

# Hive高级编程

天照



大数据论坛 更多学习资料

<http://china-hadoop.com>

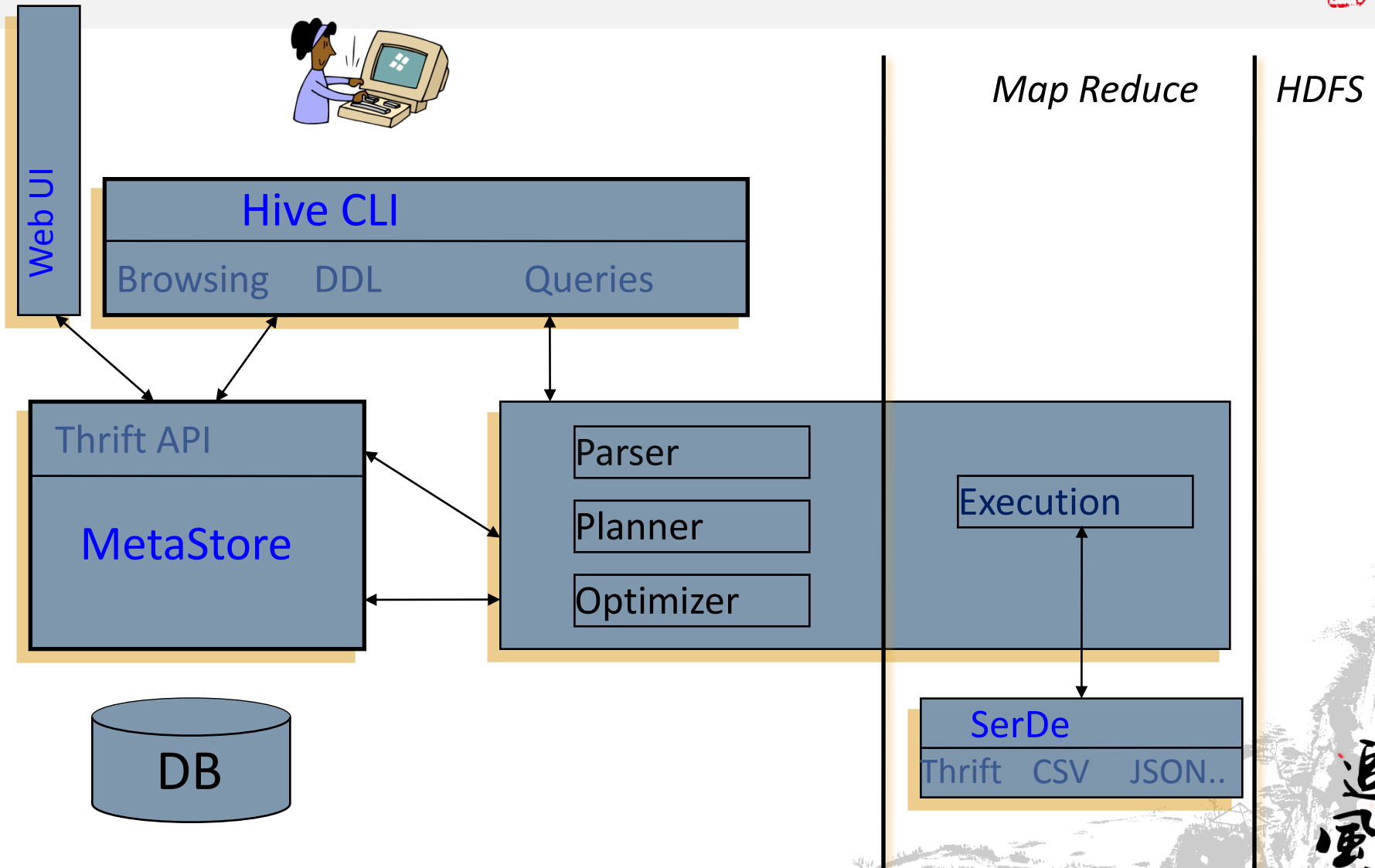
追風堂



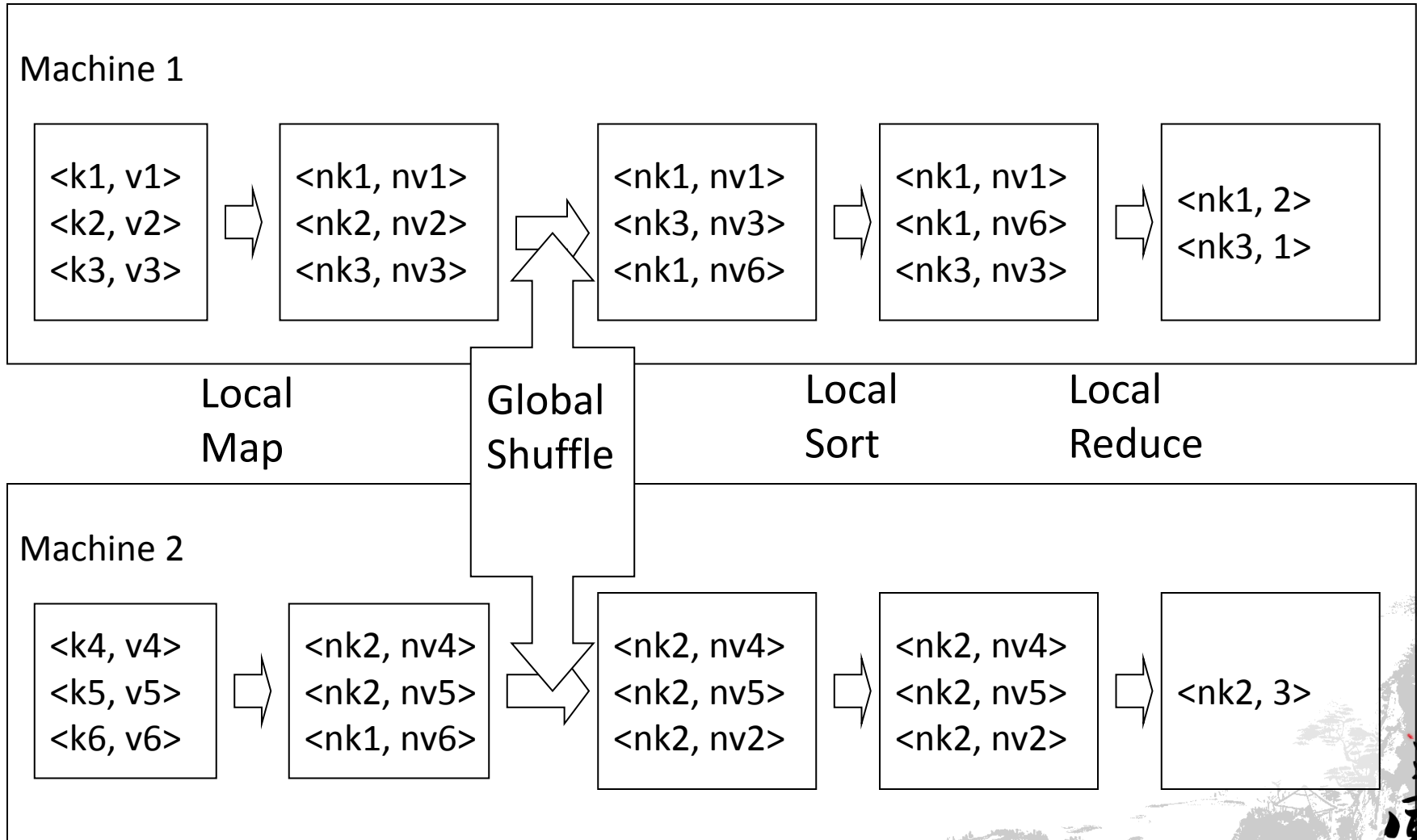
- Hive Components
- MapReduce
- Hive QL
- Hive 优化
- SQL优化



# HIVE: Components



# (Simplified) Map Reduce Review



# Hive QL – Join



page\_view

pageid	userid	time
1	111	9:08:01
2	111	9:08:13
1	222	9:08:14

X

user

userid	age	gender
111	25	female
222	32	male

=

pv\_users

pageid	age
1	25
2	25
1	32

- SQL:

```
INSERT INTO TABLE pv_users
```

```
SELECT pv.pageid, u.age
```

```
FROM page_view pv JOIN user u ON (pv.userid = u.userid);
```



# Hive QL – Join in Map Reduce



page\_view

pageid	userid	time
1	111	9:08:01
2	111	9:08:13
1	222	9:08:14

user

userid	age	gender
111	25	female
222	32	male

Map

key	value
111	<1,1>
111	<1,2>
222	<1,1>

Shuffle  
Sort

key	value
111	<1,1>
111	<1,2>
111	<2,25>

Reduce

key	value
222	<1,1>
222	<2,32>

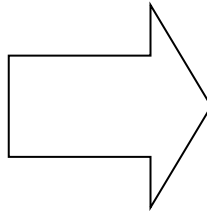


# Hive QL – Group By



pv\_users

pageid	age
1	25
2	25
1	32
2	25

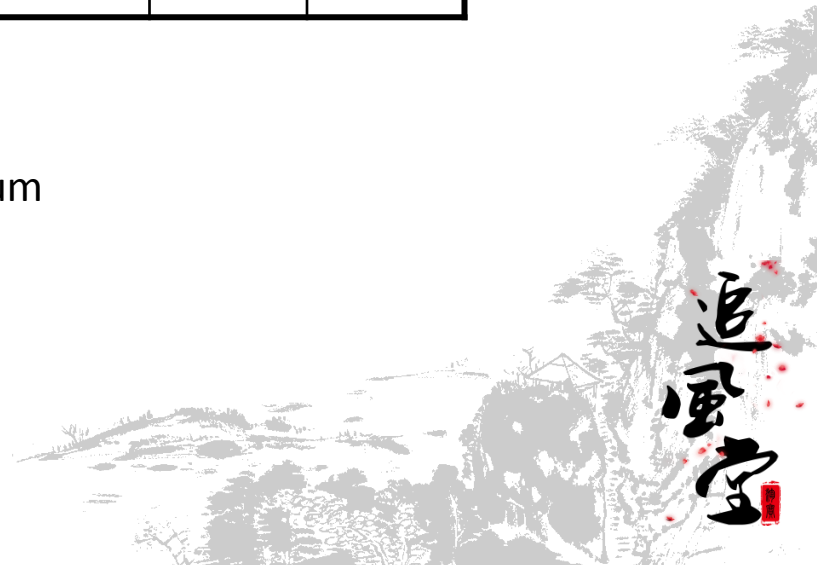


pageid\_age\_sum

pageid	age	Count
1	25	1
2	25	2
1	32	1

- **SQL:**

- INSERT INTO TABLE pageid\_age\_sum
- SELECT pageid, age, count(1)
- FROM pv\_users
- GROUP BY pageid, age;

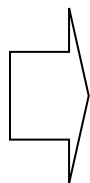


# Hive QL – Group By in Map Reduce



pv\_users

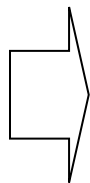
pageid	age
1	25
2	25



key	value
<1,2 5>	1
<2,2 5>	1

Map

pageid	age
1	32
2	25



key	value
<1,3 2>	1
<2,2 5>	1

Shuffle  
Sort

key	value
<1,2 5>	1
<1,3 2>	1

Reduce



key	value
<2,2 5>	1
<2,2 5>	1



pageid	age

pageid	age

追風堂

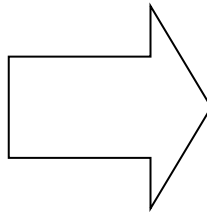


# Hive QL – Group By with Distinct



page\_view

pageid	userid	time
1	111	9:08:01
2	111	9:08:13
1	222	9:08:14
2	111	9:08:20



result

pageid	count_distinct_userid
1	2
2	1

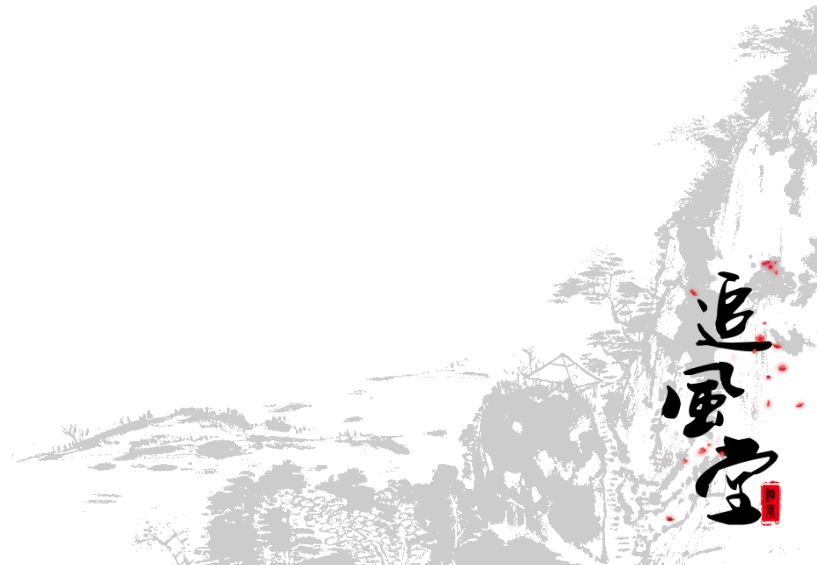
- **SQL**

- SELECT pageid, COUNT(DISTINCT userid)
- FROM page\_view GROUP BY pageid

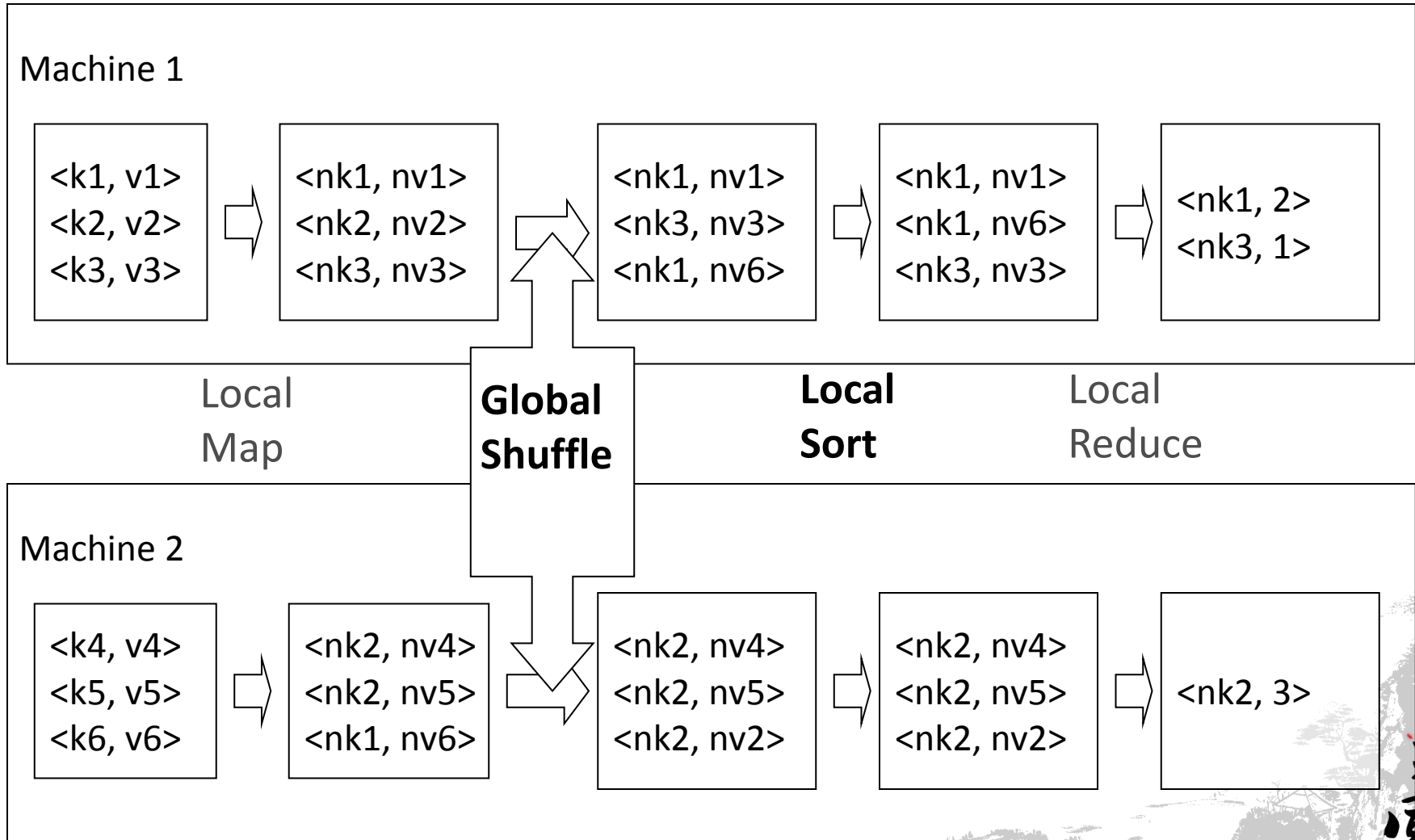


# Hive Optimizations

*Efficient execution of SQL on Map Reduce*

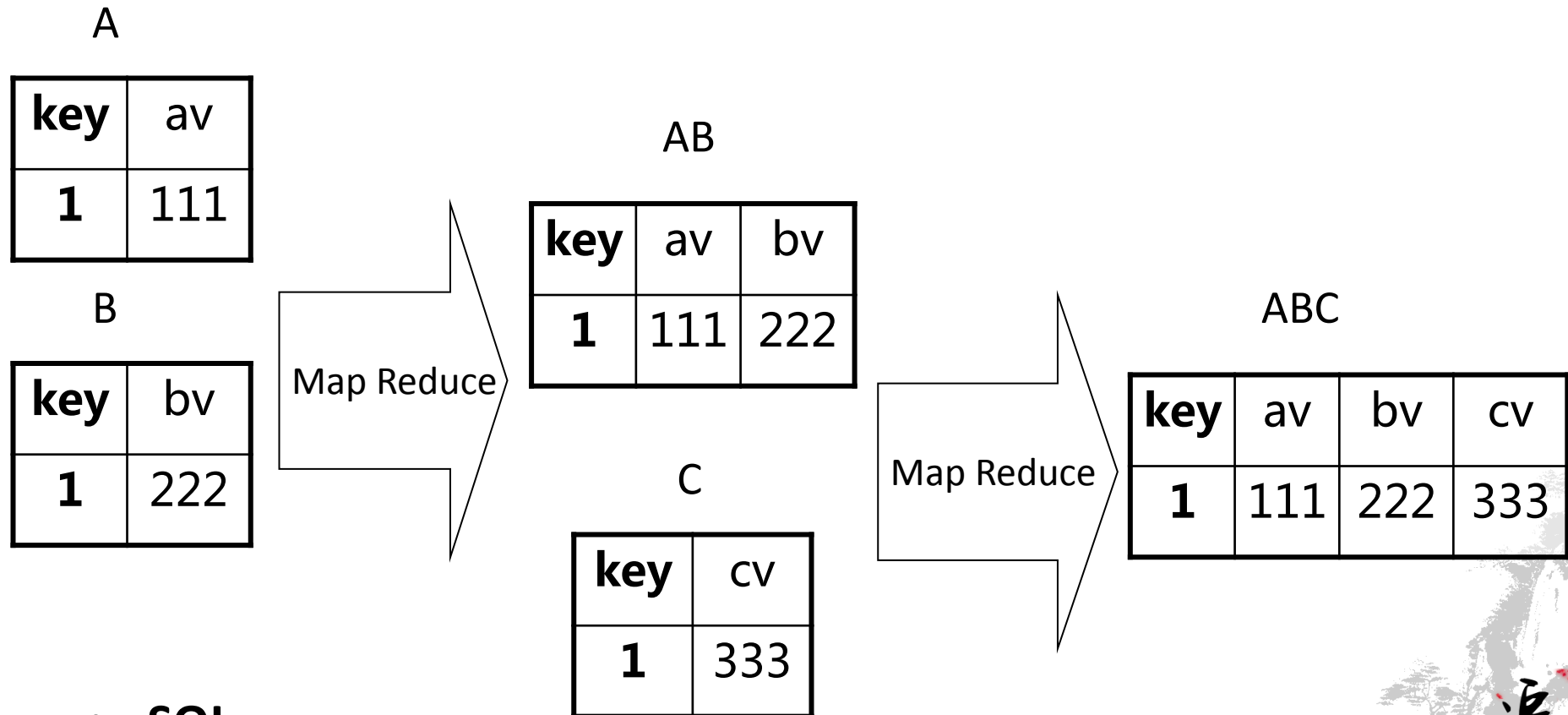


# (Simplified) Map Reduce Revisit



# Hive Optimizations

## – Merge Sequential Map Reduce Jobs



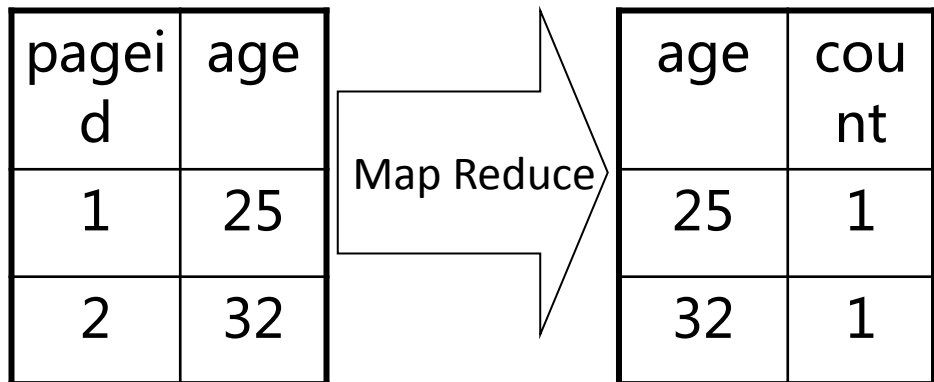
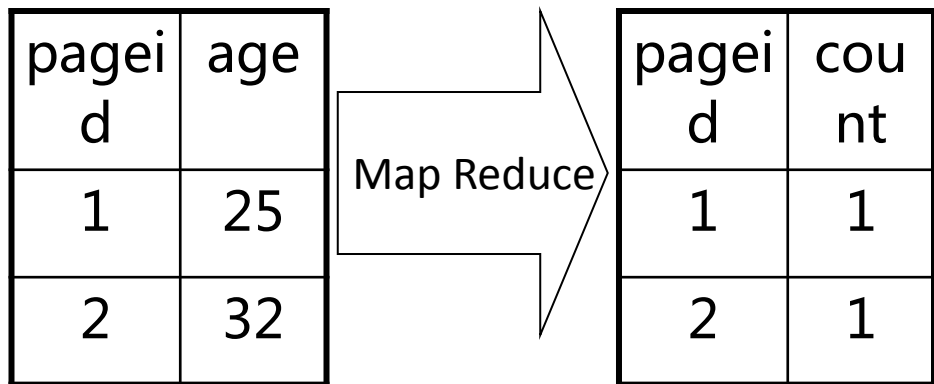
- **SQL:**

- FROM (a join b on a.key = b.key) join c on a.key = c.key SELECT ...



# Hive Optimizations

## – Share Common Read Operations



- **Extended SQL**

- **FROM** pv\_users
- **INSERT INTO TABLE** pv\_pageid\_sum
- **SELECT** pageid, count(1)
- **GROUP BY** pageid
- **INSERT INTO TABLE** pv\_age\_sum
- **SELECT** age, count(1)
- **GROUP BY** age;



# Hive Optimizations

## –Map Join



- **Map Joins**

- User specified small tables stored in hash tables on the mapper backed by jdbm
- No reducer needed

```
INSERT INTO TABLE pv_users
SELECT /*+ MAPJOIN(pv) */ pv.pageid, u.age
FROM page_view pv JOIN user u
ON (pv.userid = u.userid);
```



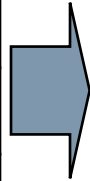
追風堂

# Hive QL – Map Join



page\_view

pageid	userid	time
1	111	9:08:01
2	111	9:08:13
1	222	9:08:14



Hash table

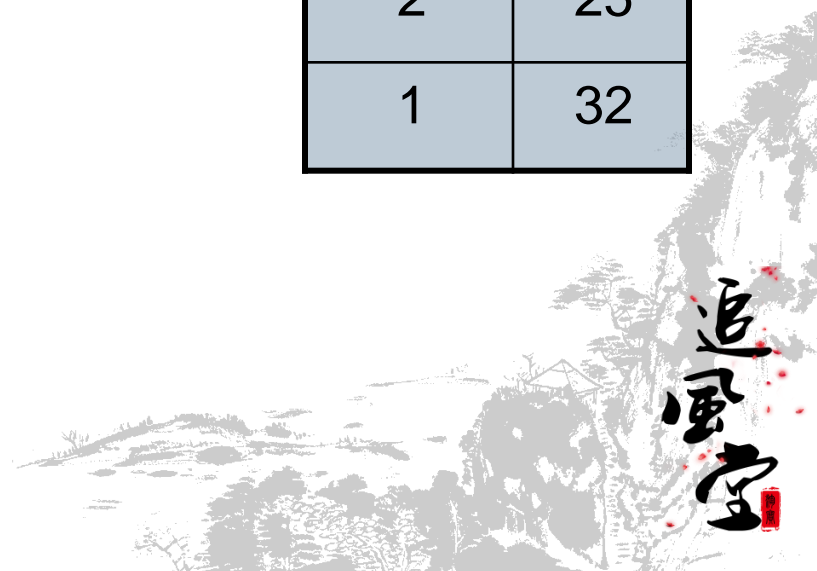
key	value
111	<1,2>
222	<2>

user

userid	age	gender
111	25	female
222	32	male

pv\_users

Pageid	age
1	25
2	25
1	32



# Group by Optimizations



- **Map side partial aggregations**
  - Hash-based aggregates
  - Serialized key/values in hash tables
  - 90% speed improvement on Query
    - `SELECT count(1) FROM t;`



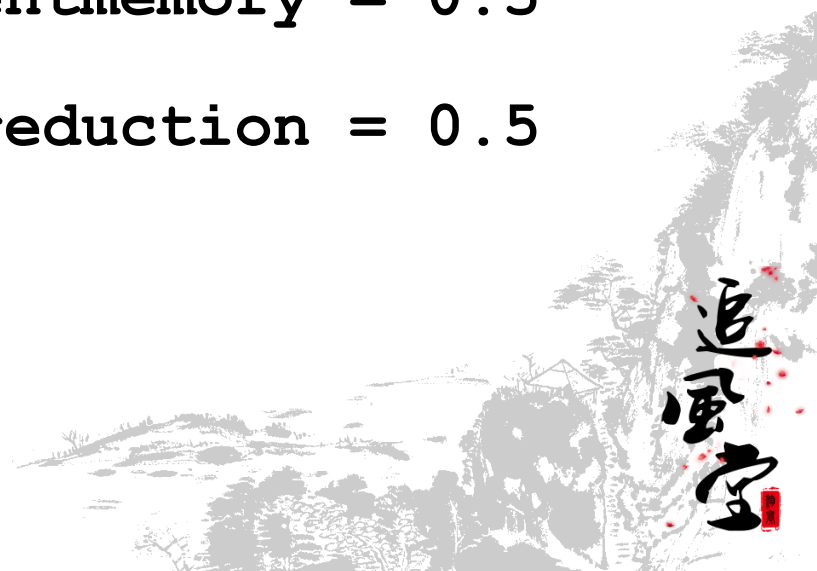
追風堂



# Parameters



- `hive.map.aggr = true`
- `hive.groupby.skewindata = false`
- `hive.groupby.mapaggr.checkinterval = 100000`
- `hive.map.aggr.hash.percentmemory = 0.5`
- `hive.map.aggr.hash.min.reduction = 0.5`



追風堂

# Multi GroupBy



```
FROM pv_users
```

```
INSERT OVERWRITE TABLE pv_gender_sum
```

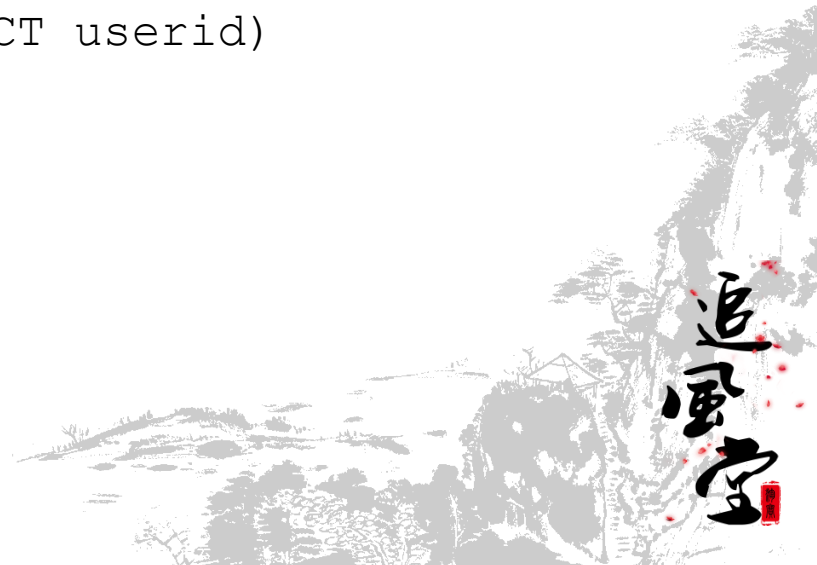
```
SELECT gender, count(DISTINCT userid),  
       count(userid)
```

```
GROUP BY gender
```

```
INSERT OVERWRITE TABLE pv_age_sum
```

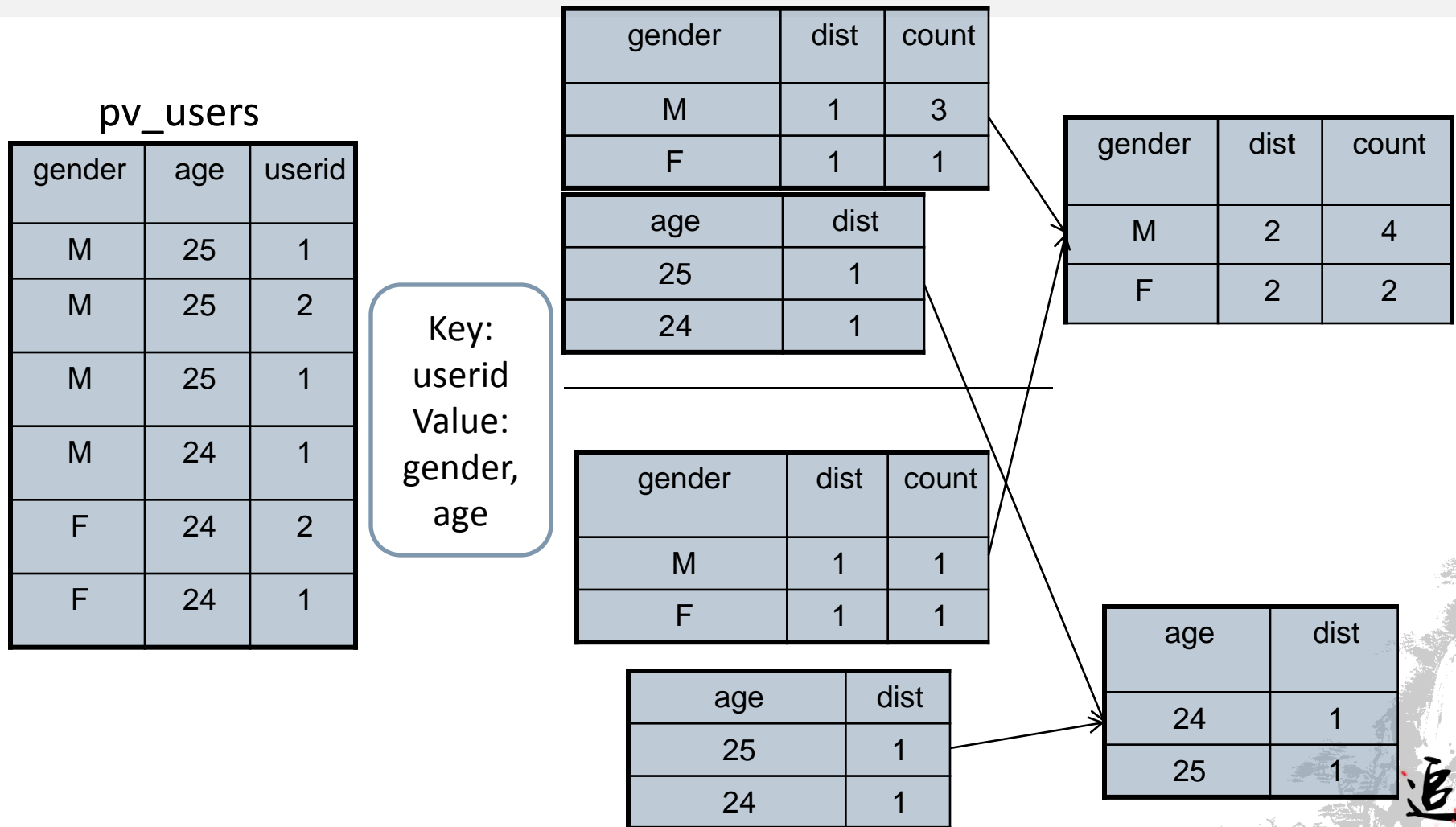
```
SELECT age, count(DISTINCT userid)
```

```
GROUP BY age
```



追風堂

# Hive QL – Group By in Map Reduce



# Load balancing for data skew

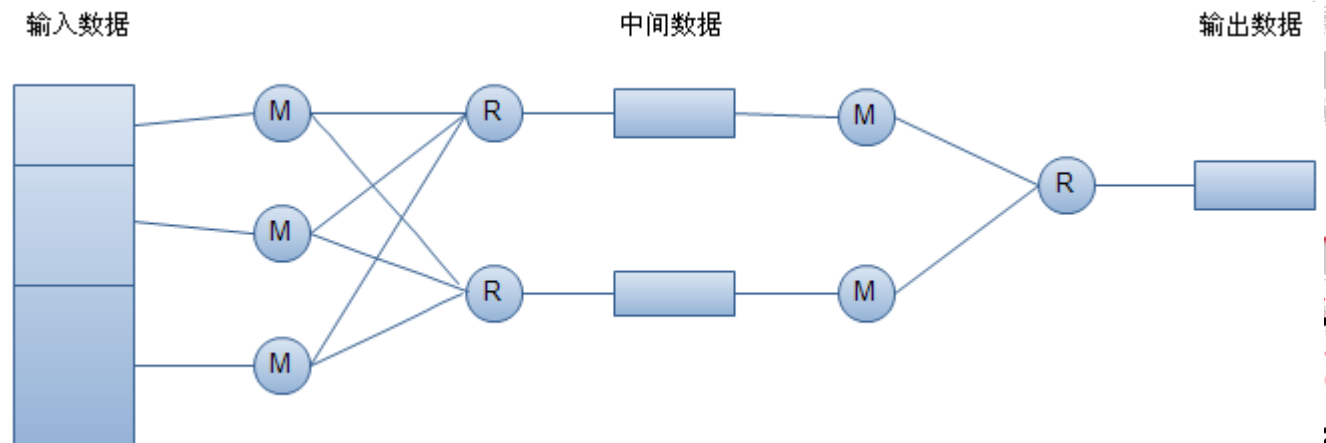
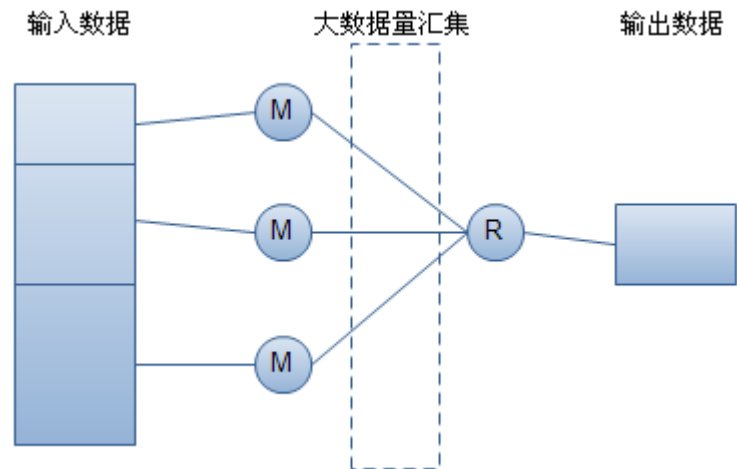


- **GroupBy数据倾斜**

- skewindata优化

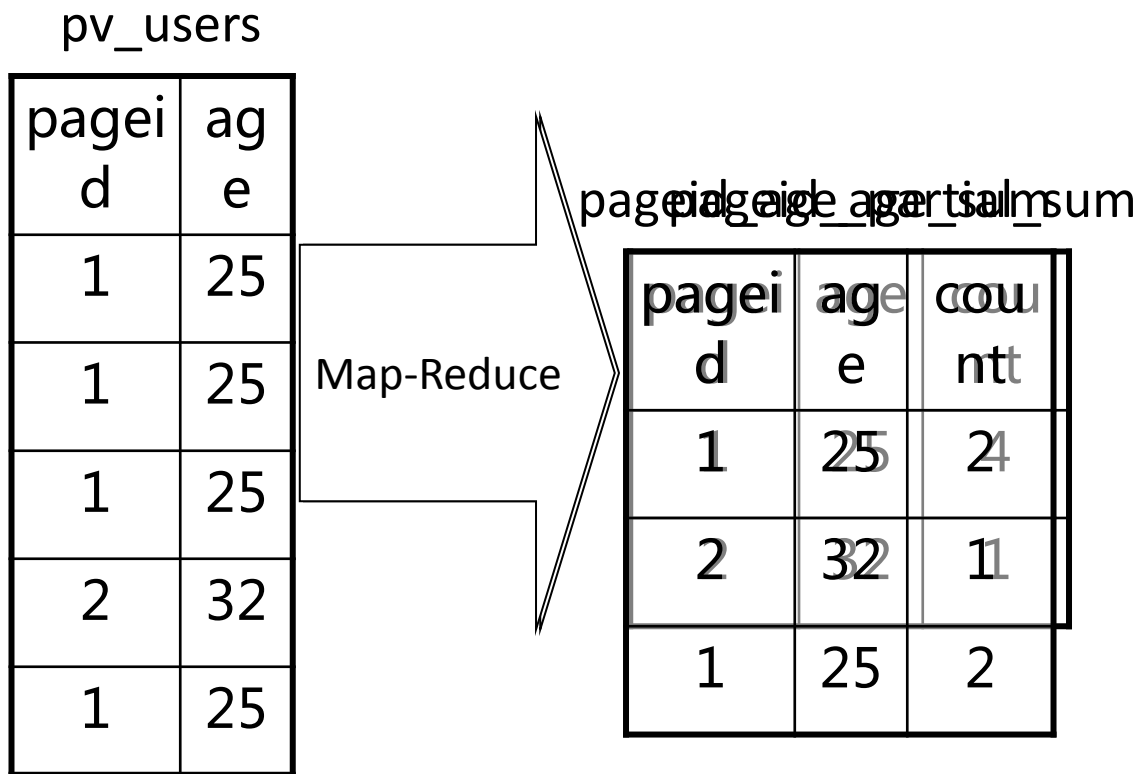
- 用法

- set hive.groupby.skewindata=true



# Hive Optimizations

## – Load Balance Problem





- 数据倾斜
- Join顺序
- Map only





- **数据倾斜**

- 倾斜的原因？

- group by/distinct
    - join





- 内存优化

- 驱动表

- 使用大表做驱动表，避免内存溢出
    - Join中最右边的表是驱动表
    - MapJoin无视Join顺序，使用大表做驱动表
    - STREAMTABLE







- **Map only**

- 特征

- 没有Join、GroupBy、Order by、Sortby等，导致无Reduce
    - 每个Map有一个输出文件，输入数据量大，Map数很多导致输出文件很多

- 缺点

- 依赖此job输出的下一个job，map数很大
    - Fetch 结果很慢





- Hive Components
- MapReduce
- Hive QL
- Hive 优化
- SQL优化



谢谢！



追風堂

