2019阿里云峰会·上海
开发者大会
DEVELOPER CONFERENCE_

Apache Spark最新技术进展和3.0+展望

李呈祥（司麟）

阿里巴巴高级技术专家
计算平台事业部

Agenda

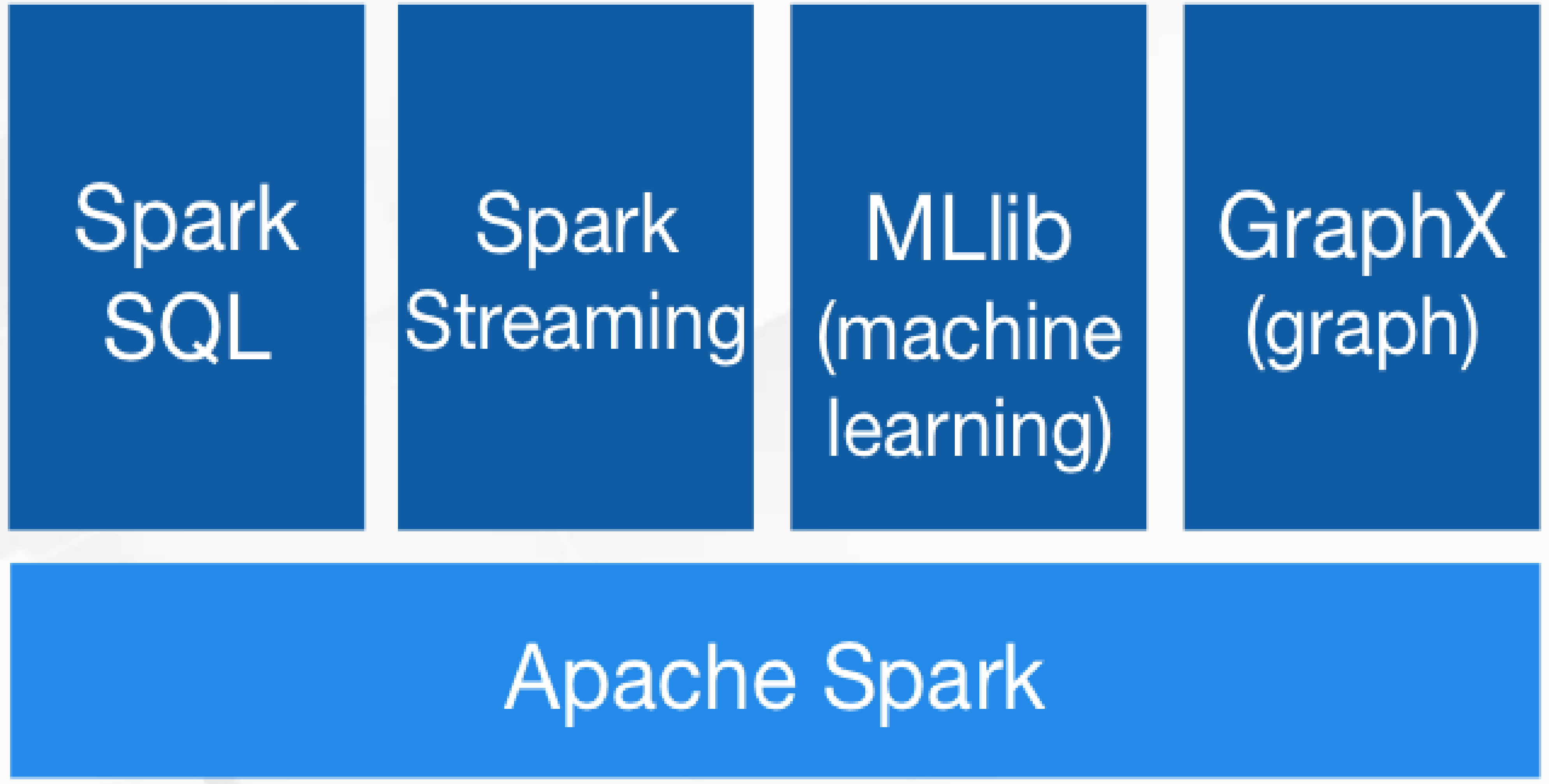
Data Warehouse Enhancement

Spark on Cloud

Spark + AI

3.0

A Unified Analytics Engine for Large-scale Data Processing



Java
SQL
hadoop HDFS
APACHE HBASE
amazon web services S3
Parquet
DELTA LAKE
{JSON} JavaScript Object Notation
hadoop YARN

Tungsten

Catalyst

Easy-to-use API

Rich Ecosystem Support

Efficient Engine

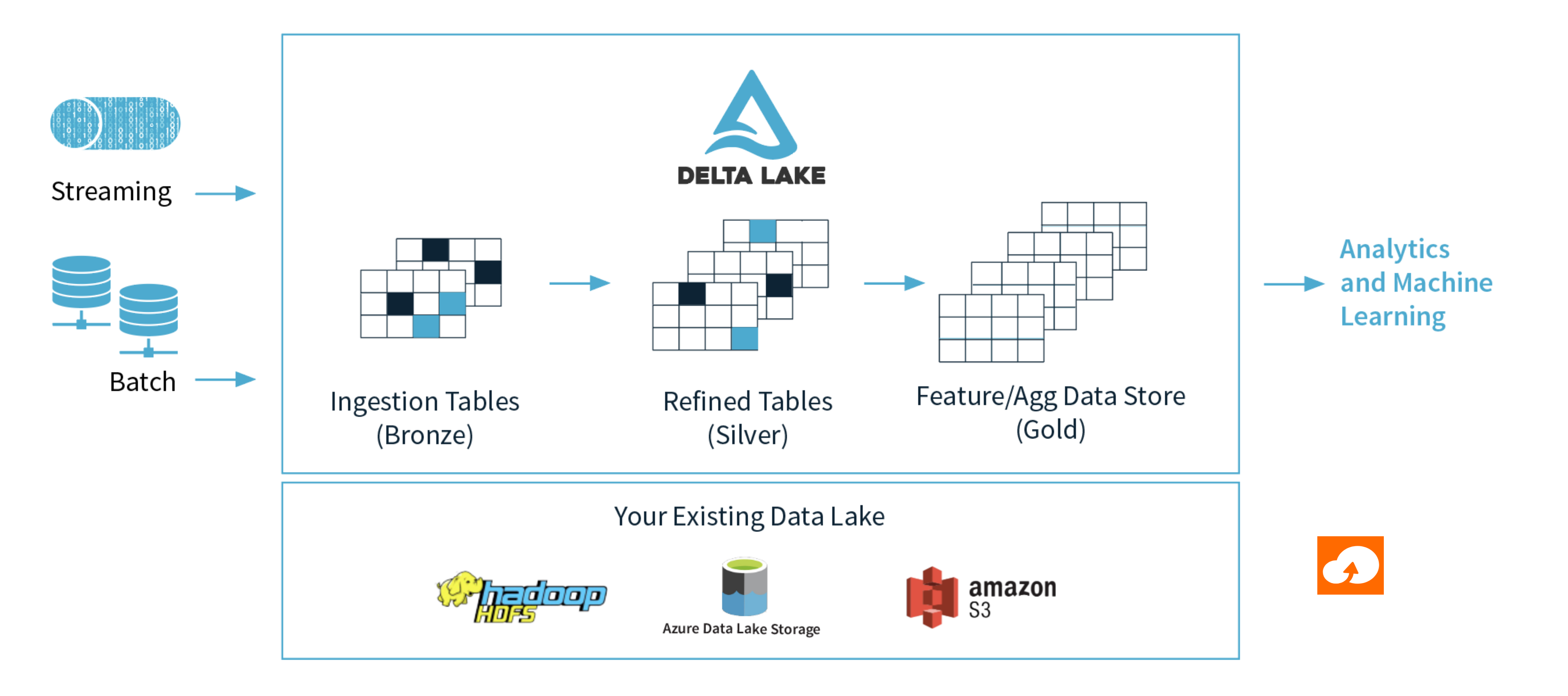
Data Warehouse Enhancement

Delta Lake

- ACID Transactions
- Scalable Metadata Handling
- Time Travel (data versioning)
- Open Format
- Unified Batch and Streaming Source and Sink
- Schema Enforcement

Coming soon:

- Audit History
- Full DML Support
- Expectations



Data Source V2

- Unified API for batch and streaming
- Flexible API for high performance implementation
- Flexible API for metadata management
- Target 3.0

Runtime Optimization

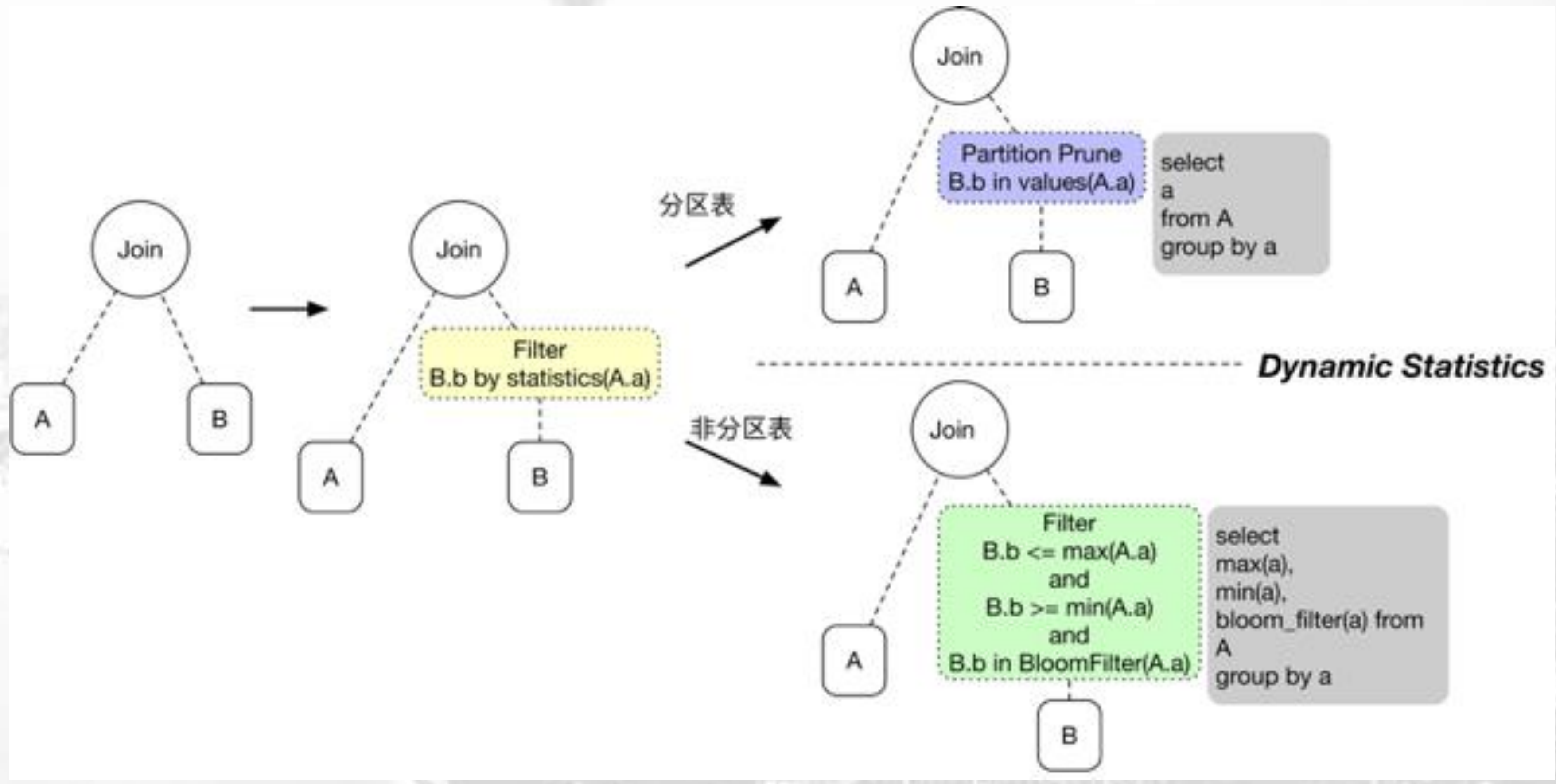
Adaptive Execution

Dynamic optimize the execution plan at runtime based on the statistic of previous stage.

- Self tuning the number of reducers
- Adaptive join strategy
- Automatic skew join handling

EMR Runtime Filter

- Filter big table with runtime statistic of join key.
- Support both partitioned table and normal table.



```
select
a
from A
group by a
```

```
select
max(a),
min(a),
bloom_filter(a) from
A
group by a
```

EMR Spark Relational Cache

User may analyze data in certain access pattern

- Regularly join 2 tables?
- Regularly aggregate by certain fields?
- Regularly filter by certain fields?
- ……

Data Organization:

- partition, bucket, sort
- file index, zorder

Data pre-computation:

- pre-filter
- denormalization
- pre-aggregation
- ……

Make data adaptive to compute, so spark compute faster.

EMR Spark Relational Cache

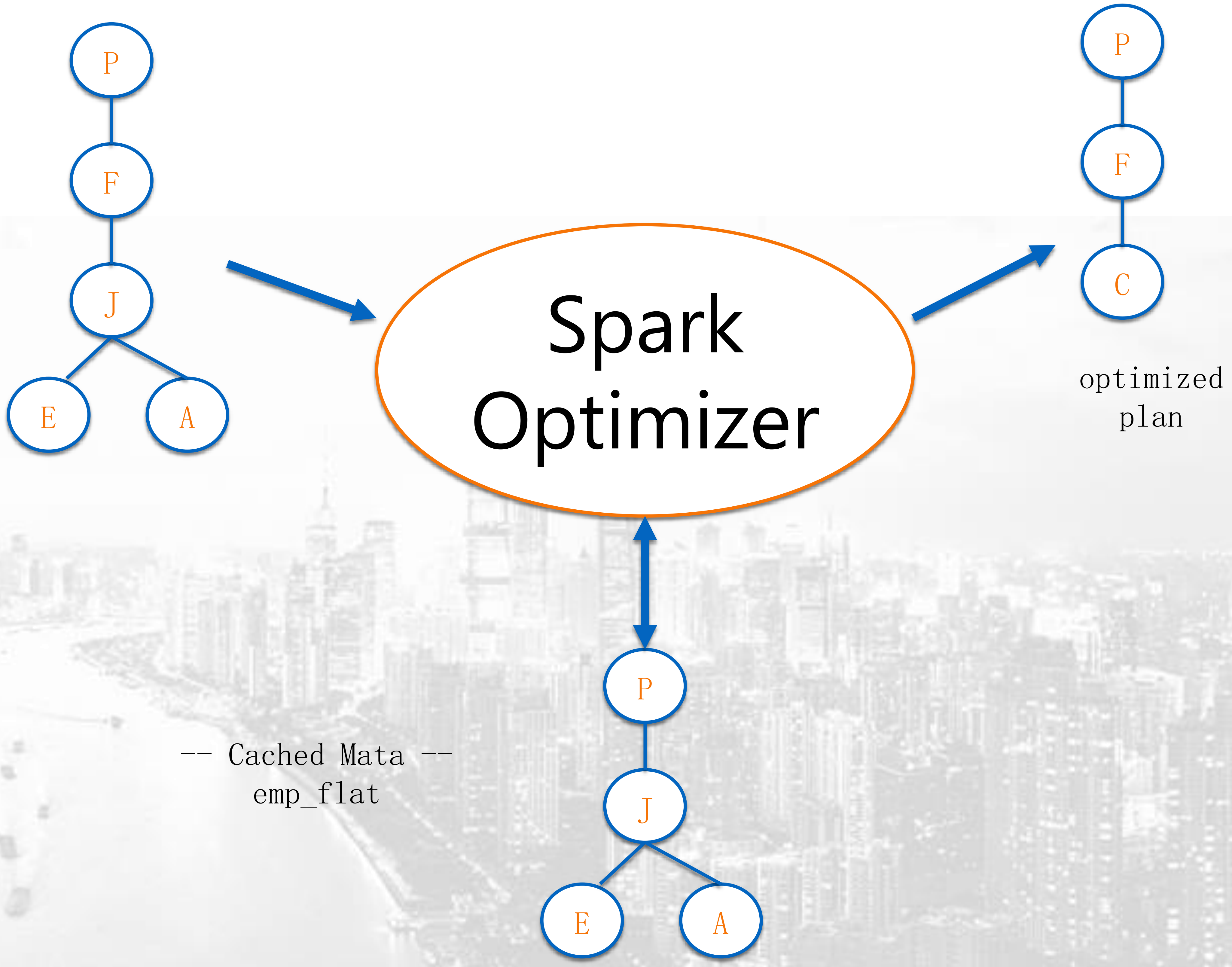
Easy to build and maintain

```
CREATE VIEW emp_flat AS
SELECT * FROM employee, address
WHERE e_addrId = a_addrId;

CACHE TABLE emp_flat
USING parquet
PARTITIONED BY (e_ob_date)
```

```
-- User Query --
SELECT * FROM
employee,
address WHERE
e_addrId =
a_addrId and
a_cityName
= 'ShangHai'
```

Transparent to user



Spark on Cloud

Storage and Computing Disaggregation

APACHE Spark

Aliyun OSS

Why disaggregate storage and computing:

- Pay as you go.
- Scale independently of each other.
- More reliable storage.

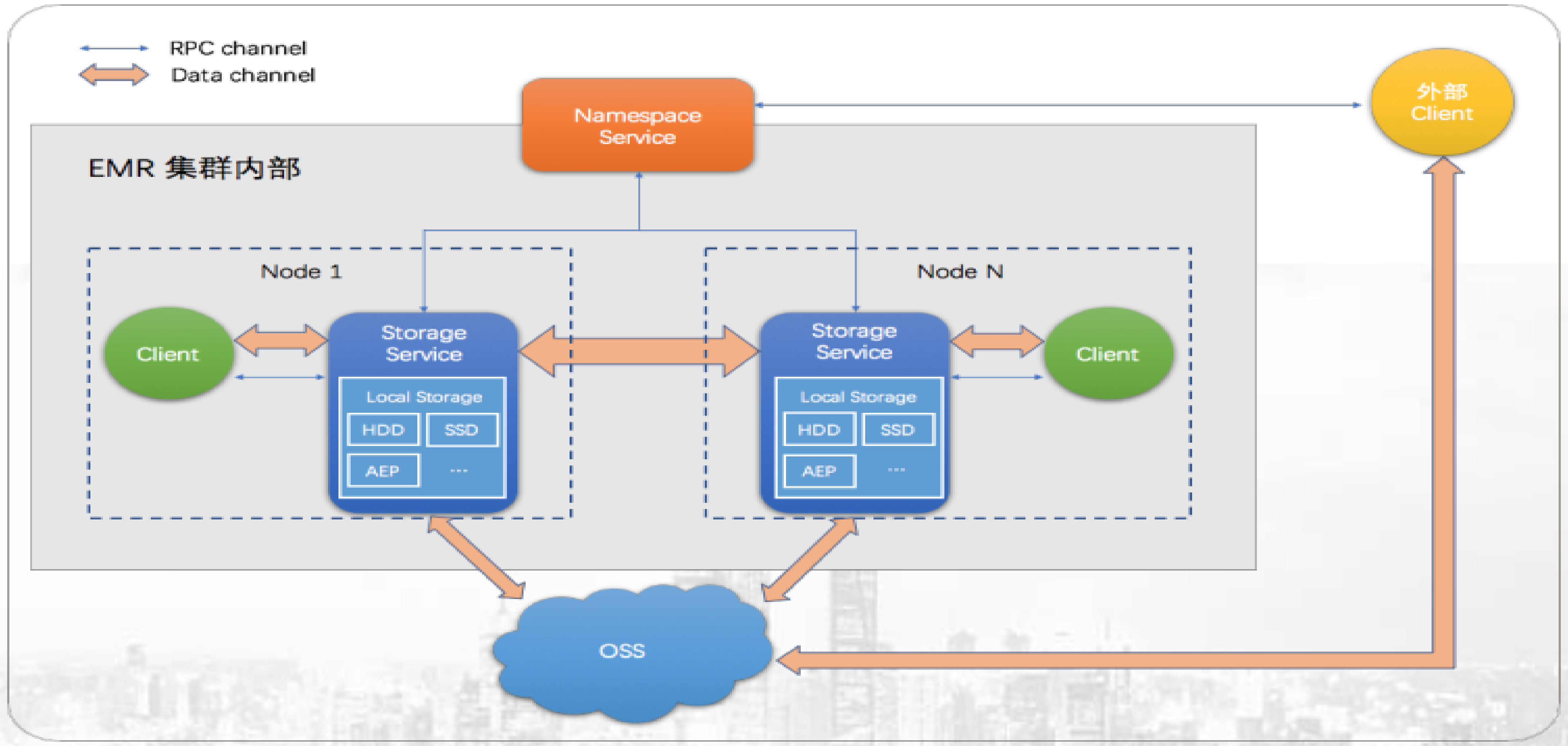
The challenge of disaggregation:

- Object store metadata management.
- Limited network resource.

Storage and Computing Disaggregation

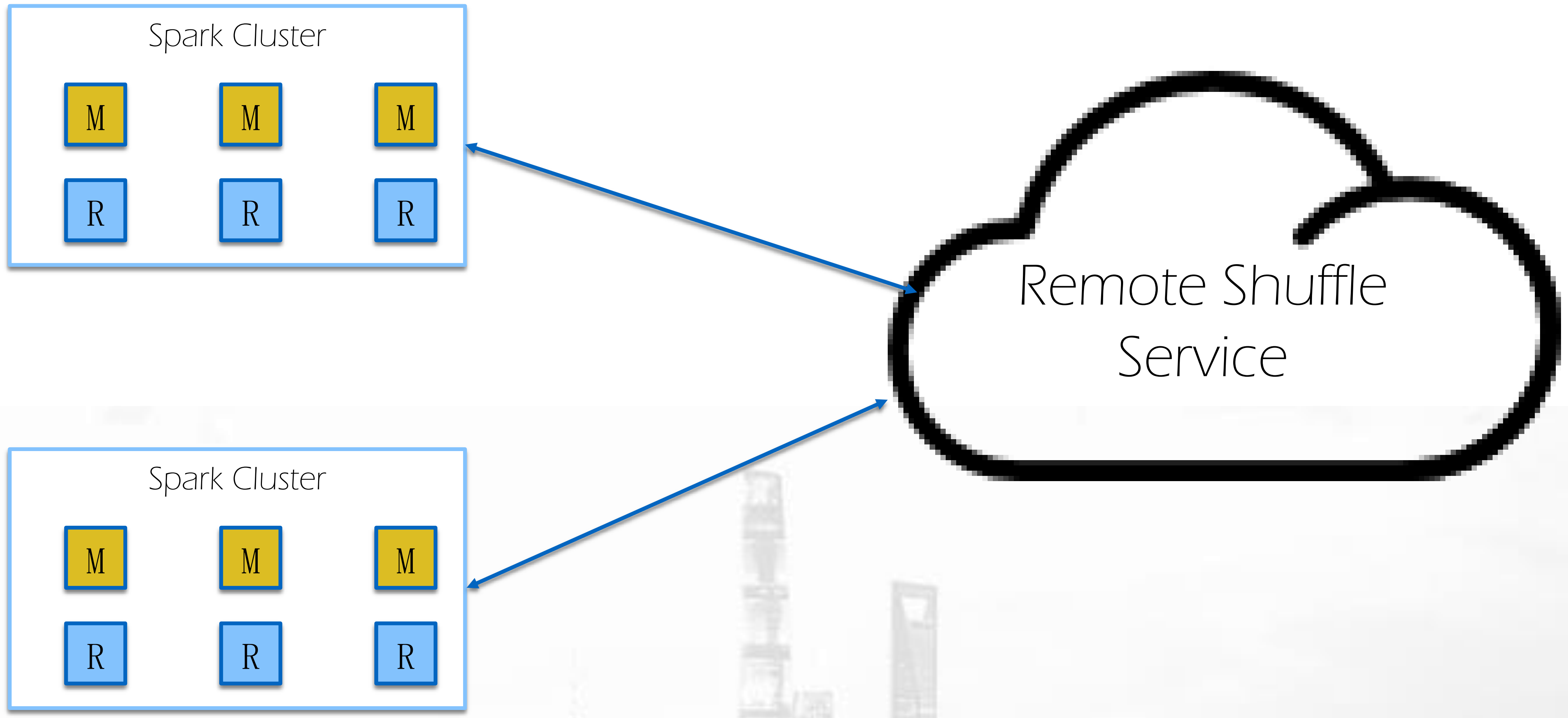
EMR JindoFS fill the gap between object store and compute framework:

- File System API and meta management.
- Local replication support. Remote reliable storage and fast local access.
- Automatic and transparent cold data separation and migration
- Optimized for machine learning and Spark AI



Spark on Cloud: Remote Shuffle Service

- Data source storage is disaggregated from computing while local shuffle data is not.
- Local storage has poor elasticity.
- Current external shuffle service make cost extra effort for worker/nodemanager, and is not available for k8s.
- [SPARK-25299] would support write shuffle file to remote storage, remote shuffle service is still on the way.



- Pay as you go.
- Service run on hosts optimized for it.

Spark on Kubernetes

Natively support since 2.3

Pyspark/R binding and client mode supported since 2.4

Spark 3.0+

- Dynamic allocation support
- Kerberos support
- …

Spark + AI

Project Hydrogen: Spark + AI

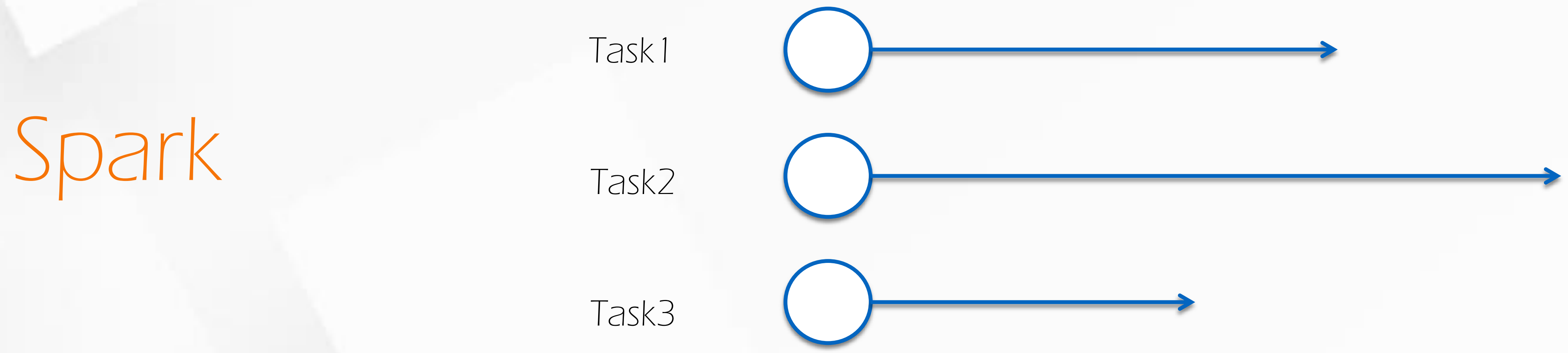
- Better AI need big data
- Data analysis get deeper
- Hydrogen make Spark a unified AI processing pipeline

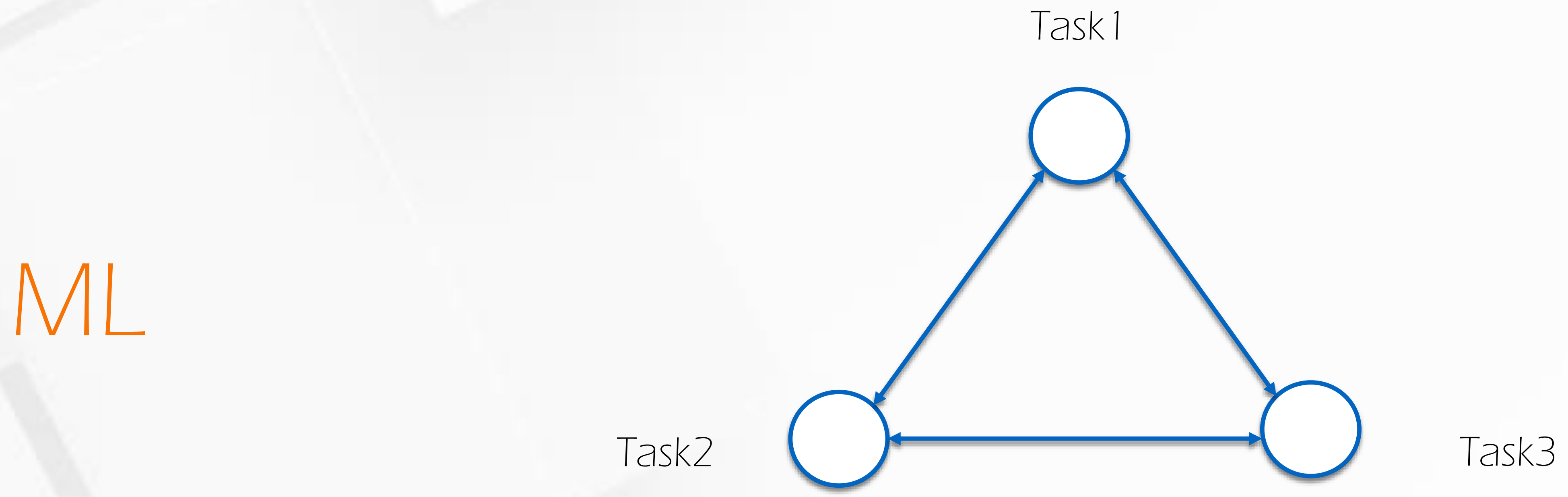
Barrier Execution Mode

Accelerator Aware Scheduling

Optimized Data Exchange

Project Hydrogen: Barrier Execution





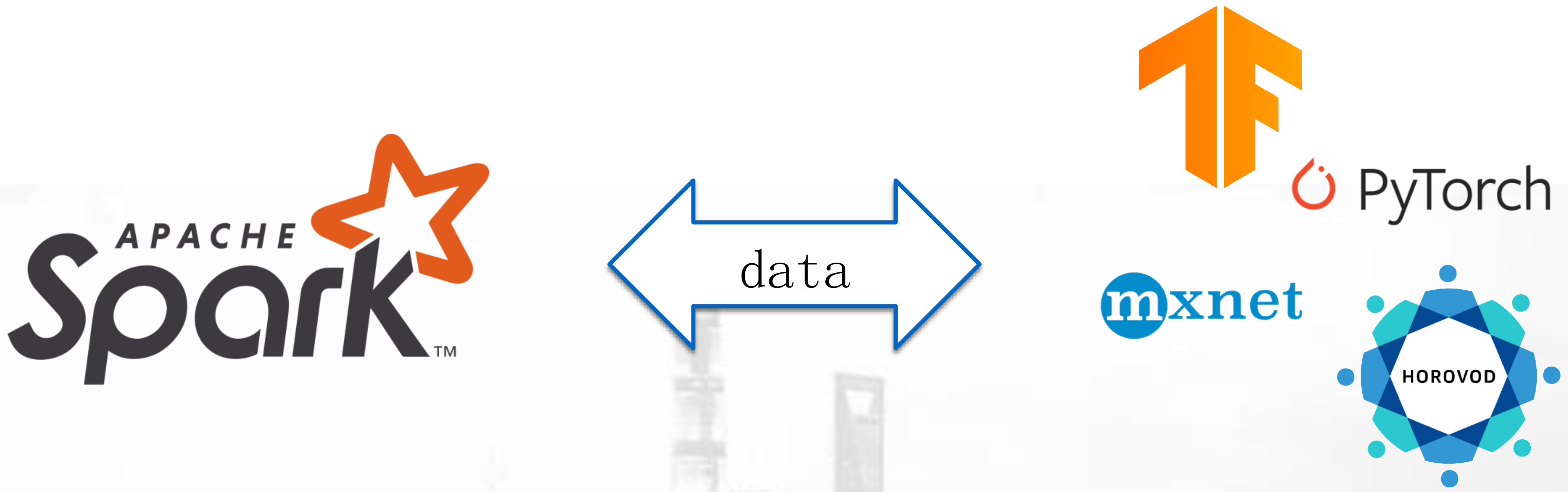
- Gang scheduling enabled to run DL job as Spark stage.
- Specific recovery strategy supported for gang scheduled stage.
- Available since 2.4

Project Hydrogen: Accelerator Aware Scheduling

- GPUs are applied at application level.
- User can retrieve assigned GPUs from task context.
- Can extend to other accelerator, such as: FPGA
- Available at 3.0, see [SPARK-27362], [SPARK-27363]

Project Hydrogen: Optimized Data Exchange

- Spark loads/saves data from/to persistent storage in a data format used by a DL/AI framework.
- Spark feeds data into DL/AI frameworks for training.
- Prefer to use Apache Arrow as exchange data format.
- [SPARK-24615] WIP



Spark 3.0

3.0 Targets

- Project Hydrogen
  - GPU-Aware scheduling
  - Optimized data exchange
- Adaptive Execution
  - Self tuning the number of reducers
  - Adaptive join strategy
- Data Source V2

- Spark on K8s
  - Dynamic resource allocation
  - Kerberos support
- Hadoop 3.x support
- Hive 2.3 support
- Scala 2.12 GA
- Better ANSI SQL compliance

This presentation may contain projections or other forward-looking statements regarding the upcoming release (Apache Spark 3.0). The statements are intended to outline our general direction. They are intended for information purposes only. They are not a commitment to deliver code or functionality. The development, release and timing of any feature or functionality described for Apache Spark remains at the sole discretion of ASF and the Apache Spark PMC.

阿里云开发者社区

扫码加入社群
与志同道合的码友一起
Code Up

大数据计算开发者...

该群属于“阿里云ACE”部门群，仅组织内部成员可以加入，如果组织外部人员收到此分享，需要先申请加入该组织。

谢谢！