

MaxCompute 的NewSQL演进之路

阿里巴巴计算平台架构师 林伟

weilin.lw@alibaba-inc.com

1

背景

2

NewSQL

3

关键技术

4

总结

SQL

NewSQL

NoSQL

关系型数据库

强语义

强结构

横向扩展性差

不灵活

容错能力弱

大数据能力差

非关系型数据库

弱语义

非结构，半结构和结构

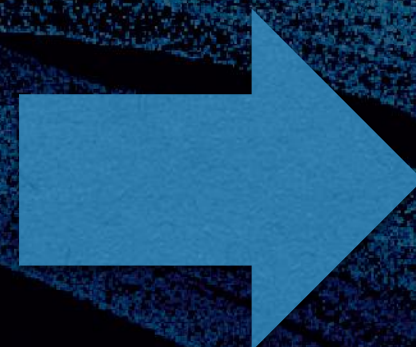
横向扩展性强

强大UDF

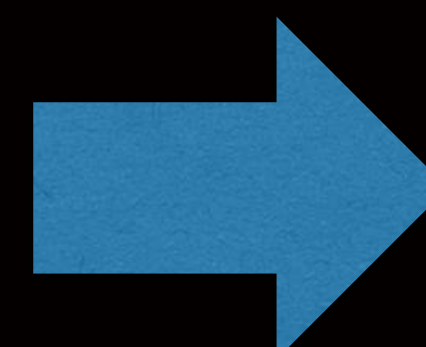
容错能力强

大数据能力强

Traditional
DB



MapReduce



Tez
Spark

NewSQL

回归**关系型**：程序员仅仅关注要**做什么**

强大**系统优化**能力：通过强大**优化器**，整合多个功能，从而能够使得系统**自适应**生产**高效**物理执行计划

NonSQL的支持：非结构，强大的UDF集合，分布式支持

How



What

Why NewSQL

- 程序员不可能及时感受数据的变化，环境变化
- 计算越来越复杂，程序员不可能很快的分析出全局最优的执行计划
- 计算需要能够分享知识，缺乏高层次强语义的语言阻碍这种分享
- 共享的资源环境，单个程序员缺乏系统全局观

NewSQL

关系型查询语言



灵活性

任意物理执行计划
非结构，半结构，结构

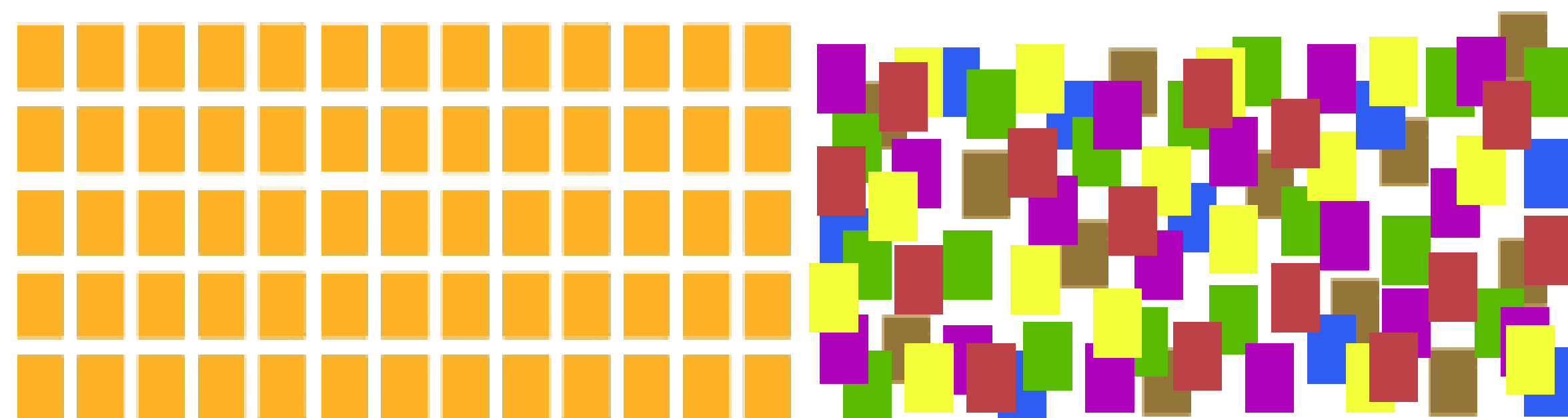
高性能，高智能，
自适应

- Dryad→Scope(微软)
- Spark→SparkSQL(DataBricks)
- MapReduce→Hive→Hive2.0
- MapReduce→Spanner (Google)
- MaxCompute1.0→
MaxCompute2.0

关键技术

非结构，半结构和结构化数据

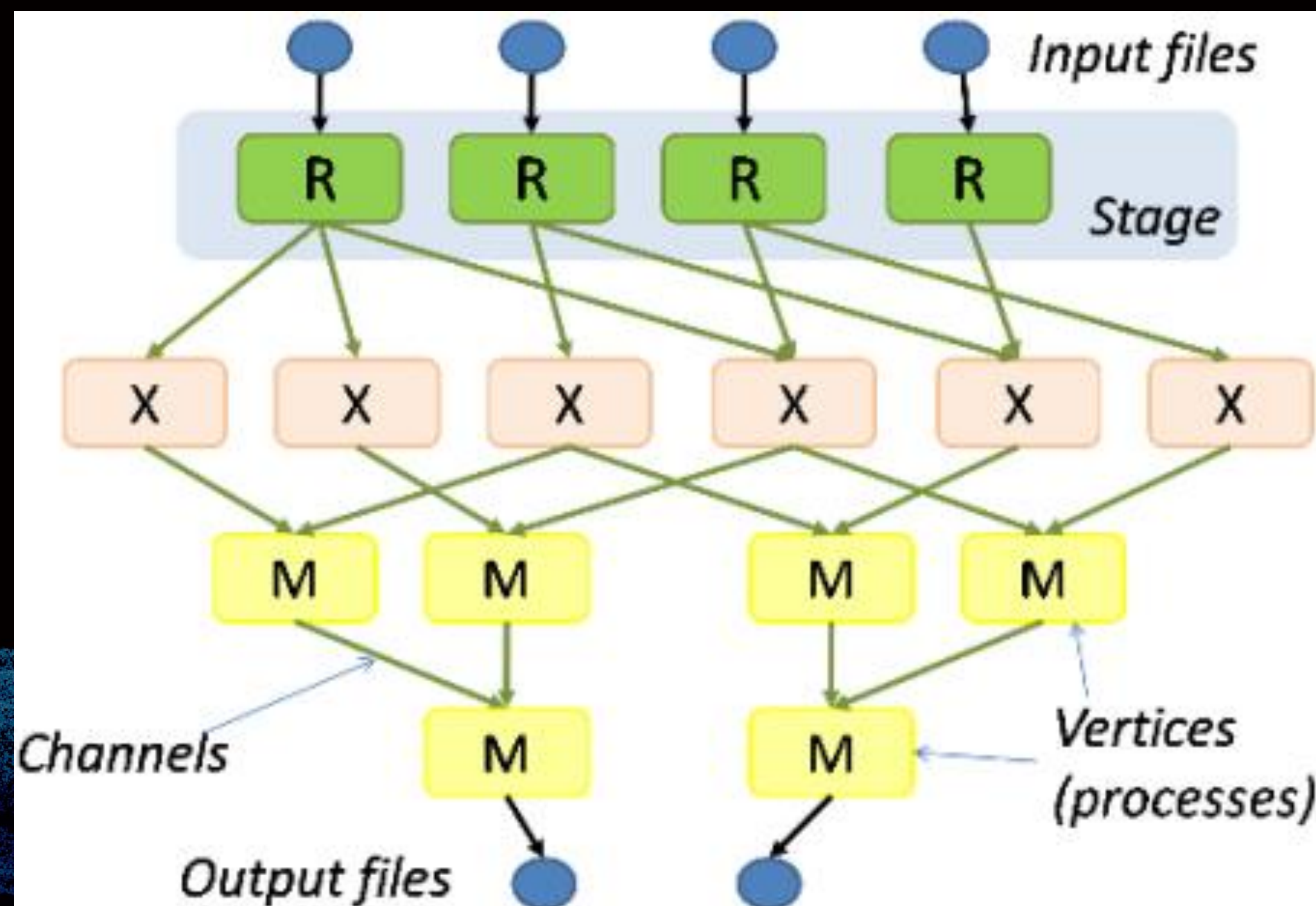
"80% of business-relevant information originates in unstructured form, primarily text."



Structured Data vs. **Unstructured Data**

- 用户提供
Serialize/Deserialize函数动态进行非结构到结构化转换
- 支持用户自定义类型
- 用户自定义分区从而能够有效连接上下流

强大的DAG执行图表示



- 突破MapReduce的束缚
- 进行循环迭代展开为DAG
- 非对称图表达，从而能够支持复杂的物理执行计划

完整的用户自定义函数体系

- 完整UDF集合：必要的能使得我们关系型退化为函数型语言，能够构造任意的DAG执行计划
 - Serialize/Deserialize
 - 多路Join函数
 - 聚合处理函数
 - Processor
 - 完备分区函数：Hash/Range/Direct Hash
 - ...

强大的优化器

- 存储过程的支持 (单一语句→成千上万存储过程)
 - 能够写出更加复杂大的查询存储过程，从而使得逻辑执行计划非常庞大，优化空间更大，需要更先进的优化器

```
lookbackdays := 100;  
sampleRate := 0.5;  
runDate := CAST('2015-01-01 00:00:00' AS DATETIME);  
debugMode := true;  
review_info :=  
FROM ob_task WHERE current_step = 'finished' AND run_time > 0  
SELECT  
  id,  
  user_id,  
  review_id,  
  run_time/60.0 as run_time;  
  
checkUsers BOOLEAN := false;  
NEW TABLE (key BIGINT, value STRING);  
IF @checkUsers  
  @kv := FROM @review_info SELECT id AS key, user_id AS value;  
ELSE  
  @kv := FROM @review_info SELECT id AS key, review_id AS value;  
  
INSERT OVERWRITE TABLE dw_dev_kv AS SELECT * FROM @kv;
```

Rule-
Based Opt

Cost-
Based Opt
volcano

Cost-
Based Opt
cascading

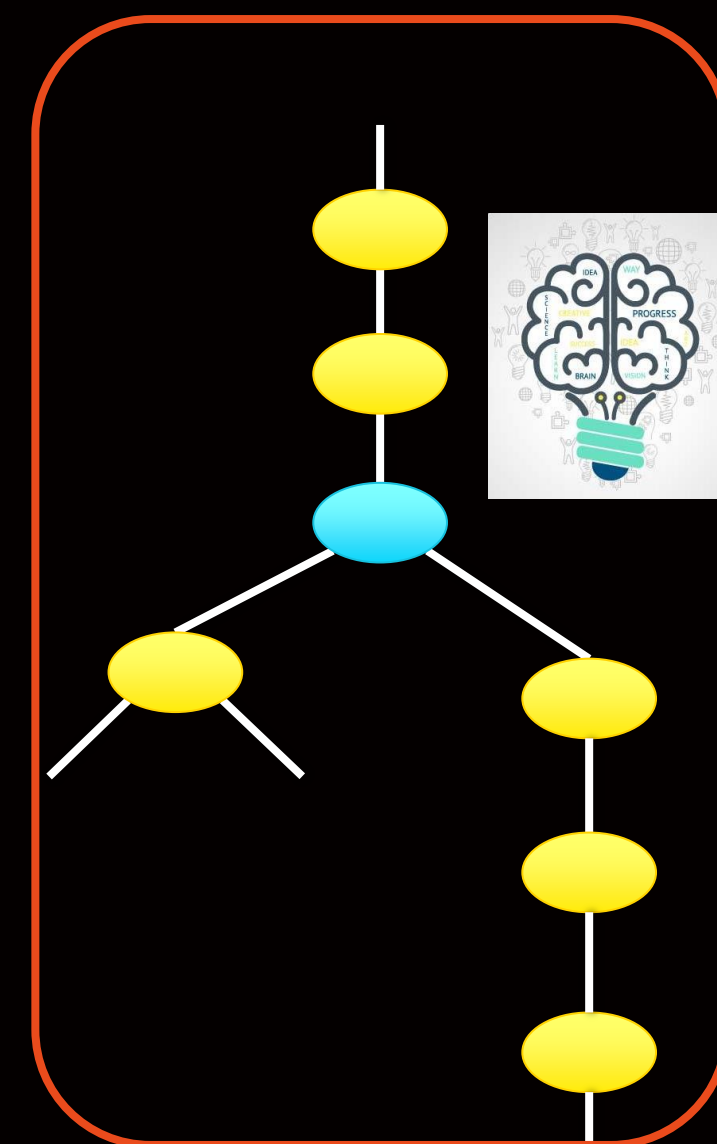
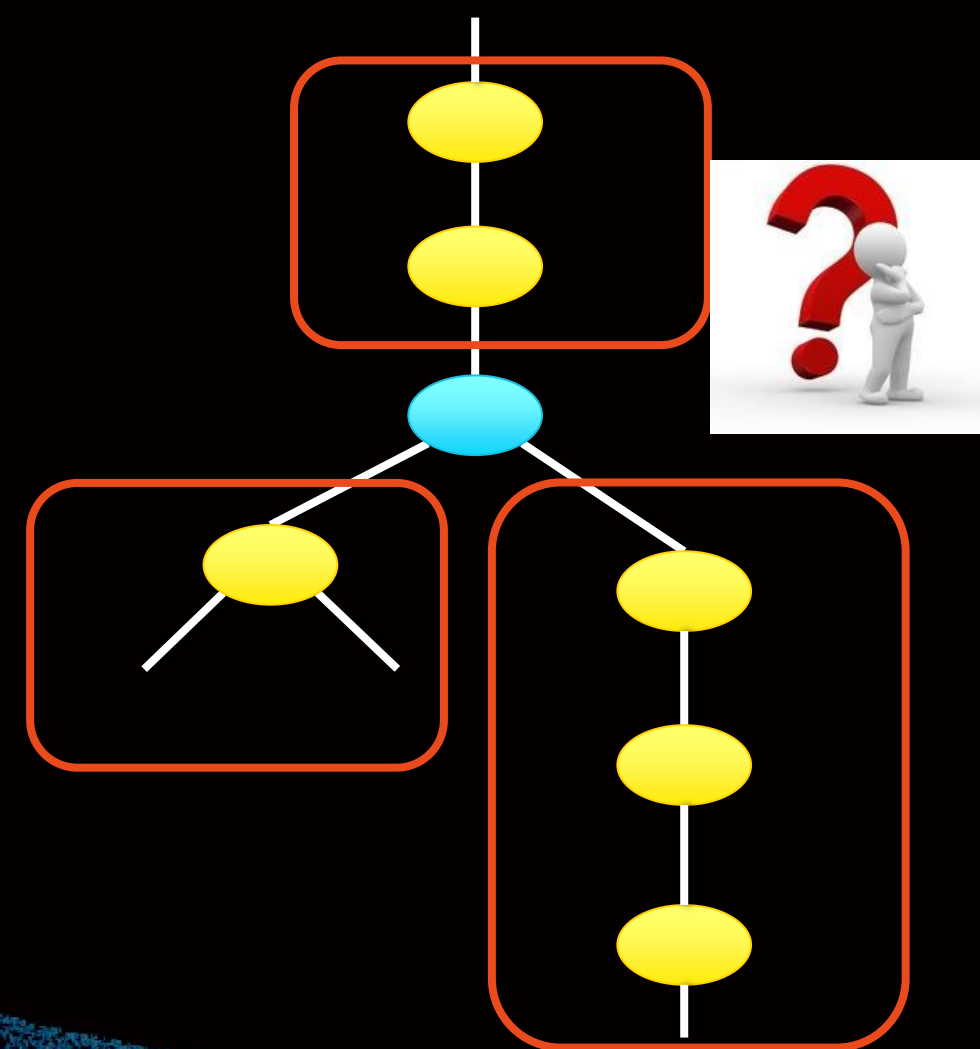
强大的优化器

- 分布式
- Non-SQL的场景下的众多UDF扩展



优化器与用户自定义函数（UDF）

- 优化**碎片化**
（性能低）
- **不感知**UDF
输出特性，
从而产生低
效物理执行
计划



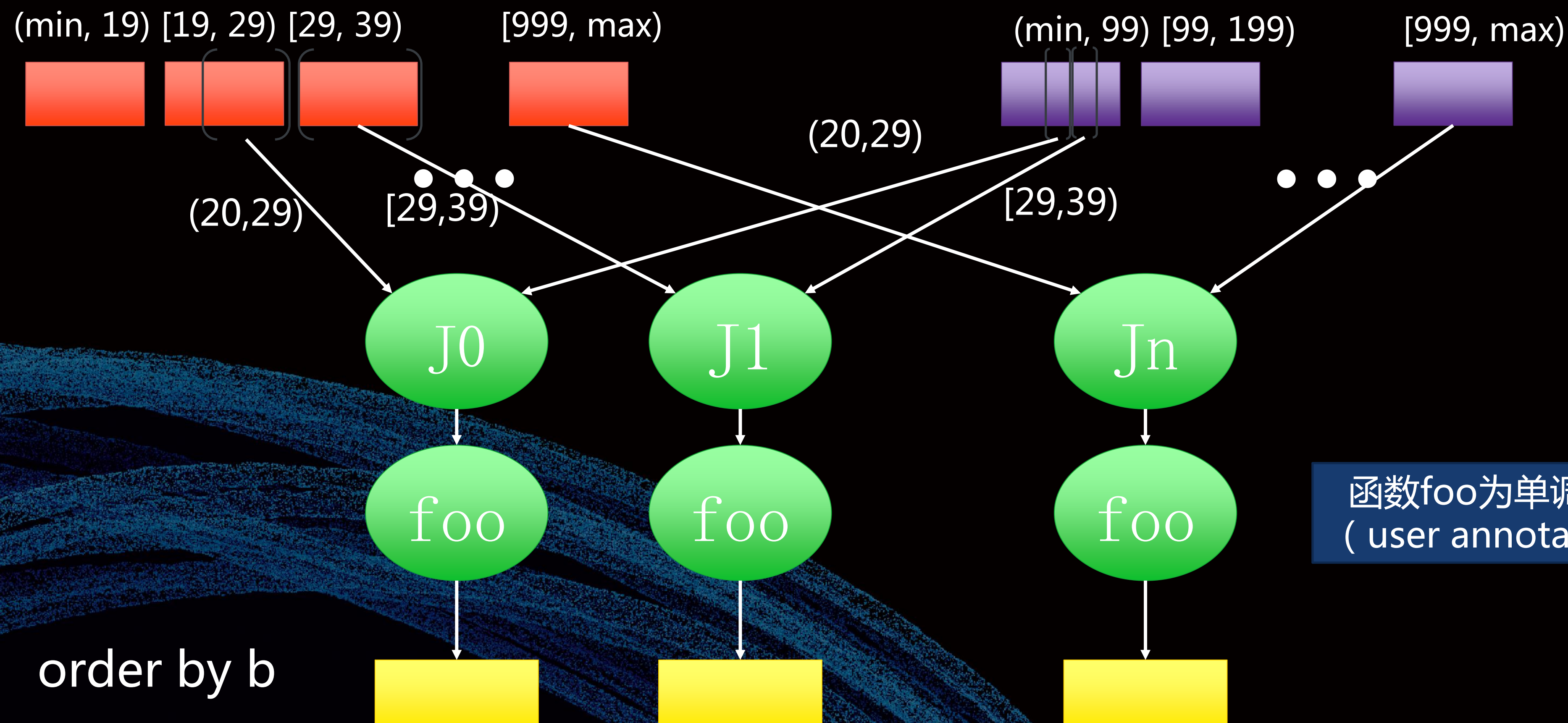
- **全局**优化
- 有效和用户
交互**理解**
UDF的特性



包含UDF

一个实际例子

```
select t1.a, foo(t1.a) AS b, *  
from t1 join t2 on t1.a == t2.a  
where t1.a > 20  
order by b;
```



UDF的特性

- Row-wise ? 单调函数 ?
- 某些column不变 (pass through) ?
- 保持分片 ? 保持排序 ?
- Selectivity , data distribution of output等等

运行时优化

- 大量的NonSQL的用户自定义函数
- 分布式场景中各种动态环境
 - 分配worker的拓扑结构
 - Failure Region的分布
- 需要运行时优化
 - 选择分区数目，边界
 - 选择Join方式
 - 高效的Datashuffle方式

总结

强大的系统优化

NoSQL

OldSQL



NewSQL

How



What

高易用

高可解释

高性能

大规模

高自适应

THANK YOU

Scan QR Code



关注MaxCompute产品社区
Community



了解MaxCompute产品详情
Product Details



加入MaxCompute钉群咨询
Join DingTalk Group



诚聘MaxCompute英才
We are hiring!