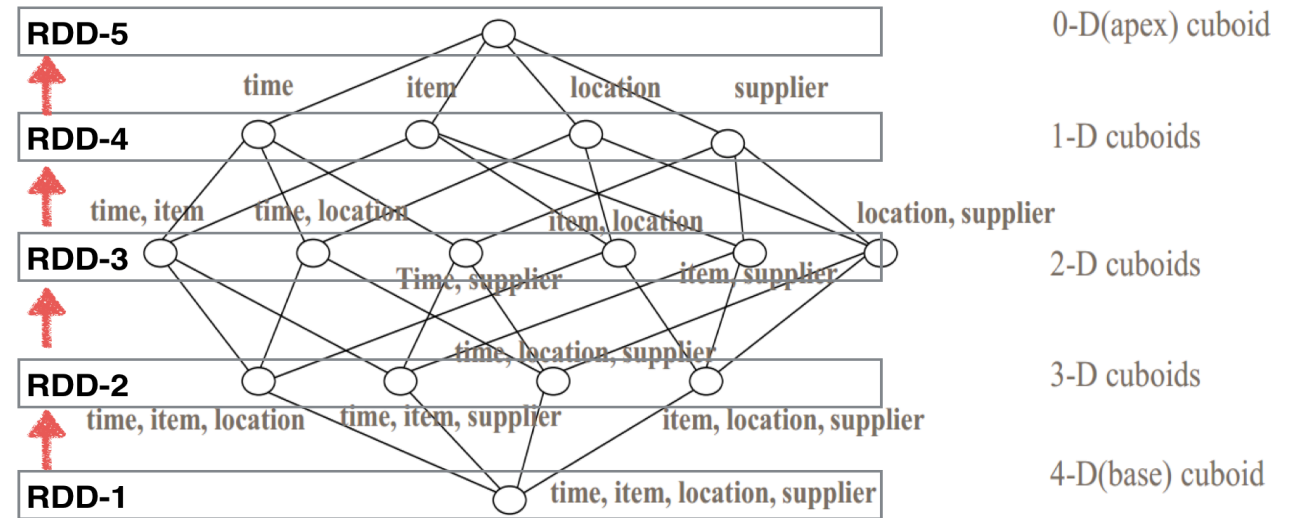
- Kylin automatically generate the cubing jobs, and then execute/track them.



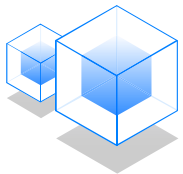


How Kylin build cube on Hadoop (cont.)

- Overall cubing steps:
 - Extract source data from Hive/Kafka/RDBMS to HDFS;
 - Build dimension dictionaries for encoding;
 - Build cuboids with MapReduce or Spark;
 - Convert cube to HBase format;
 - Load into HBase.



By-layer cubing



How Kylin persist cube into HBase

- Cube is composed by:
 - Cuboids
 - Dimensions
 - Metrics (measures)
- Cuboid ID + Dimension values → row key
- Metrics → HBase column value

Logical table for cuboid 00011111

Dimensions					Metrics		
D1	D2	D3	D4	D5	M1	M2	M2
a1	b1	c1	d1	e1	100	200	300
a2	b2	c2	d2	e2	200	400	600
a1	xxx	c1	yyy	e1	1	1	1

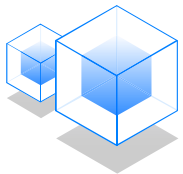
Logical table for cuboid 00010101

Dimensions			Metrics		
D1	D3	D5	M1	M2	M2
a1	c1	e1	101	201	301
a2	c2	e2	200	400	600

Row Key	
00010101	a1,c1,e1
00010101	a2,c2,e2

Metric
101,201,301
200,400,600

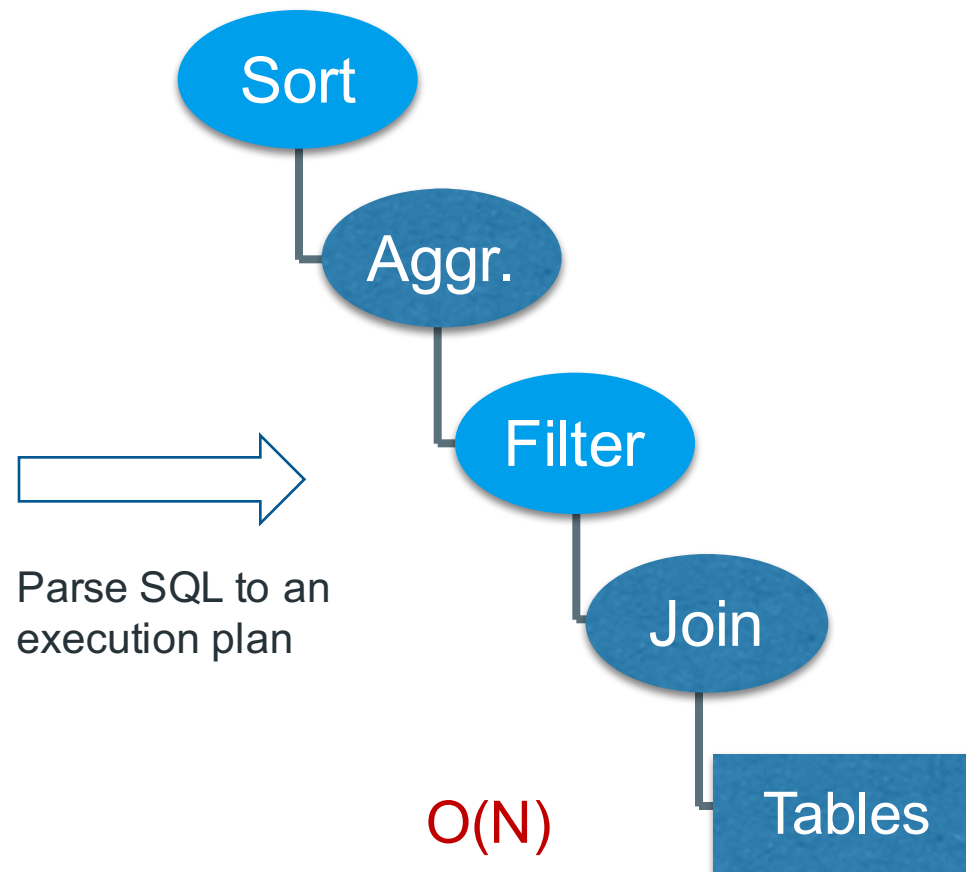
HBase schema

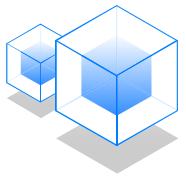


How Kylin query cube with SQL

- Kylin uses Apache Calcite as the SQL parser and optimizer;

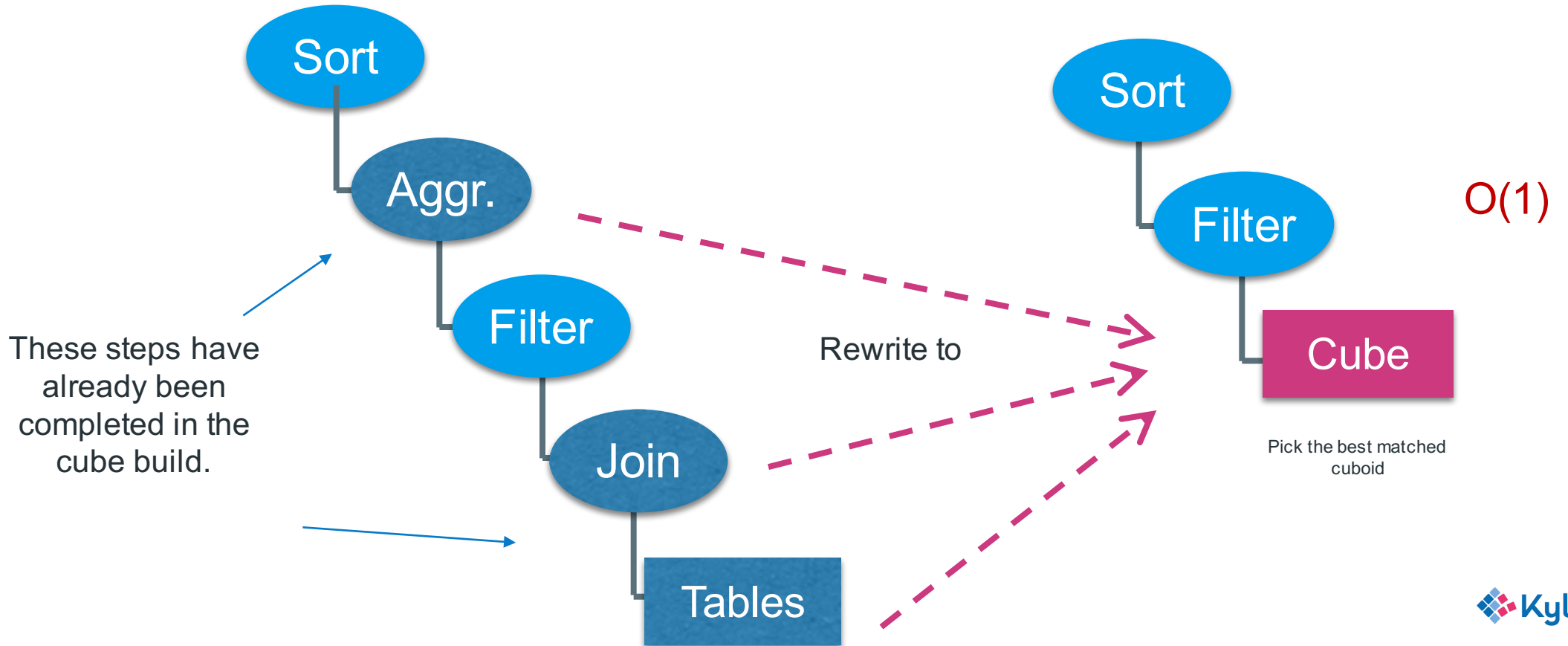
```
select
  l_returnflag,
  o_orderstatus,
  sum(l_quantity) as sum_qty,
  sum(l_extendedprice) as sum_base_price
from
  v_lineitem
  inner join v_orders on l_orderkey = o_orderkey
where
  l_shipdate <= '1998-09-16'
group by
  l_returnflag,
  o_orderstatus
order by
  l_returnflag,
  o_orderstatus;
```

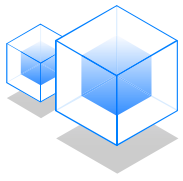




How Kylin query cube with SQL (cont.)

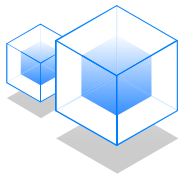
- Kylin optimize and adapt the plan to OLAP cube.
- With less processing, return result instantly.



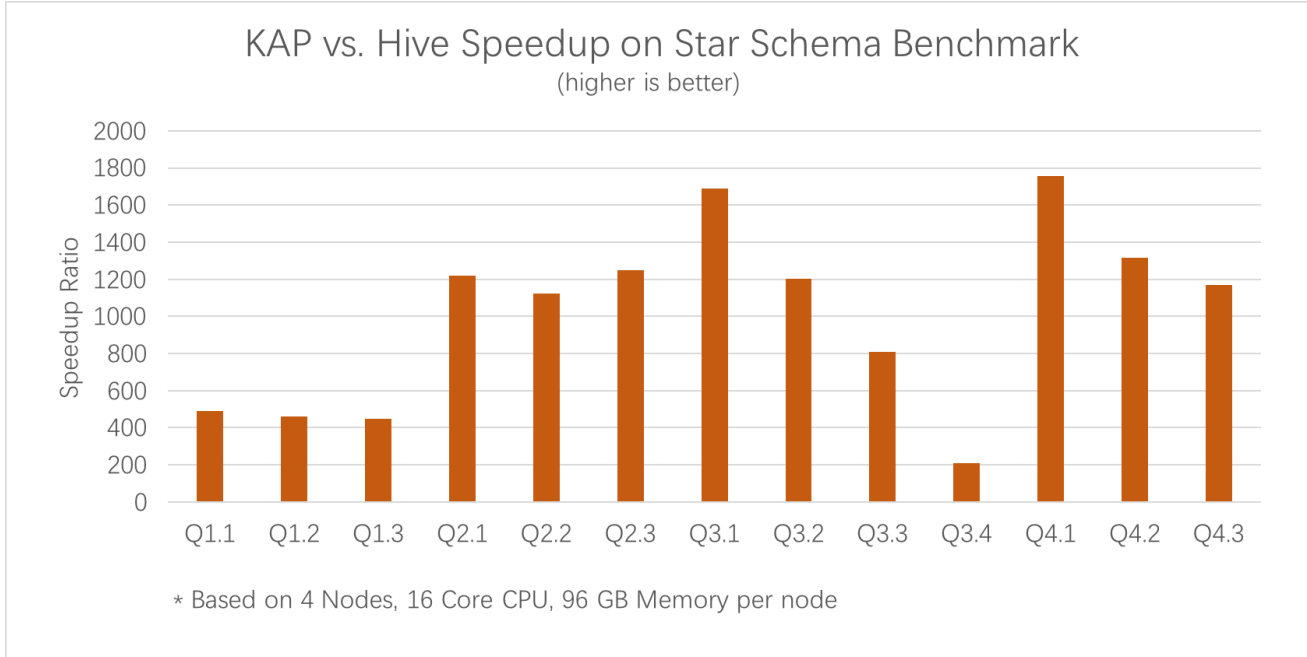


How Kylin query cube with SQL (cont.)

- Translate cube query into HBase table scan;
 - Group by, Filter -> Identify the best cuboid
 - Filters -> HBase scan ranges (row key start/end), fuzzy row filter
 - Aggregations -> Measure columns
- Scan HBase table and translate HBase result into cube result;
 - HBase Coprocessor is used for condition pushdown and region side aggregation.
 - HBase Result (row key + col value) decodes to cube result (dimensions + measures)
- Let Calcite do the final round processing (filtering, sorting, grouping, window, etc).

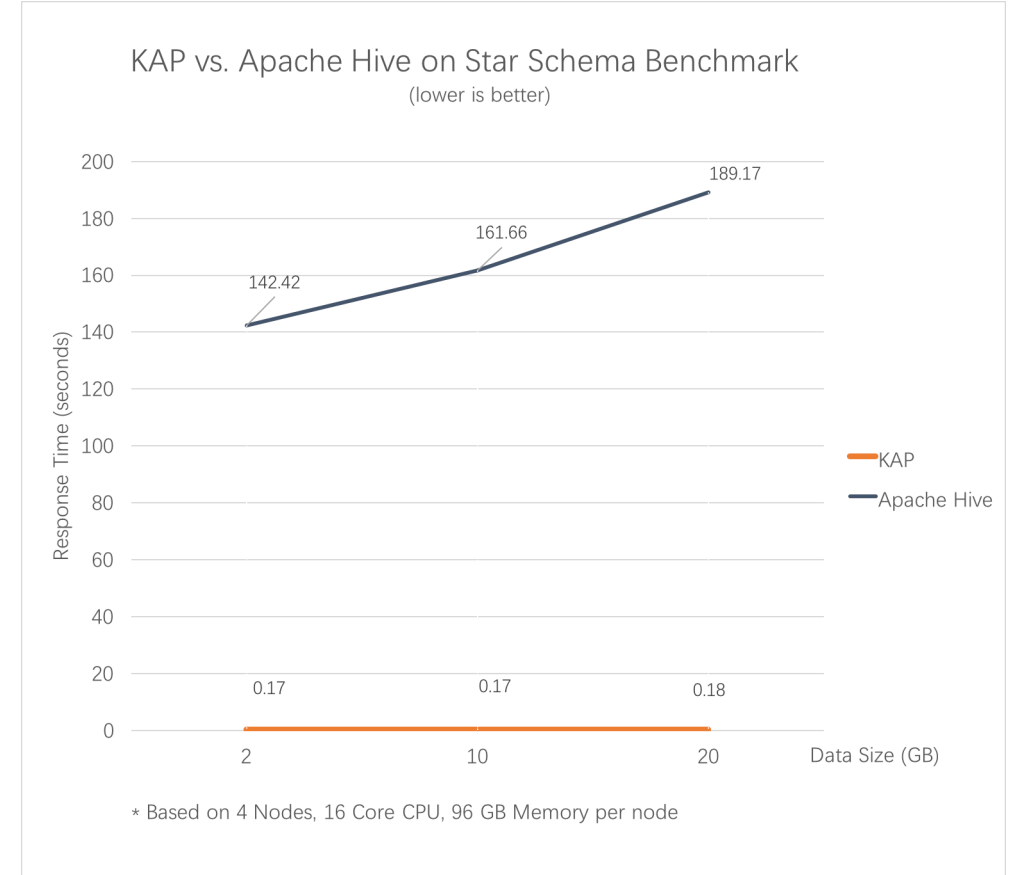


Performance benchmark



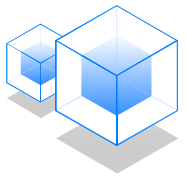
Tested on SSB dataset, 200X to 1700X faster than Apache Hive

KAP was the name of Kyligence release.



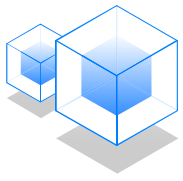
As data size increase, the response time keeps stable





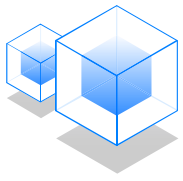
Advanced features

- Build a **partial cube** by manual (with rules) or automatically (cube planner);
- Support **UHC dimension** (e.g., cell number, cardinality > 100 million);
- Support **precisely count distinct on UHC** (with bitmap), and roll up at any level;
- Support fully and **incrementally data load**, also allow refreshing history data;
- Support **Kafka and RDBMS** (MySQL, Redshift etc) as data source;
- **Multi-nodes** deployment for HA;
- **Read-write separation** deployment (dedicated HBase) for high performance;
- The **Real-time OLAP** (v3.0) is at alpha now;



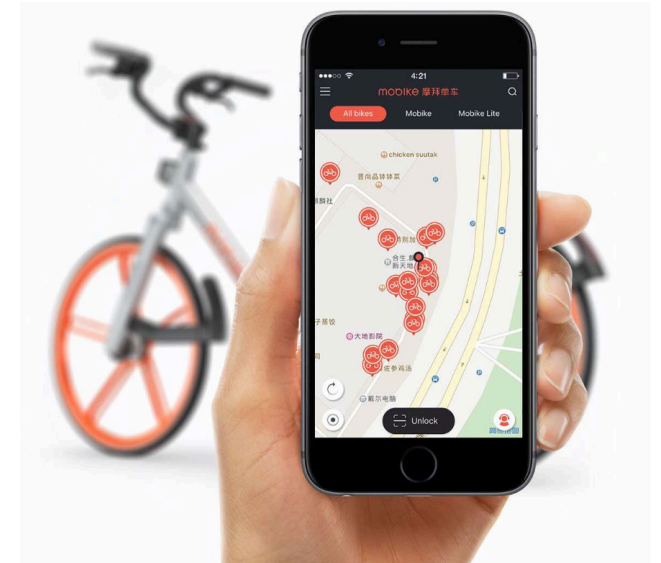
Scenarios for Apache Kylin

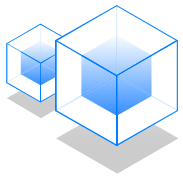
- ✓ Dashboard, reporting and business intelligence on big data (support **Tableau**, Cognos, MSTR, Qlik, Superset, Zeppelin...)
- ✓ Multi-dimensional data exploration
- ✓ Traditional data warehouse offload/migration to Hadoop
- ✓ User behavior analysis (PV, UV, retention, funnel, etc)
- ✓ Transparent query acceleration



Use case: Meituan & Dianping

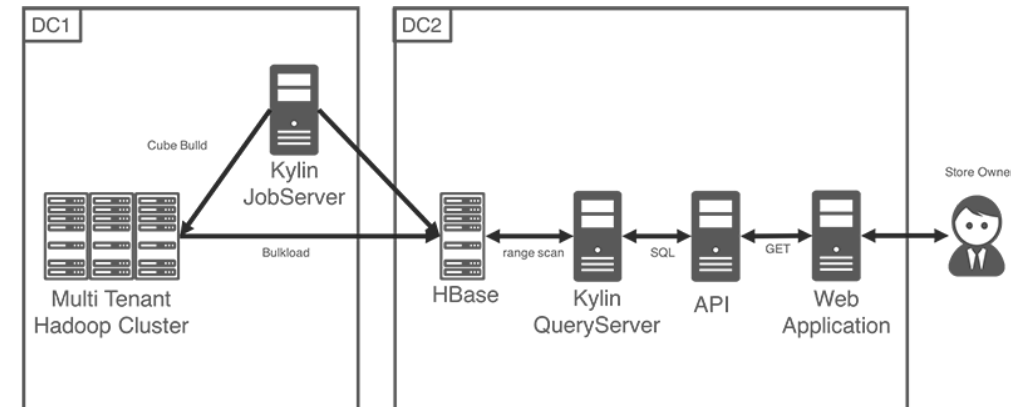
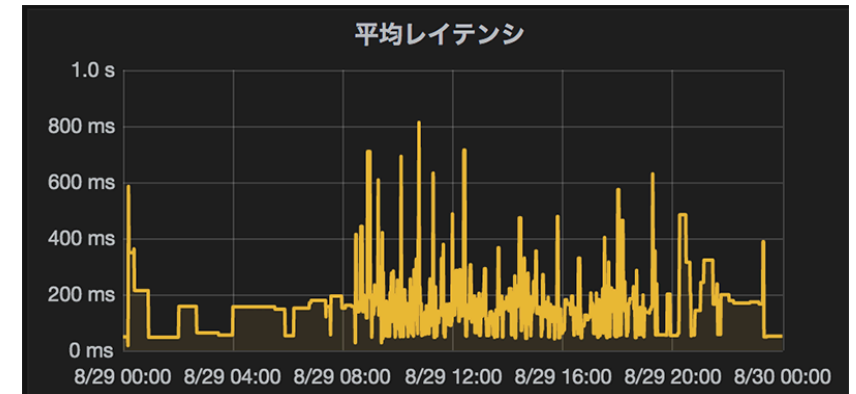
- The biggest O2O service provider in China
 - Catering takeaway, rating, hotel, travel, bike (mobike), etc;
 - More than 300 millions active users and 3 million merchants;
- Apache Kylin was selected as the **central OLAP platform** since 2015, serving thousands of business analysts from all business lines.
- Till Aug, 2018, total data row in Kylin is **8.9 trillion** , cube storage is **971 TB**.
- **3.8 million** queries per day; 50% queries < **200ms**; 90% queries < **1.2s**

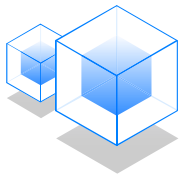




Use case: Yahoo! Japan

- Yahoo! Japan' s online shop data analysis.
- Use **Kylin to replace Impala**, fulfill the low latency requirement to business analysts;
- Query latency reduced **from 1 minute to <1s**;
- With Kylin' s cross-region deployment, cube instead of raw data is pushed to the DC that nearby the analysts, saved the bandwidth ;
- Read more:
<https://techblog.yahoo.co.jp/oss/apache-kylin/>

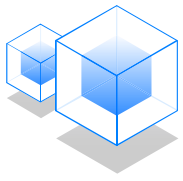




Use case: Telecoming

- About Telecoming
 - Mobile payments (Direct Carrier Billing) and digital marketing company in Spain.
- To meet the demand of analytics over the large volumes of data, deployed **Hadoop and Kylin** to support interactive analytical queries.
- “Thanks to this new architecture, Telecoming has improved the quality and productivity of its decision-making processes, which translates into **better results for their business.**”
- Read more: <http://www.stratebi.com/-/big-data-marketing-telecoming>



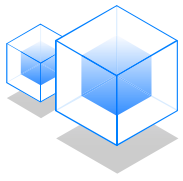


Use case: OLX Group

- About OLX Group
 - A global online marketplace, headquartered in Amsterdam, owned by Naspers, operating in 45 countries;
- Built DW with Amazon Redshift in the past; Redshift couldn't scale, moving to data lake on S3;
- Need a solution to enable analyst users doing interactive analysis with **Tableau**.
- Tried several solutions; finally, **Kylin + EMR** well solved the challenge!
- *"Kylin is the game changer with its extreme fast performance and seamless integration with Tableau."*

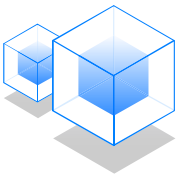


Picture from <https://www.olx.co.id/>

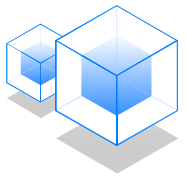


Useful links

- Apache Kylin home: <https://kylin.apache.org/>
- Apache Kylin source code: <https://github.com/apache/kylin>
- Join Kylin mailing list: dev@kylin.apache.org and user@kylin.apache.org
- Kylin meetup: <https://www.meetup.com/Apache-Kylin-Meetup/>
- Need enterprise solution? contact Kyligence: <https://kyligence.io/> or info@kyligence.io



See a demo?



Thank you!

谢谢!