SDCC
中国软件开发者大会
SOFTWARE DEVELOPER CONFERENCE CHINA

CSDN
全球最大中文IT社区

Palo分析型数据库在百度内的应用实践

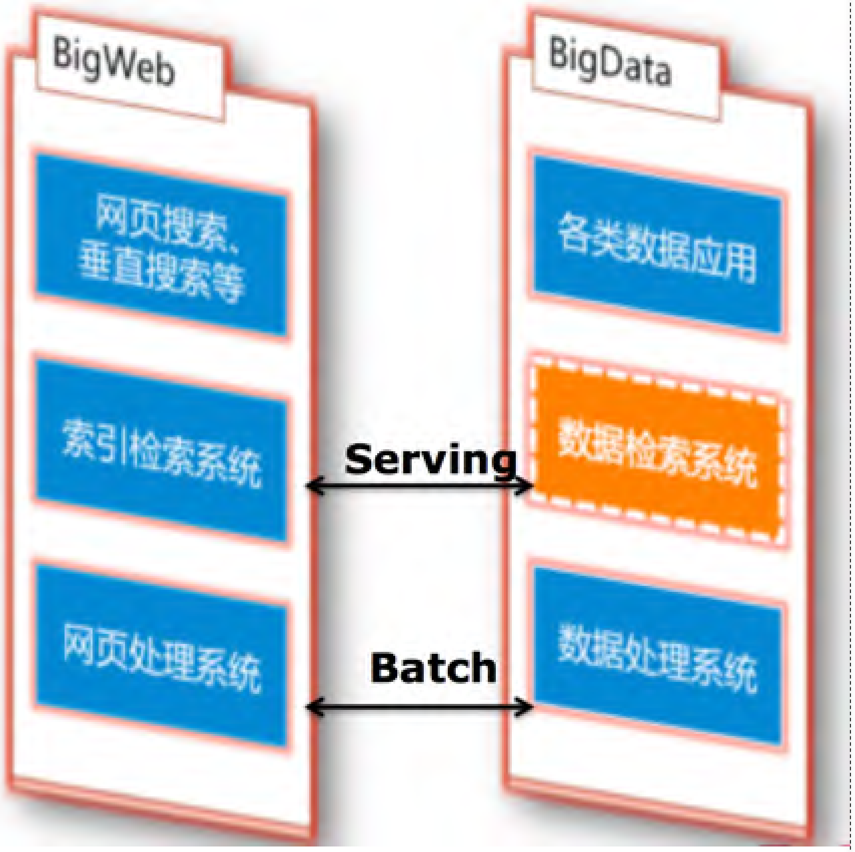
马如悦 2015.11

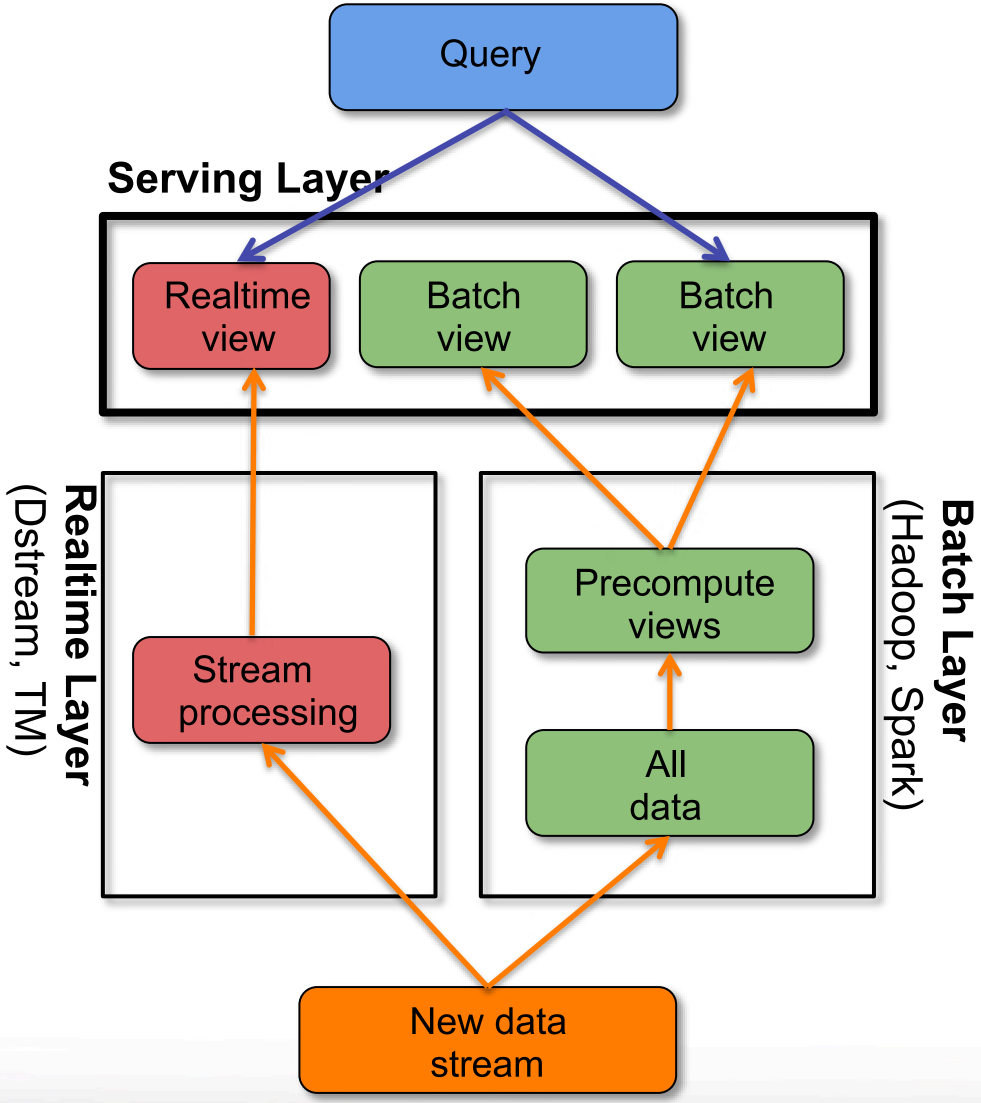
内容

- 背景介绍
- 使用场景@案例介绍
- 整体架构与使用介绍
- 关键技术
- 对外开放

背景介绍

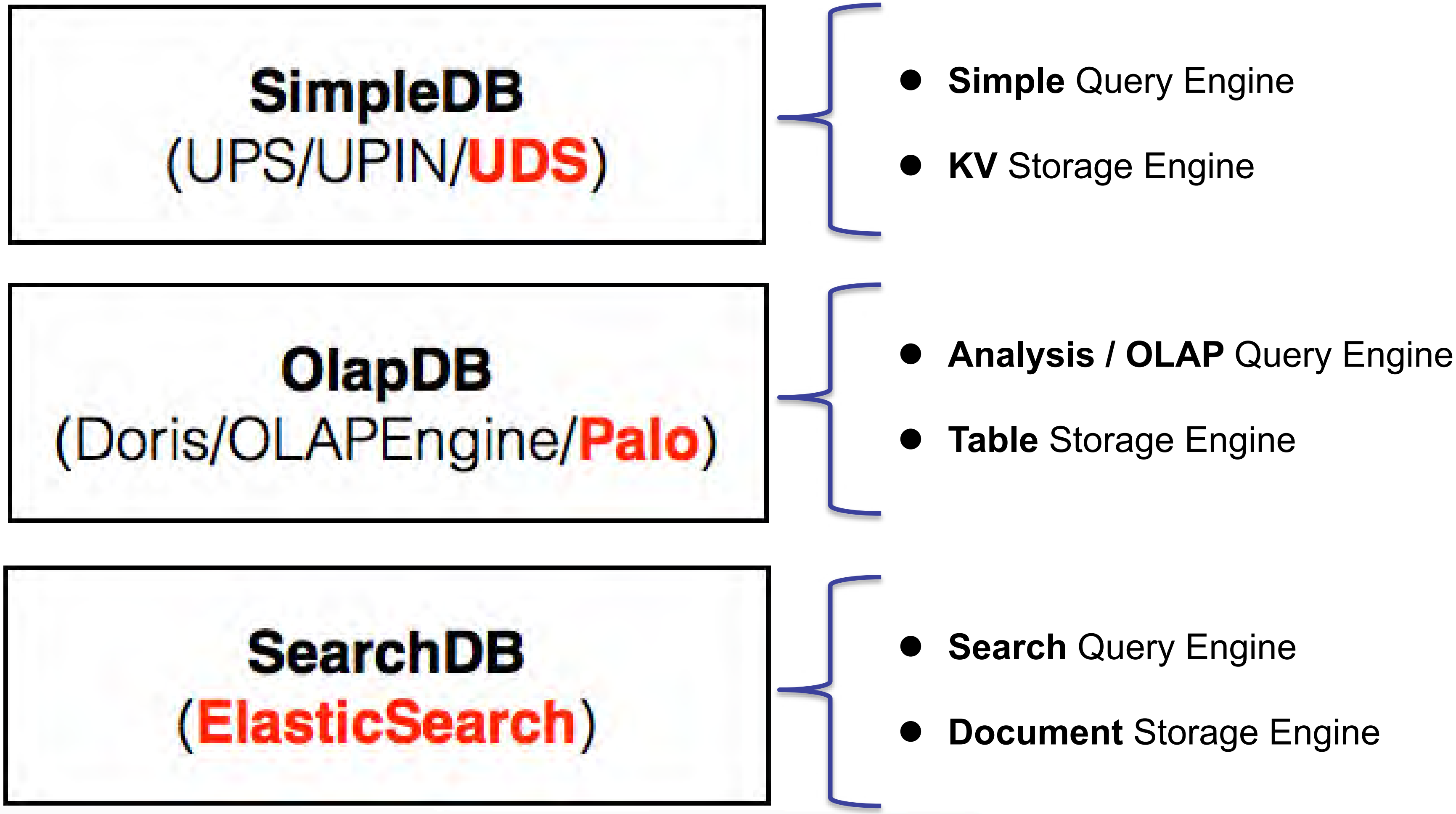
Online Data Serving





Bigdata Lambda Architecture

Online Data Serving



SimpleDB
(UPS/UPIN/**UDS**)

- **Simple** Query Engine
- **KV** Storage Engine

OlapDB
(Doris/OLAPEngine/**Palo**)

- **Analysis / OLAP** Query Engine
- **Table** Storage Engine

SearchDB
(**ElasticSearch**)

- **Search** Query Engine
- **Document** Storage Engine

Palo

- Palo名字来由 PALO <-> OLAP
- Online Analytical Processing
  - Analytical Processing vs. Transactional Processing
  - Online vs. Offline (Interactive vs. Batch)
- *A MPP-based Interactive Data Analysis SQL DB*
- 面向百TB ~ PB级别，结构化数据，毫秒/秒级分析
- 自研第三代产品：Doris -> OlapEngine -> Palo
- 120+产品线使用，500+台机器，单一业务最大百TB

OLAP vs. OLTP

	OLTP	OLAP
面向应用	日常交易处理	明细查询，分析决策
访问模式	简单小事务，操作少量数据	复杂聚合查询，查询大量数据
数据	当前最新数据	历史数据
数据规模	GB	TB ~ PB
数据更新	实时更新	批量更新
数据组织	满足3NF	反范式，星型模型

Palo定位

低成本

1/10 ~1/100 Cost

线性扩展

100~200节点 / 1000 TB

支持云化部署

高可用

99.9999 % Uptime

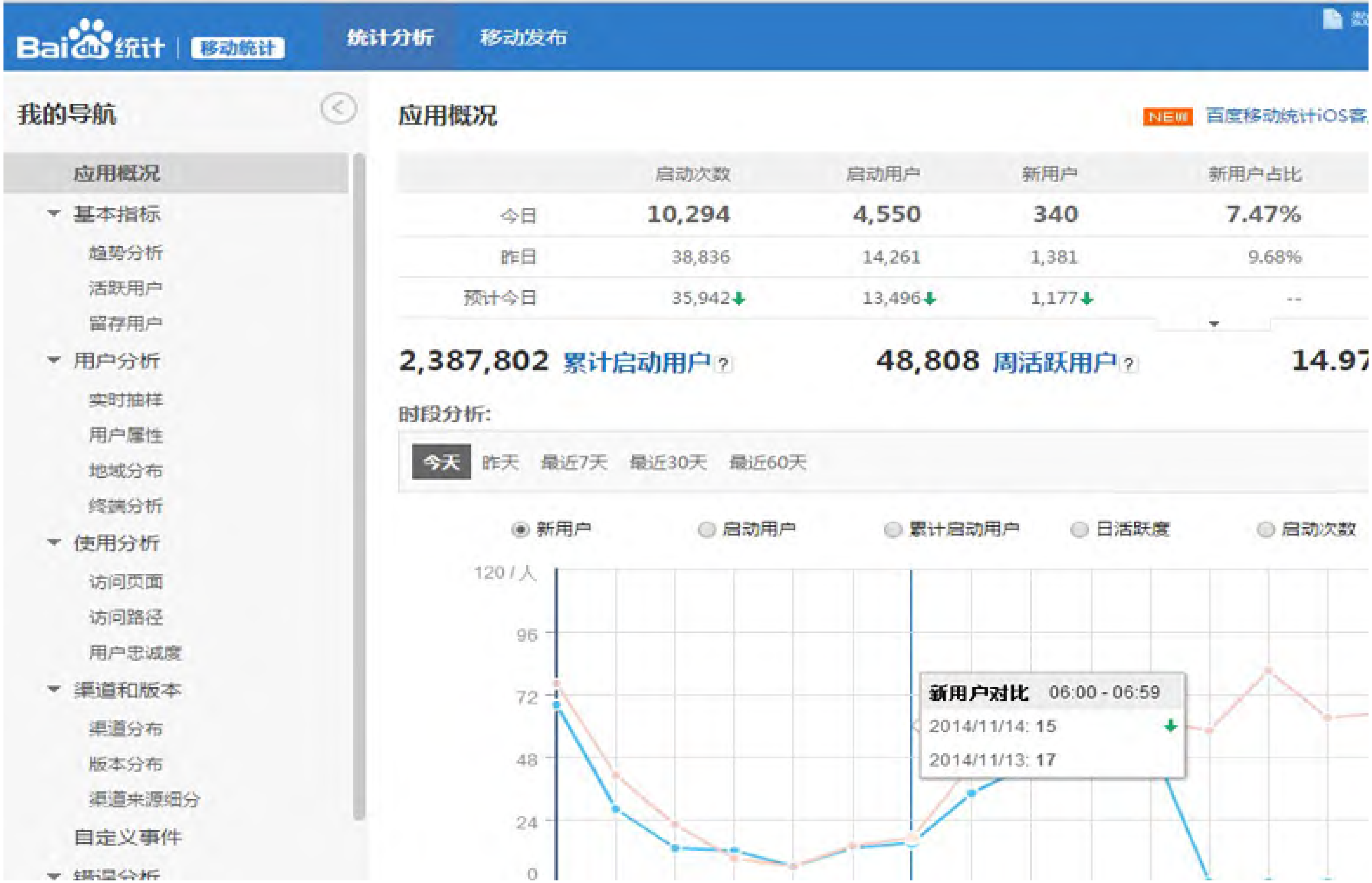
高查询性能

10W QPS/ 100GB/s

高加载性能

10 TB / Hour

OLAP – 在线报表



应用概况

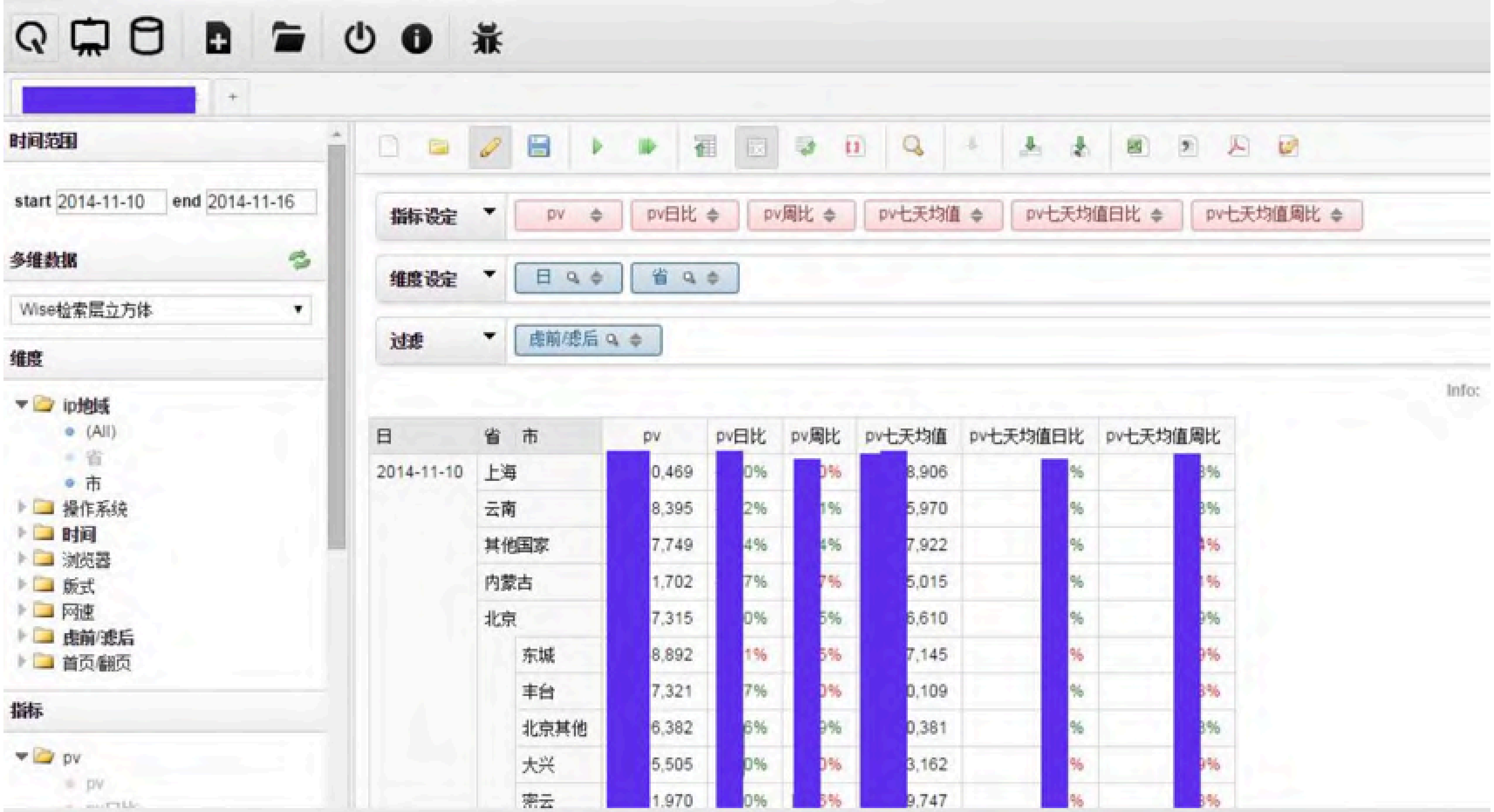
	启动次数	启动用户	新用户	新用户占比
今日	10,294	4,550	340	7.47%
昨日	38,836	14,261	1,381	9.68%
预计今日	35,942↓	13,496↓	1,177↓	--

2,387,802 累计启动用户 ?　　48,808 周活跃用户 ?　　14.97

时段分析:

新用户对比　06:00 - 06:59
2014/11/14: 15 ↓
2014/11/13: 17

OLAP – 多维分析



[redacted] +

时间范围

start 2014-11-10 end 2014-11-16

多维数据

Wise检索层立方体 ▼

维度

- ip地域
  - (All)
  - 省
  - 市
- 操作系统
- 时间
- 浏览器
- 版式
- 网速
- 点前/滤后
- 首页/翻页

指标

- pv
  - pv

指标设定 ▼ pv ⇕ pv日比 ⇕ pv周比 ⇕ pv七天均值 ⇕ pv七天均值日比 ⇕ pv七天均值周比 ⇕

维度设定 ▼ 日 ⇕ 省 ⇕

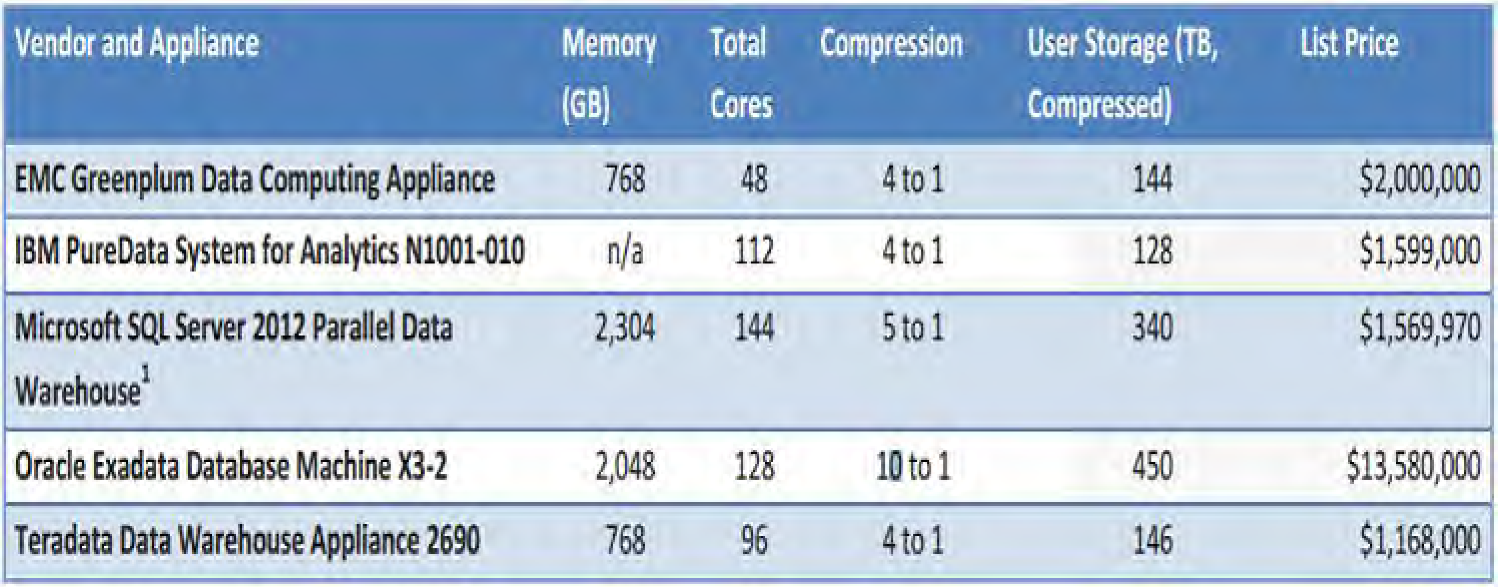
过滤 ▼ 点前/滤后 ⇕

Info:

日	省	市	pv	pv日比	pv周比	pv七天均值	pv七天均值日比	pv七天均值周比
2014-11-10	上海		[redacted]0,469	[redacted]0%	[redacted]0%	[redacted]8,906	[redacted]%	[redacted]3%
	云南		[redacted]8,395	[redacted]2%	[redacted]1%	[redacted]5,970	[redacted]%	[redacted]3%
	其他国家		[redacted]7,749	[redacted]4%	[redacted]4%	[redacted]7,922	[redacted]%	[redacted]4%
	内蒙古		[redacted]1,702	[redacted]7%	[redacted]7%	[redacted]5,015	[redacted]%	[redacted]1%
	北京		[redacted]7,315	[redacted]0%	[redacted]5%	[redacted]6,610	[redacted]%	[redacted]9%
		东城	[redacted]8,892	[redacted]1%	[redacted]5%	[redacted]7,145	[redacted]%	[redacted]9%
		丰台	[redacted]7,321	[redacted]7%	[redacted]0%	[redacted]0,109	[redacted]%	[redacted]3%
		北京其他	[redacted]6,382	[redacted]6%	[redacted]9%	[redacted]0,381	[redacted]%	[redacted]3%
		大兴	[redacted]5,505	[redacted]0%	[redacted]0%	[redacted]3,162	[redacted]%	[redacted]9%
		密云	[redacted]1,970	[redacted]0%	[redacted]5%	[redacted]9,747	[redacted]%	[redacted]3%

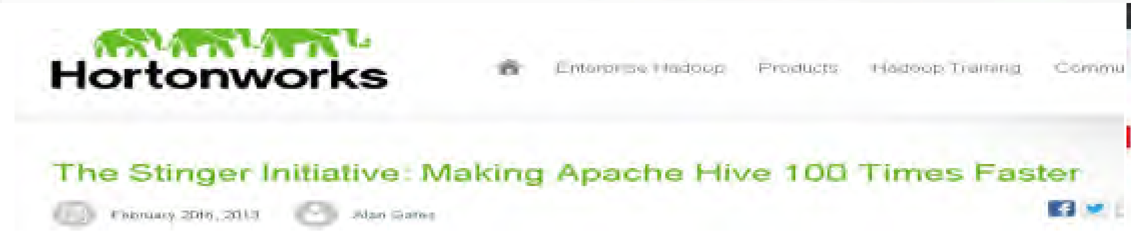
OLAP – 商业产品

产品	简介	技术特点	收购情况
Netezza	2000年在美国成立 Netezza TwinFin	✓ 软硬一体机 ✓ 采用FPGA数据过滤代替索引	2010年9月20日，IBM出资17.8亿美元收购
Greenplum	2003年在美国成立 Greenplum Database	✓ 行存 + 列存 ✓ Shared-Nothing集群	2010年7月6日，EMC出资3亿美元收购
Vertica	2005年在美国成立 Vertica Analytic Database	✓ 列存 ✓ Shared-Nothing集群	2011年2月，HP出资3.5亿美元收购
Aster Data	2005年在美国成立 nCluster	✓ SQL-MapReduce ✓ Shared-Nothing集群	2011年7月6日，Teradata出资2.63亿美元收购
ParAccel	**2005年在美国成立** **PADB**	✓ **列存 + 自适应压缩** ✓ **Shared-Nothing集群**	**2013年Actian出资1.5亿美元收购，Redshift宣称使用ParAccel**



Vendor and Appliance	Memory (GB)	Total Cores	Compression	User Storage (TB, Compressed)	List Price
EMC Greenplum Data Computing Appliance	768	48	4 to 1	144	$2,000,000
IBM PureData System for Analytics N1001-010	n/a	112	4 to 1	128	$1,599,000
Microsoft SQL Server 2012 Parallel Data Warehouse[1]	2,304	144	5 to 1	340	$1,569,970
Oracle Exadata Database Machine X3-2	2,048	128	10 to 1	450	$13,580,000
Teradata Data Warehouse Appliance 2690	768	96	4 to 1	146	$1,168,000

OLAP – 开源社区



The Stinger Initiative: Making Apache Hive 100 Times Faster

February 20th, 2013 Alan Gates

Cloudera Impala: Real-Time Queries in Apache Hadoop, For Real

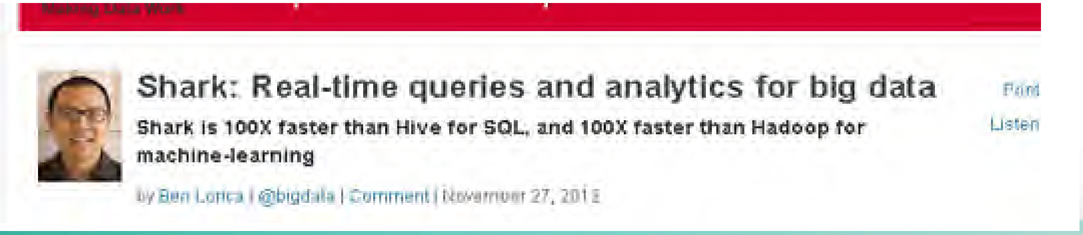
by Marcel Kornacker & Justin Erickson | October 24, 2012 | 13 comments | Tweet

Apache Drill Distributed system for interactive analysis.

Apache Drill (incubating) is a distributed system for interactive analysis of large-scale datasets, based on Google's Dremel. Its goal is to efficiently process nested data. It is a design goal to scale to 10,000 servers or more and to be able to process petabytes of data and trillions o records in seconds.

MemSQL, The Real-Time Analytics Platform.

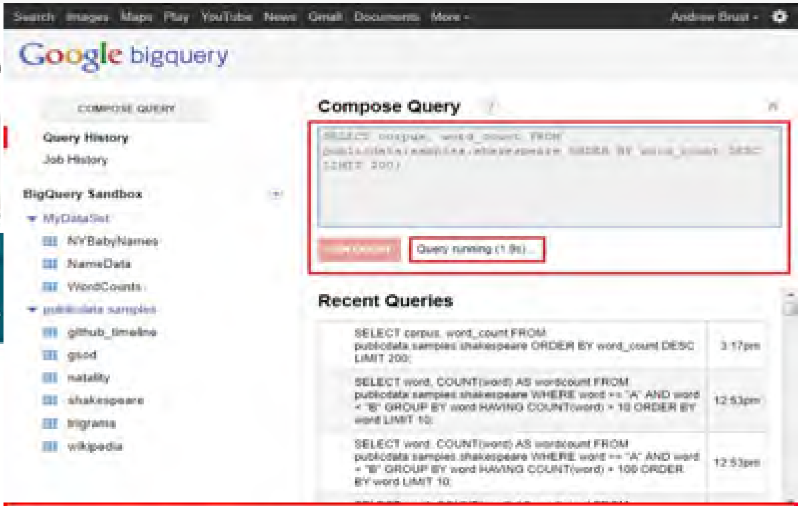
MemSQL's real-time analytics platform is built on the world's fastest, most scalable in-memory database, capable of simultaneously handling real-time transactions and analytic workloads. MemSQL unleashes the full potential of Big Data by consuming and returning data instantly



Shark: Real-time queries and analytics for big data

Shark is 100X faster than Hive for SQL, and 100X faster than Hadoop for machine-learning

by Ben Lorica | @bigdata | Comment | November 27, 2012

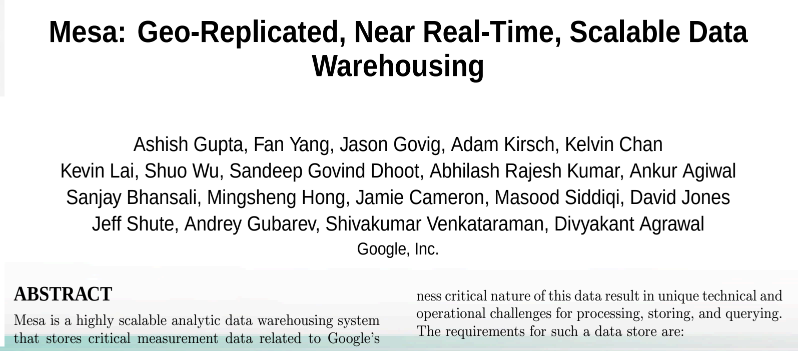


```
SELECT corpus, word_count FROM
publicdata:samples.shakespeare ORDER BY word_count DESC
LIMIT 200;
```

SELECT corpus, word_count FROM publicdata:samples.shakespeare ORDER BY word_count DESC LIMIT 200;	3:17pm
SELECT word, COUNT(word) AS wordcount FROM publicdata:samples.shakespeare WHERE word >= "A" AND word < "B" GROUP BY word HAVING COUNT(word) > 10 ORDER BY word LIMIT 10;	12:53pm
SELECT word, COUNT(word) AS wordcount FROM publicdata:samples.shakespeare WHERE word >= "A" AND word < "B" GROUP BY word HAVING COUNT(word) > 100 ORDER BY word LIMIT 10;	12:53pm

Introducing Amazon Redshift

A fast and powerful, fully managed petabyte-scale data warehouse service in the AWS Cloud



Mesa: Geo-Replicated, Near Real-Time, Scalable Data Warehousing

Ashish Gupta, Fan Yang, Jason Govig, Adam Kirsch, Kelvin Chan
Kevin Lai, Shuo Wu, Sandeep Govind Dhoot, Abhilash Rajesh Kumar, Ankur Agiwal
Sanjay Bhansali, Mingsheng Hong, Jamie Cameron, Masood Siddiqi, David Jones
Jeff Shute, Andrey Gubarev, Shivakumar Venkataraman, Divyakant Agrawal
Google, Inc.

ABSTRACT

Mesa is a highly scalable analytic data warehousing system that stores critical measurement data related to Google's ness critical nature of this data result in unique technical and operational challenges for processing, storing, and querying. The requirements for such a data store are:

为什么要做Palo

- 大家想要一套系统
  - 报表
  - 分析
  - 有时当个离线数据仓库也行
- 可能用到的系统
  - Mesa
  - Dremel
  - SparkSQL+HDFS
  - Impala+HDFS
  - Impala+Hbase
  - 传统MPP数据系统：teradata, vertica, greenplum
- 问题
  - 维护多个系统，多份数据
  - 功能不完备
  - 成本高
- 解决方案
  - Palo

适用场景和案例介绍

适用场景

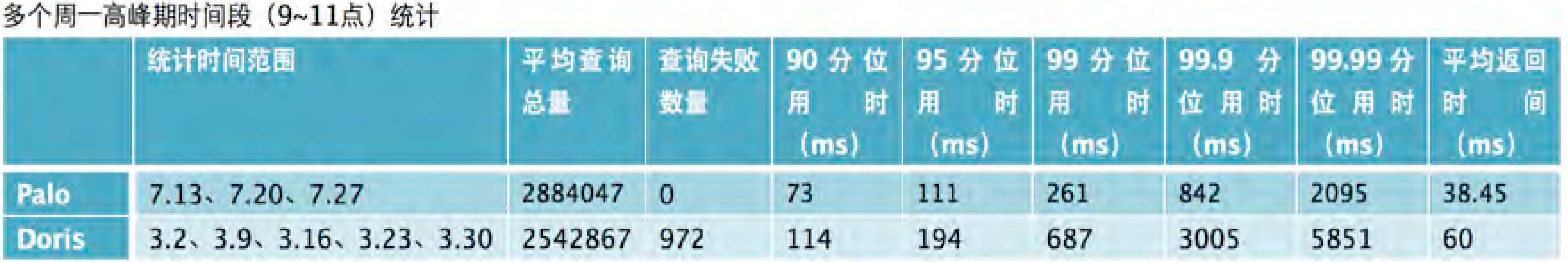
- 数据的统计分析统计
- 报表
  - MySQL存结果数据
  - 跑批处理，发送邮件
- 多维分析
  - Hadoop + Hive

业务推广

- 120+产品线
- 500+台
- 糯米、钱包、凤巢、移动等多个部门的BI报表和分析平台

案例分析 – 百度统计

- 百度统计
  - 为网站站长提供流量分析，网站分析，受众分析等多种分析服务
  - 450w网站, 每天查询量1500w，峰值QPS1400+
  - 300+表, 日导入数据量1TB+, 5分钟导入
  - 完成从Doris3->Palo迁移,机器数220+->58+，查询平均延时60+ ms-> 30ms

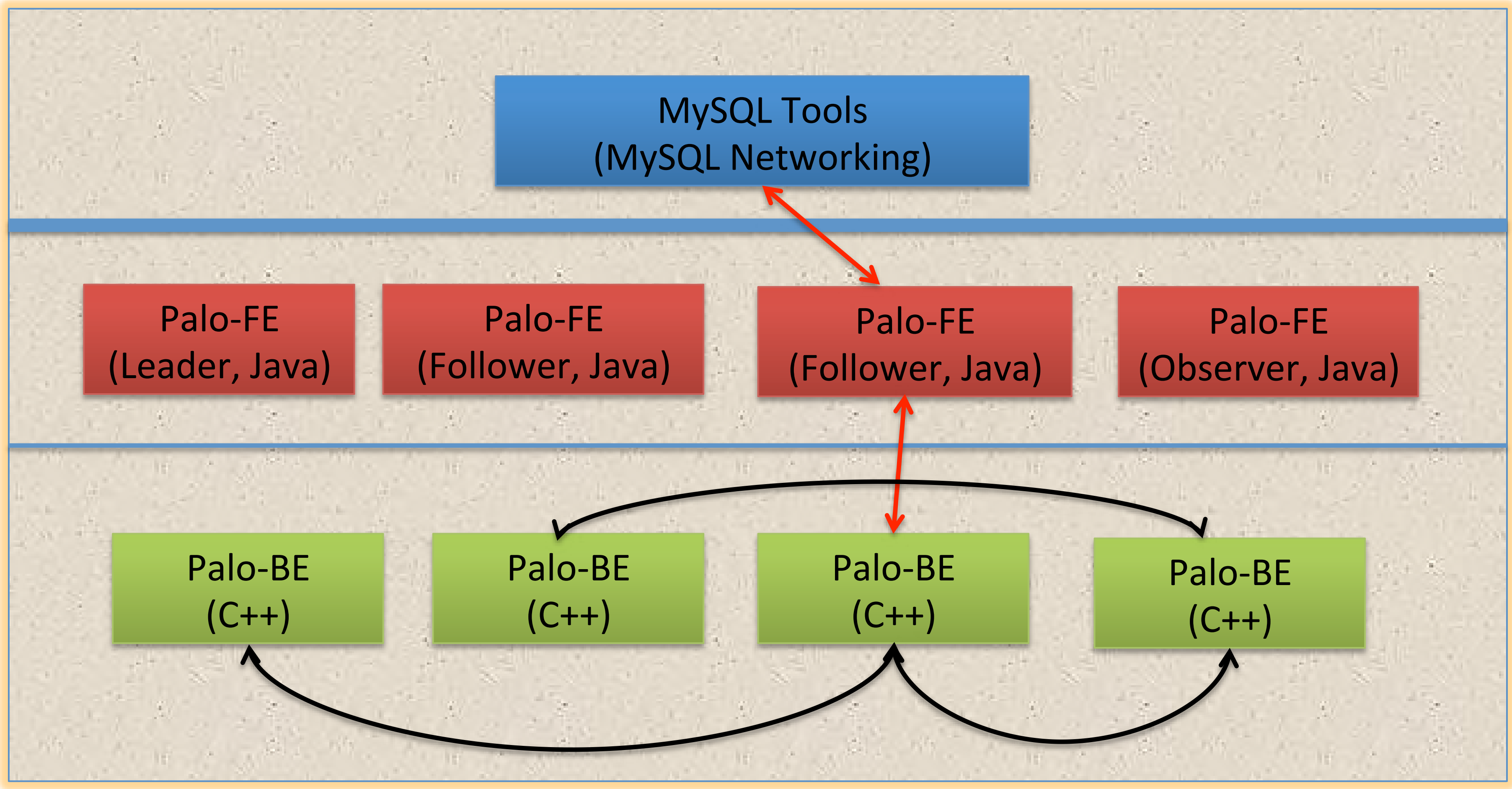


多个周一高峰期时间段（9~11点）统计

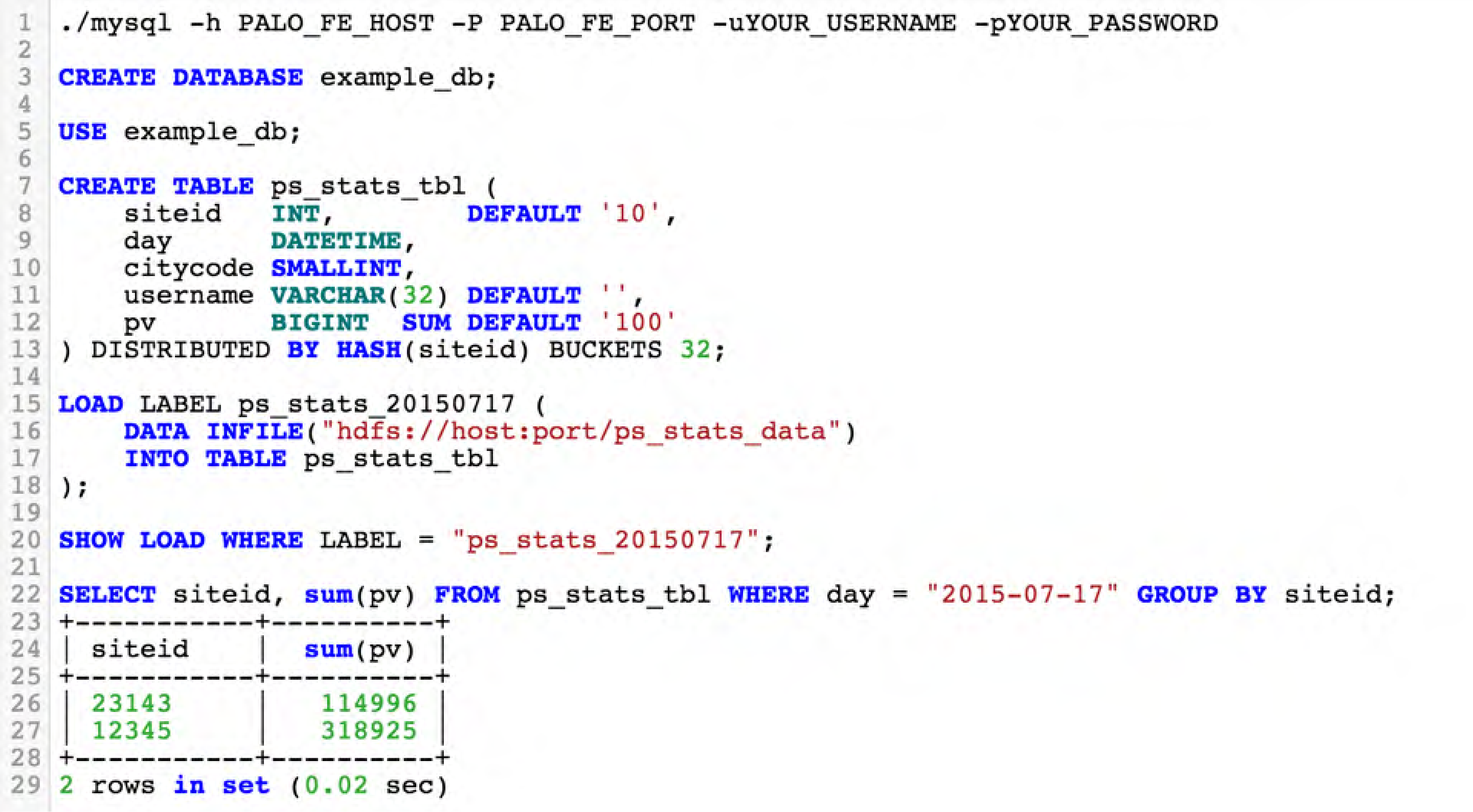
	统计时间范围	平均查询总量	查询失败数量	90分位用时（ms）	95分位用时（ms）	99分位用时（ms）	99.9分位用时（ms）	99.99分位用时（ms）	平均返回时间（ms）
Palo	7.13、7.20、7.27	2884047	0	73	111	261	842	2095	38.45
Doris	3.2、3.9、3.16、3.23、3.30	2542867	972	114	194	687	3005	5851	60

整体架构与使用介绍

Palo整体架构


MySQL Tools
(MySQL Networking)
Palo-FE
(Leader, Java)
Palo-FE
(Follower, Java)
Palo-FE
(Follower, Java)
Palo-FE
(Observer, Java)
Palo-BE
(C++)
Palo-BE
(C++)
Palo-BE
(C++)
Palo-BE
(C++)

Palo使用

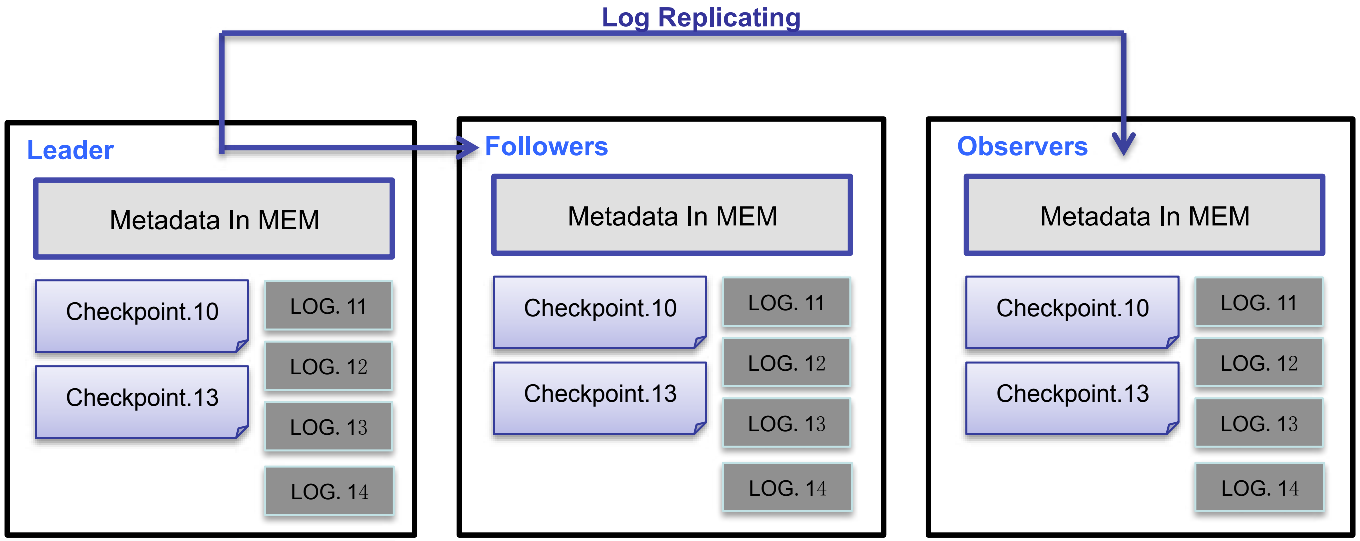


```
./mysql -h PALO_FE_HOST -P PALO_FE_PORT -uYOUR_USERNAME -pYOUR_PASSWORD

CREATE DATABASE example_db;

USE example_db;

CREATE TABLE ps_stats_tbl (
    siteid   INT,        DEFAULT '10',
    day      DATETIME,
    citycode SMALLINT,
    username VARCHAR(32) DEFAULT '',
    pv       BIGINT  SUM DEFAULT '100'
) DISTRIBUTED BY HASH(siteid) BUCKETS 32;

LOAD LABEL ps_stats_20150717 (
    DATA INFILE("hdfs://host:port/ps_stats_data")
    INTO TABLE ps_stats_tbl
);

SHOW LOAD WHERE LABEL = "ps_stats_20150717";

SELECT siteid, sum(pv) FROM ps_stats_tbl WHERE day = "2015-07-17" GROUP BY siteid;
+-----------+----------+
| siteid    |  sum(pv) |
+-----------+----------+
| 23143     |   114996 |
| 12345     |   318925 |
+-----------+----------+
2 rows in set (0.02 sec)
```

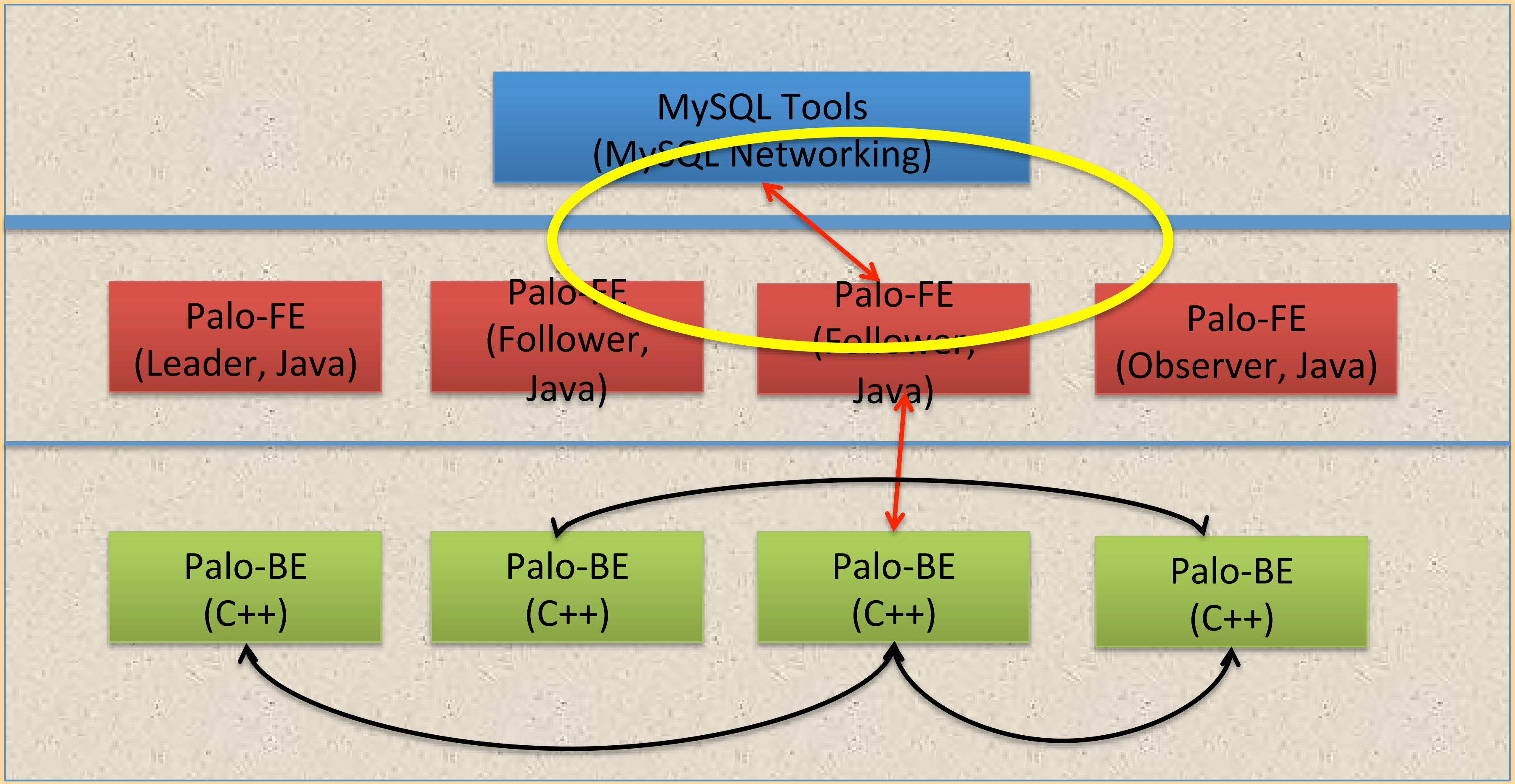
关键技术

容错

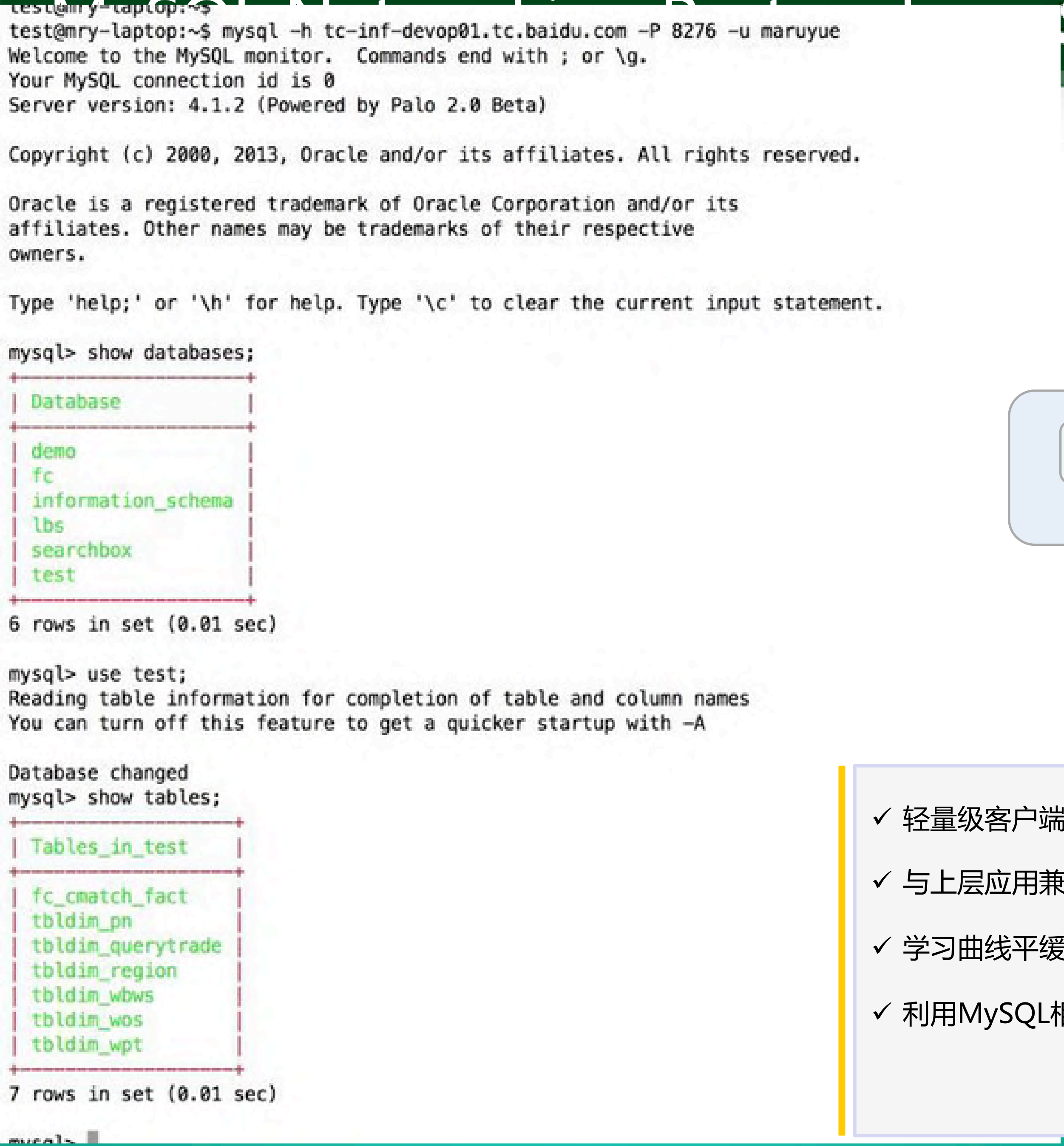
- 元数据
  - Memory + Checkpoint + Journal
  - 类Raft协议实现，高可靠&高可用性
- 数据
  - 多副本
  - 自动修复



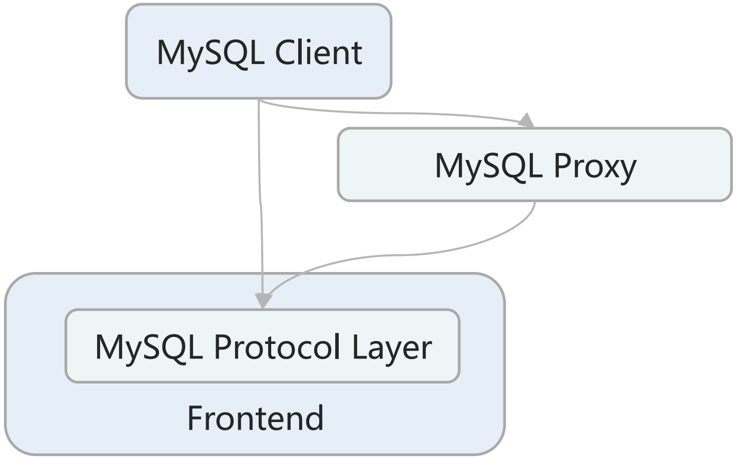
MySQL Networking Protocol



MySQL Networking Protocol

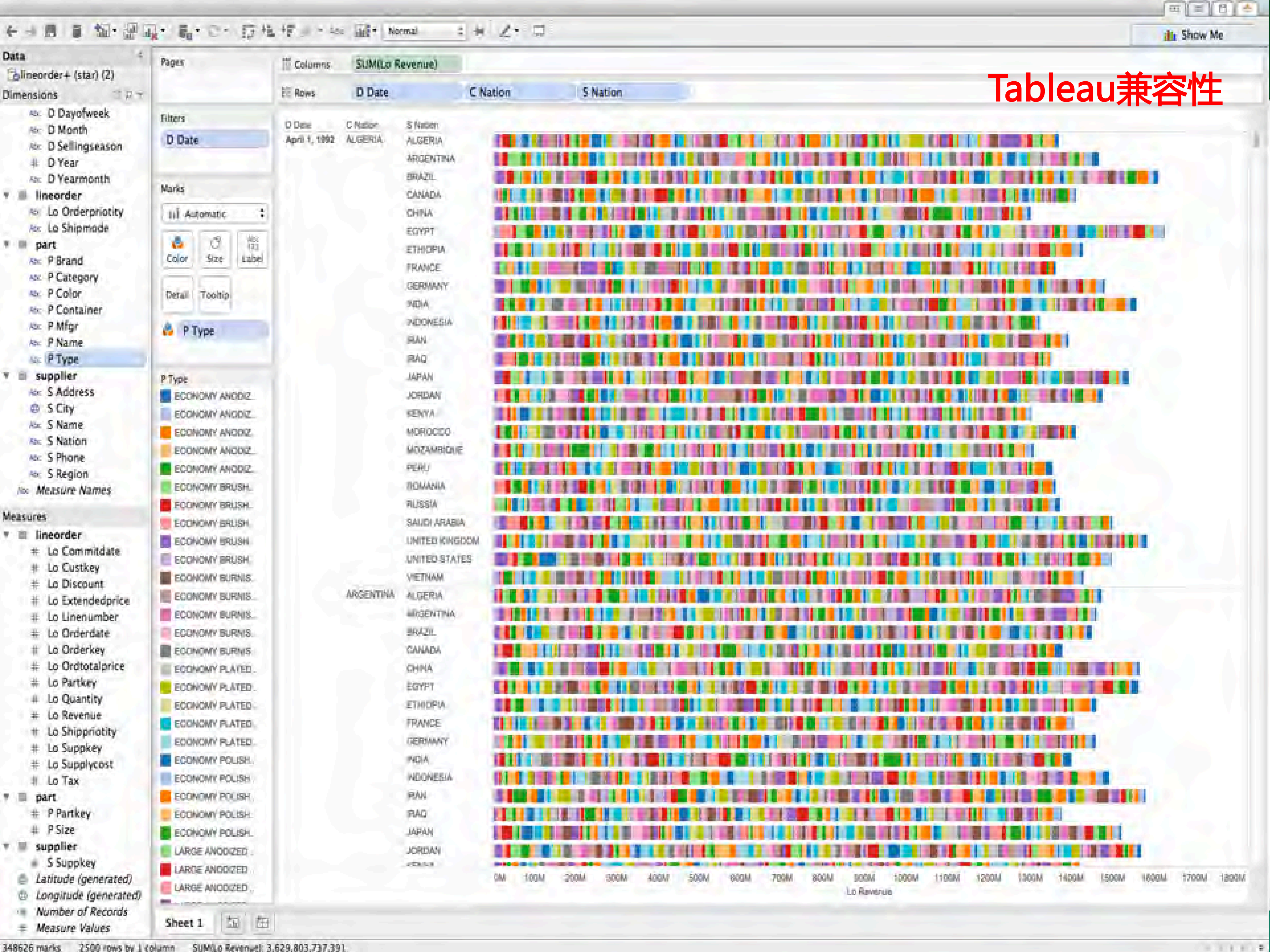


```
test@mry-laptop:~$
test@mry-laptop:~$ mysql -h tc-inf-devop01.tc.baidu.com -P 8276 -u maruyue
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 0
Server version: 4.1.2 (Powered by Palo 2.0 Beta)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| demo               |
| fc                 |
| information_schema |
| lbs                |
| searchbox          |
| test               |
+--------------------+
6 rows in set (0.01 sec)

mysql> use test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-------------------+
| Tables_in_test    |
+-------------------+
| fc_cmatch_fact    |
| tbldim_pn         |
| tbldim_querytrade |
| tbldim_region     |
| tbldim_wbws       |
| tbldim_wos        |
| tbldim_wpt        |
+-------------------+
7 rows in set (0.01 sec)

mysql> 
```



- ✓ 轻量级客户端
- ✓ 与上层应用兼容容易
- ✓ 学习曲线平缓，方便用户上手使用
- ✓ 利用MySQL相关工具，比如MySQL Proxy

Tableau兼容性



Columns: SUM(Lo Revenue)

Rows: D Date | C Nation | S Nation

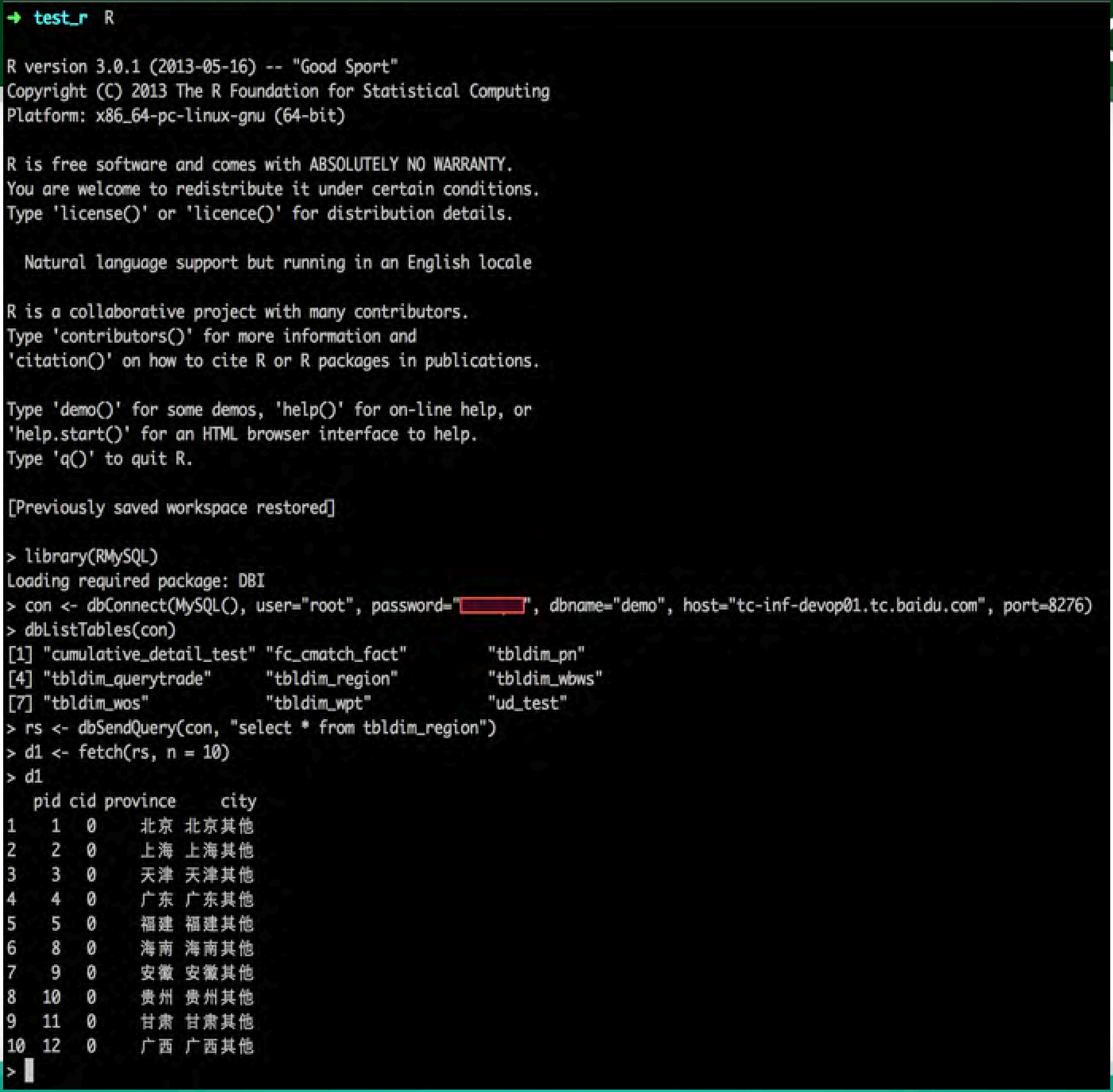
Filters: D Date

Marks: Automatic

Color: P Type

348626 marks   2500 rows by 1 column   SUM(Lo Revenue): 3,629,803,737,391

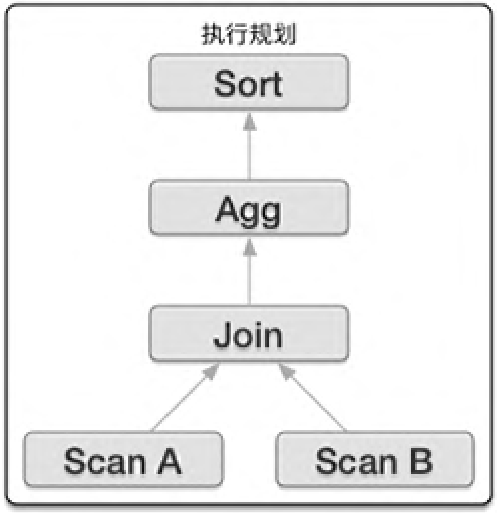
R语言兼容性

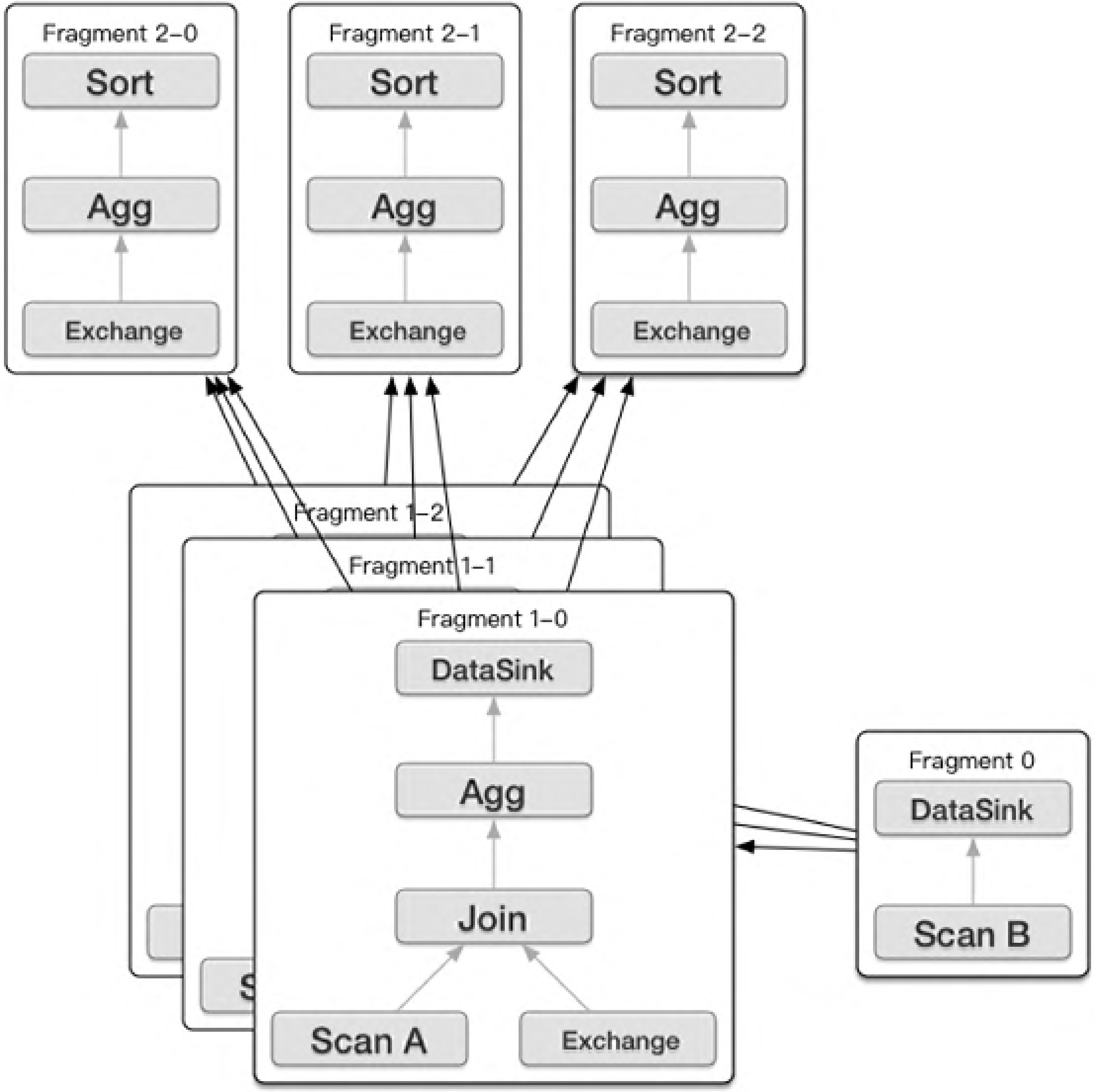


```
➜ test_r  R

R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> library(RMySQL)
Loading required package: DBI
> con <- dbConnect(MySQL(), user="root", password="[redacted]", dbname="demo", host="tc-inf-devop01.tc.baidu.com", port=8276)
> dbListTables(con)
[1] "cumulative_detail_test" "fc_cmatch_fact"         "tbldim_pn"
[4] "tbldim_querytrade"      "tbldim_region"          "tbldim_wbws"
[7] "tbldim_wos"             "tbldim_wpt"             "ud_test"
> rs <- dbSendQuery(con, "select * from tbldim_region")
> d1 <- fetch(rs, n = 10)
> d1
   pid cid province     city
1    1   0     北京 北京其他
2    2   0     上海 上海其他
3    3   0     天津 天津其他
4    4   0     广东 广东其他
5    5   0     福建 福建其他
6    8   0     海南 海南其他
7    9   0     安徽 安徽其他
8   10   0     贵州 贵州其他
9   11   0     甘肃 甘肃其他
10  12   0     广西 广西其他
> 
```

MPP

```
SELECT k1, SUM(v1)
FROM A, B
WHERE A.k2 = B.k2
GROUP BY k1
ORDER BY SUM(v1)
```





数据模型 - 1

Time	Id	Country	Clicks	Cost
2013/12/31	1	US	10	32
2014/01/01	2	UK	40	20
2014/01/01	2	US	150	80

- Key列，Value列
- Key列全局有序
  - 查询快速定位
- 全Key全局唯一
  - 相同Key的行，其Value列合并 (SUM,MIN,MAX,REPLACE)

数据模型 - 2

	Time	Id	Country	Clicks	Cost
Base	2013/12/31	1	US	10	32
	2014/01/01	2	UK	40	20
	2014/01/01	2	US	150	80

+

	Time	Id	Country	Clicks	Cost
Delta	2014/01/01	1	US	5	3
	2014/01/01	2	UK	60	30
	2014/01/01	2	US	50	20

数据模型 - 3

	Time	Id	Country	Clicks	Cost
New Base	2013/12/31	1	US	10	32
	2014/01/01	1	US	+5	+3
	2014/01/01	2	UK	40+60	20+30
	2014/01/01	2	US	150+50	80+20

	Time	Id	Country	Clicks	Cost
Delta	2014/01/01	1	US	5	3
	2014/01/01	2	UK	60	30
	2014/01/01	2	US	50	20

数据模型 - 4

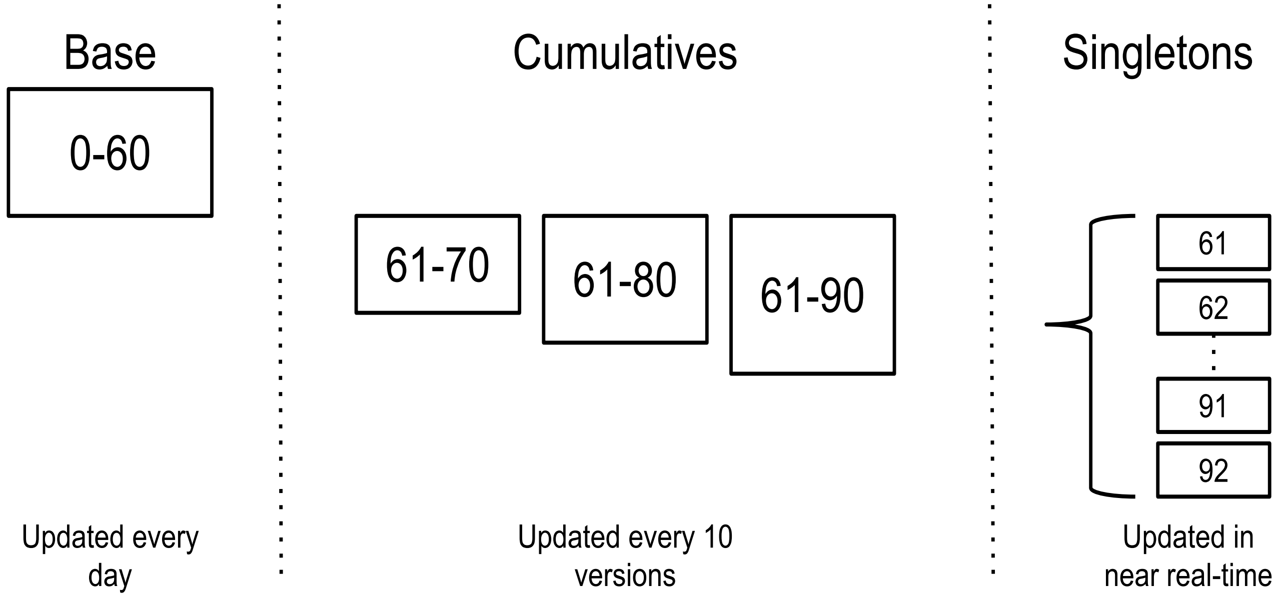
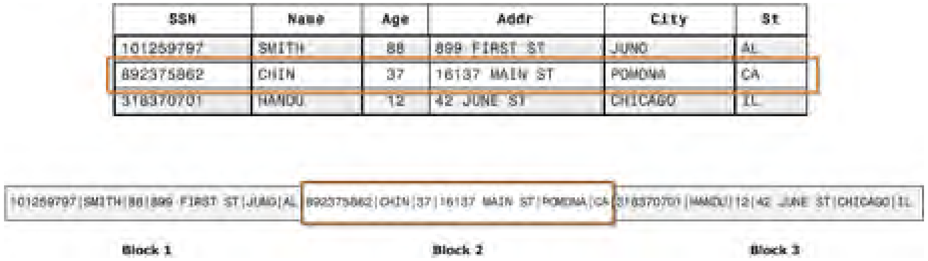


Figure 3: A two level delta compaction policy

引自Google Mesa Paper

列式存储

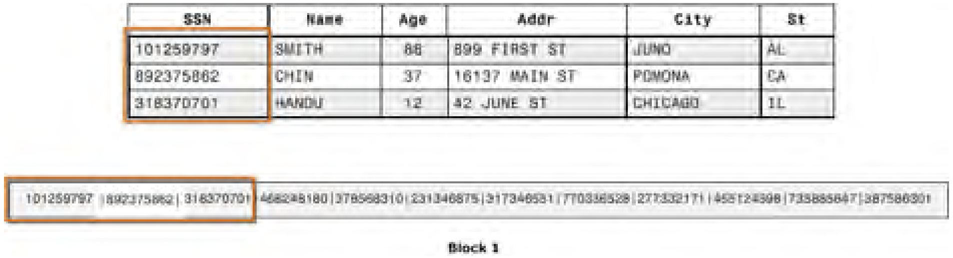


SSN	Name	Age	Addr	City	St
101259797	SMITH	88	899 FIRST ST	JUNO	AL
892375862	CHIN	37	16137 MAIN ST	POMONA	CA
318370701	HANDU	12	42 JUNE ST	CHICAGO	IL

```
101259797|SMITH|88|899 FIRST ST|JUNO|AL 892375862|CHIN|37|16137 MAIN ST|POMONA|CA 318370701|HANDU|12|42 JUNE ST|CHICAGO|IL
```

行存储

- ✓ 数据是按行存储的
- ✓ 没有索引的查询使用大量I/O
- ✓ 建立索引和物化视图需要花费大量时间和资源
- ✓ 面对查询的需求，数据库必须被大量膨胀才能满足性能要求



SSN	Name	Age	Addr	City	St
101259797	SMITH	88	899 FIRST ST	JUNO	AL
892375862	CHIN	37	16137 MAIN ST	POMONA	CA
318370701	HANDU	12	42 JUNE ST	CHICAGO	IL

```
101259797 |892375862| 318370701|468248180|378568310|231346875|317346531|770336528|277332171|455124398|735885647|387586301
```

列存储

- ✓ 数据按列存储，每一列单独存放
- ✓ 只访问查询涉及的列，大量降低I/O
- ✓ 数据类型一致，方便压缩
- ✓ 数据包建索引，数据即索引

Rollup Table

时间	Id	省份	pv
2014.01.01	1	北京	10
2014.01.01	2	天津	30
2014.01.02	1	北京	20
2014.01.02	2	北京	40

重新排序

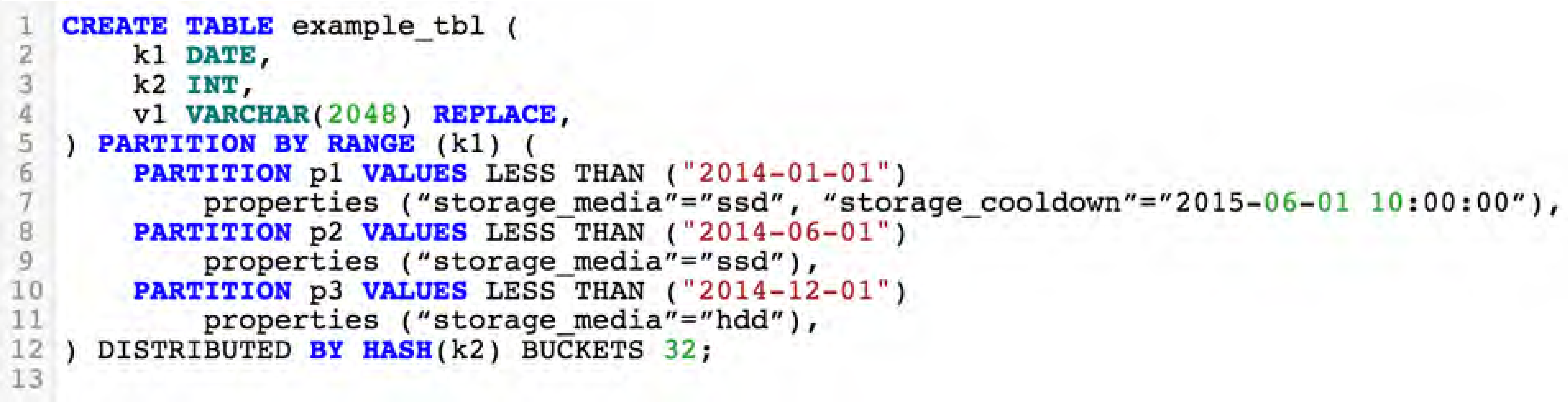
Id	时间	省份	pv
1	2014.01.01	北京	10
1	2014.01.02	北京	20
2	2014.01.01	天津	30
2	2014.01.02	北京	40

聚合表

Id	pv
1	30
2	70

两层分区与分级存储

- 两层分区
  - 方便新旧数据分离，使用不同的存储介质（新数据SSD，历史数据SATA）
  - 减少了大量历史数据不必要的重复BE/CE，节省了大量的IO和CPU开销
  - 简化了表的扩容，shard调整
- 分级存储
  - 用户可以指定数据放到SSD上或者SATA盘上，也支持根据TTL将冷数据从SSD迁移到SATA上，高效利用SSD提高查询性能

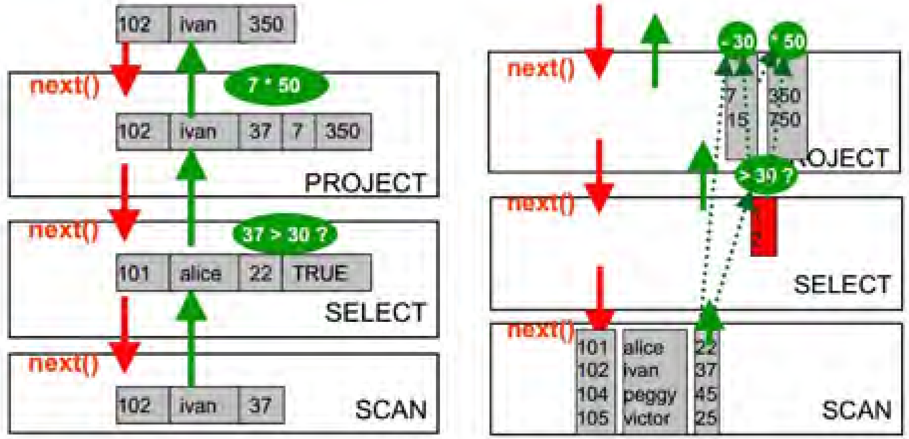


```
CREATE TABLE example_tbl (
    k1 DATE,
    k2 INT,
    v1 VARCHAR(2048) REPLACE,
) PARTITION BY RANGE (k1) (
    PARTITION p1 VALUES LESS THAN ("2014-01-01")
        properties (“storage_media”=”ssd”, “storage_cooldown”=”2015-06-01 10:00:00”),
    PARTITION p2 VALUES LESS THAN ("2014-06-01")
        properties (“storage_media”=”ssd”),
    PARTITION p3 VALUES LESS THAN ("2014-12-01")
        properties (“storage_media”=”hdd”),
) DISTRIBUTED BY HASH(k2) BUCKETS 32;

```

向量化执行

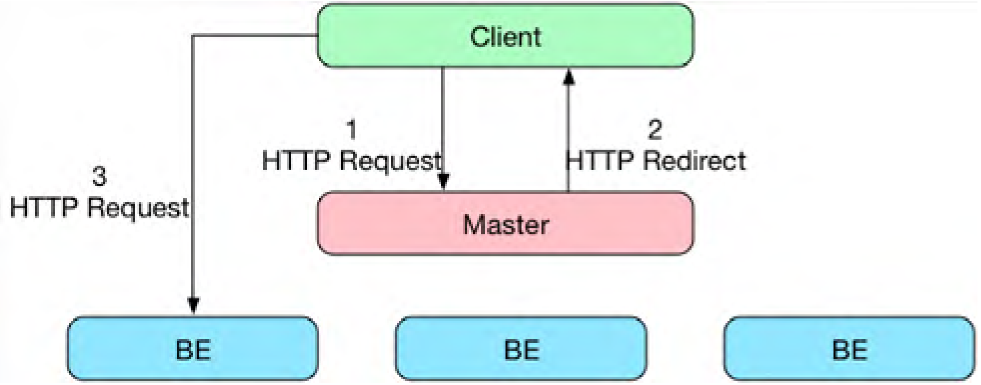
- **行式执行引擎问题**
  - 每行一次函数调用，打断CPU流水，不利于分支预测
  - 指令和数据cache miss
  - 编译器不友好，不利于循环展开，SIMD
- **设计思想**
  - 单条处理到批量处理
  - 行式处理转化为列式处理
- **效果**
  - star-schema测试整体提升3~4倍

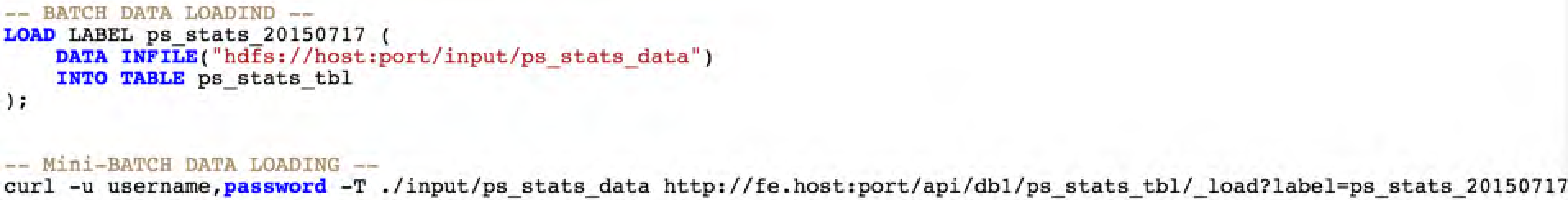
```
SELECT  id, name
        (age-30)*50 AS bonus
FROM    employee
WHERE   age > 30
```



Mini-Batch数据导入

- 补充原来基于Hadoop的Bulk-Batch导入
- Mini-Batch Data Loading
- 使用使用HTTP即可导入，减少客户端对其它组件的依赖
- 实现了多导入的事务提交



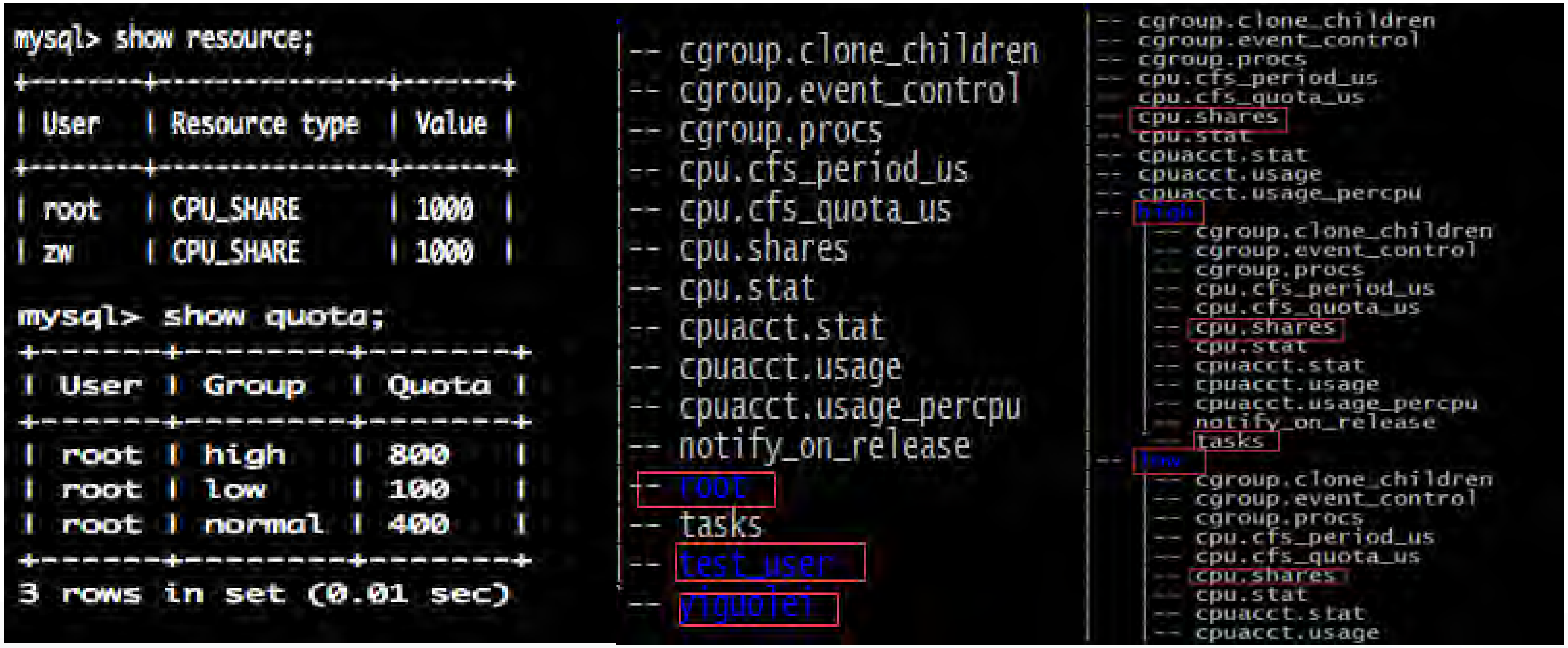


```
-- BATCH DATA LOADIND --
LOAD LABEL ps_stats_20150717 (
    DATA INFILE("hdfs://host:port/input/ps_stats_data")
    INTO TABLE ps_stats_tbl
);

-- Mini-BATCH DATA LOADING --
curl -u username,password -T ./input/ps_stats_data http://fe.host:port/api/db1/ps_stats_tbl/_load?label=ps_stats_20150717
```

资源隔离

- 问题
  - 多用户影响
  - 单用户多任务影响

- 解决
  - 线程级cgroup
  - 两级资源组织
  - 基于用户名的方式:
    username#high



```
mysql> show resource;
+--------+------------------+-------+
| User   | Resource type    | Value |
+--------+------------------+-------+
| root   | CPU_SHARE        | 1000  |
| zw     | CPU_SHARE        | 1000  |

mysql> show quota;
+------+-----------+---------+
| User | Group     | Quota   |
+------+-----------+---------+
| root | high      | 800     |
| root | low       | 100     |
| root | normal    | 400     |
+------+-----------+---------+
3 rows in set (0.01 sec)
```

```
|-- cgroup.clone_children
|-- cgroup.event_control
|-- cgroup.procs
|-- cpu.cfs_period_us
|-- cpu.cfs_quota_us
|-- cpu.shares
|-- cpu.stat
|-- cpuacct.stat
|-- cpuacct.usage
|-- cpuacct.usage_percpu
|-- notify_on_release
|-- root
|-- tasks
|-- test_user
`-- yiguolei
```

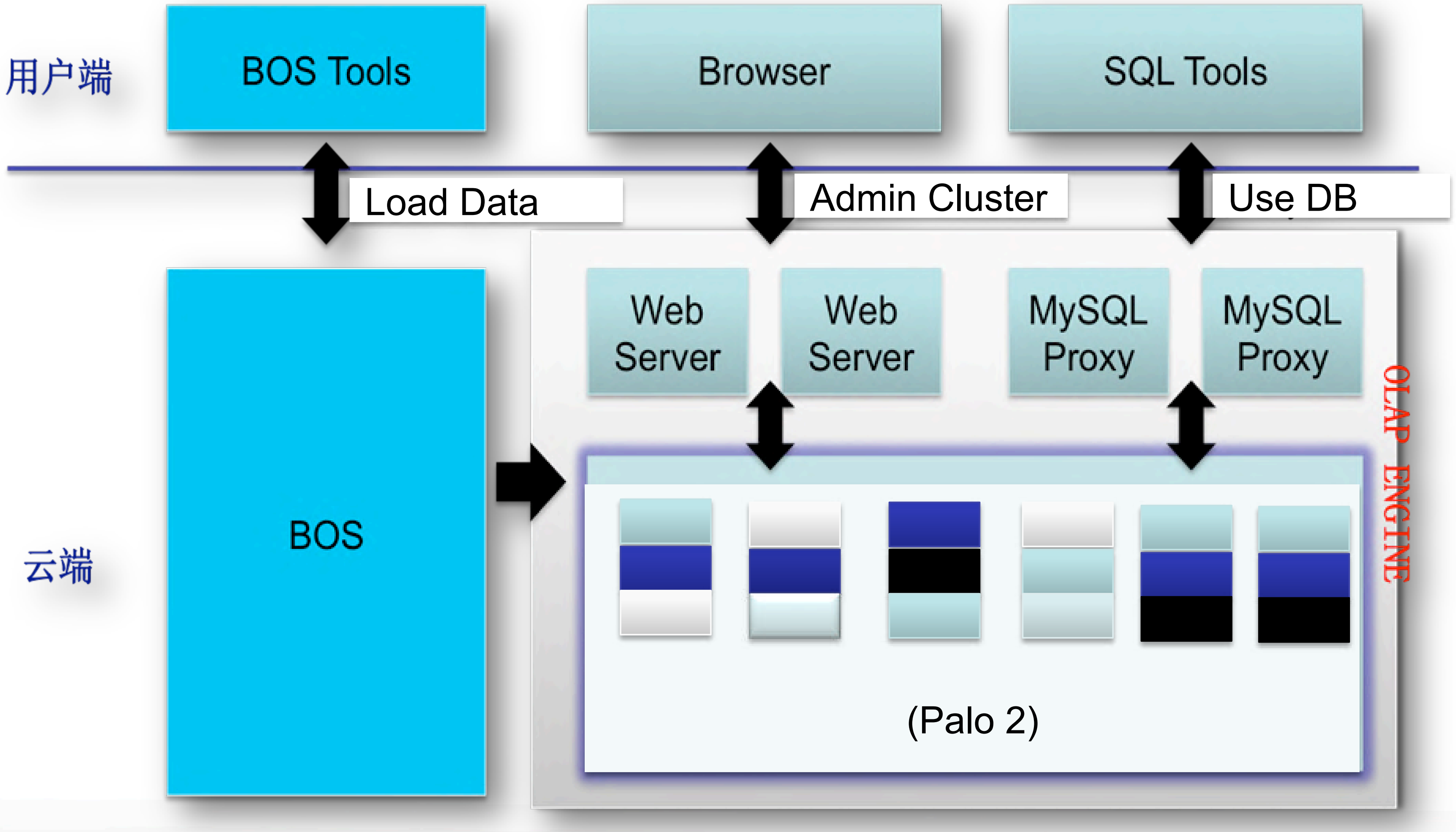
```
|-- cgroup.clone_children
|-- cgroup.event_control
|-- cgroup.procs
|-- cpu.cfs_period_us
|-- cpu.cfs_quota_us
|-- cpu.shares
|-- cpu.stat
|-- cpuacct.stat
|-- cpuacct.usage
|-- cpuacct.usage_percpu
|-- high
|   |-- cgroup.clone_children
|   |-- cgroup.event_control
|   |-- cgroup.procs
|   |-- cpu.cfs_period_us
|   |-- cpu.cfs_quota_us
|   |-- cpu.shares
|   |-- cpu.stat
|   |-- cpuacct.stat
|   |-- cpuacct.usage
|   |-- cpuacct.usage_percpu
|   |-- notify_on_release
|   `-- tasks
|-- low
|   |-- cgroup.clone_children
|   |-- cgroup.event_control
|   |-- cgroup.procs
|   |-- cpu.cfs_period_us
|   |-- cpu.cfs_quota_us
|   |-- cpu.shares
|   |-- cpu.stat
|   |-- cpuacct.stat
|   |-- cpuacct.usage
```

对外开放

Palo公有云化

- Palo云化
  - AWS redshift
  - on-demand provisioning
  - 百度公有云的需求
- Roadmap
  - 15.09: OLAP Engine Alpha
  - 15.12: OLAP Engine Beta
  - 16.06: OLAP Engine GA
- 当前正在使用客户
  - 20+外部客户试用

云化Palo实现鸟瞰


用户端
BOS Tools
Browser
SQL Tools
Load Data
Admin Cluster
Use DB
云端
BOS
Web Server
Web Server
MySQL Proxy
MySQL Proxy
(Palo 2)
OLAP ENGINE

SDCC
中国软件开发者大会
SOFTWARE DEVELOPER CONFERENCE CHINA

CSDN
全球最大中文IT社区

Thanks & QA

sdcc.csdn.net