

ODPS MapReduce 实现和开放实践

大纲

- ODPS 介绍
- ODPS MapReduce
- MapReduce 作为开放服务

ODPS

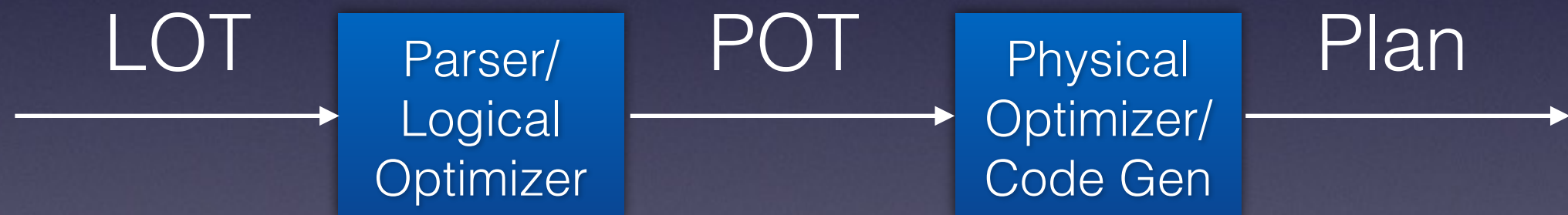
- ODPS: Open Data Process System
- 作为大规模数据处理的底层平台
 - 每日提交数十万计任务
 - 跨数据中心的超大规模集群
- 多种编程模型、范式支持

ODPS 用户视角



Logical Operator Tree (LOT)

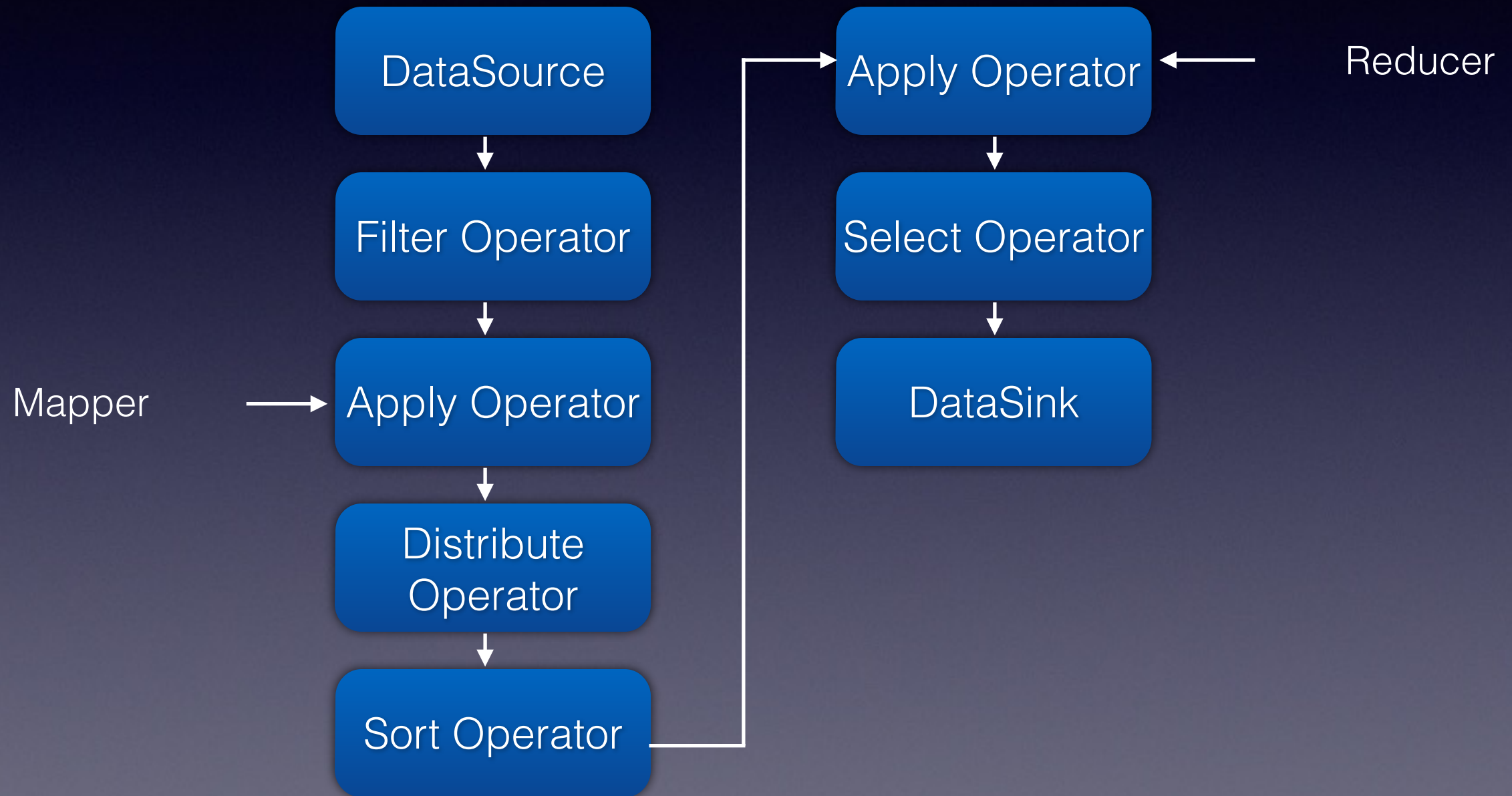
- 工作流程



LOT抽象

- 关系代数抽象
- 支持的Operator
 - Apply / Window
 - Select / Join / Union / Sort
 - DataSource / DataSink
 - ...

MapReduce on LOT



MapReduce API (预览)

```
/**
 * Counts the words in each record. For each record, emit each column as
 * (<b>word</b>, <b>1</b>).
 */
public static class TokenizerMapper extends MapperBase {

    Record word;
    Record one;

    @Override
    public void setup(TaskContext context) throws IOException {
        word = context.createMapOutputKeyRecord();
        one = context.createMapOutputValueRecord();
        one.setBigint(0, 1L);
    }

    @Override
    public void map(long recordNum, Record record, TaskContext context)
        throws IOException {
        for (int i = 0; i < record.getColumnCount(); i++) {
            String[] words = record.get(i).toString().split("\\s+");
            for (String w : words) {
                word.setString(0, w);
                context.write(word, one);
            }
        }
    }
}
```


MapReduce API (预览)

```
public static void main(String[] args) throws OdpsException {
    if (args.length != 2) {
        System.err.println("Usage: wordcount <in_table> <out_table>");
        System.exit(2);
    }

    JobConf job = new JobConf();
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(SumCombiner.class);
    job.setReducerClass(SumReducer.class);
    job.setMapOutputKeySchema(new Column[] { new Column("word", OdpsType.STRING) });
    job.setMapOutputValueSchema(new Column[] { new Column("count",
        OdpsType.BIGINT) });

    InputUtils.addTable(TableInfo.builder().tableName(args[0]).build(), job);
    OutputUtils.addTable(TableInfo.builder().tableName(args[1]).build(), job);

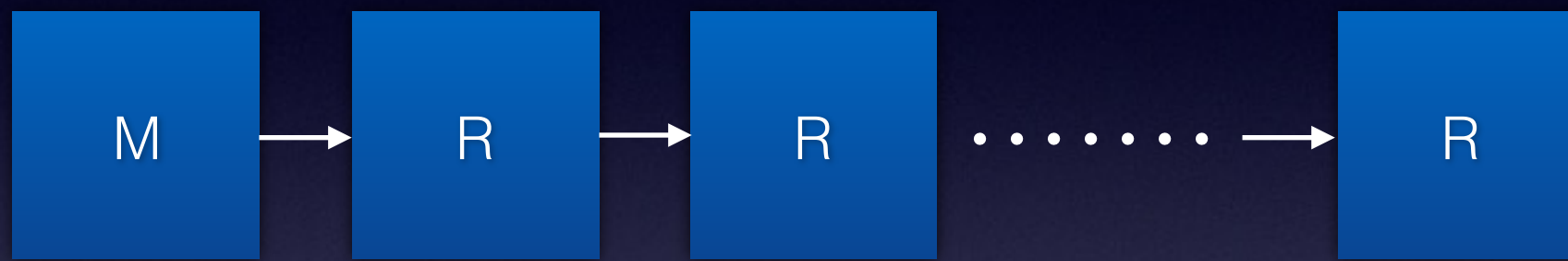
    JobClient.runJob(job);
}
```

MapReduce API

- Hadoop API风格
- 结构化数据
- 强类型接口
- 更多特性：文件接口，非结构化数据的shuffle，
.....

MapReduce 更进一步

- Pipeline: 单任务级连的多次shuffle



- 优势
 - 减少IO消耗
 - 降低调度代价
 - 进程重用

Pipeline 范例

```
/**
 * 构造Pipeline的过程中, 如果不指定Mapper的OutputKeySortColumns, PartitionColumns, OutputGroupingColumns,
 * 框架会默认使用其OutputKey作为此三者的默认配置
 */
Pipeline pipeline = Pipeline.builder()
    .addMapper(TokenizerMapper.class)
    .setOutputKeySchema(
        new Column[] { new Column("word", OdpsType.STRING) })
    .setOutputValueSchema(
        new Column[] { new Column("count", OdpsType.BIGINT) })
    .setOutputKeySortColumns(new String[] { "word" })
    .setPartitionColumns(new String[] { "word" })
    .setOutputGroupingColumns(new String[] { "word" })

    .addReducer(SumReducer.class)
    .setOutputKeySchema(
        new Column[] { new Column("count", OdpsType.BIGINT) })
    .setOutputValueSchema(
        new Column[] { new Column("word", OdpsType.STRING) })

    .addReducer(IdentityReducer.class).createPipeline();

job.setPipeline(pipeline);

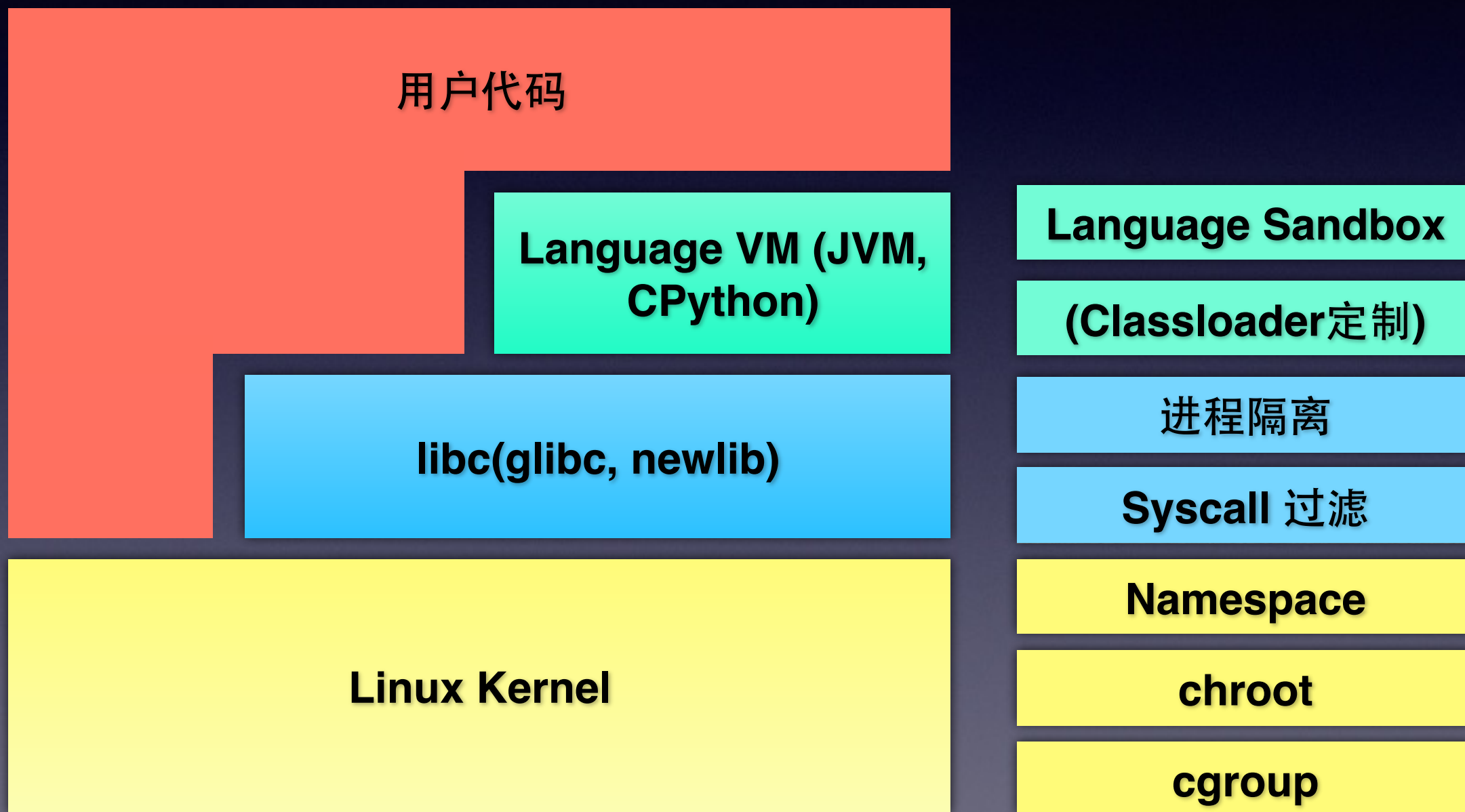
job.addInput(TableInfo.builder().tableName(args[0]).build());
job.addOutput(TableInfo.builder().tableName(args[1]).build());

job.submit();
```


MapReduce 作为开放服务

- 2013.12 内部算法竞赛
- 2014.4-7 天池算法大赛
 - 全球1500+支团队参赛
 - 提交MapReduce任务超过100,000次
- 2015 对外正式发布

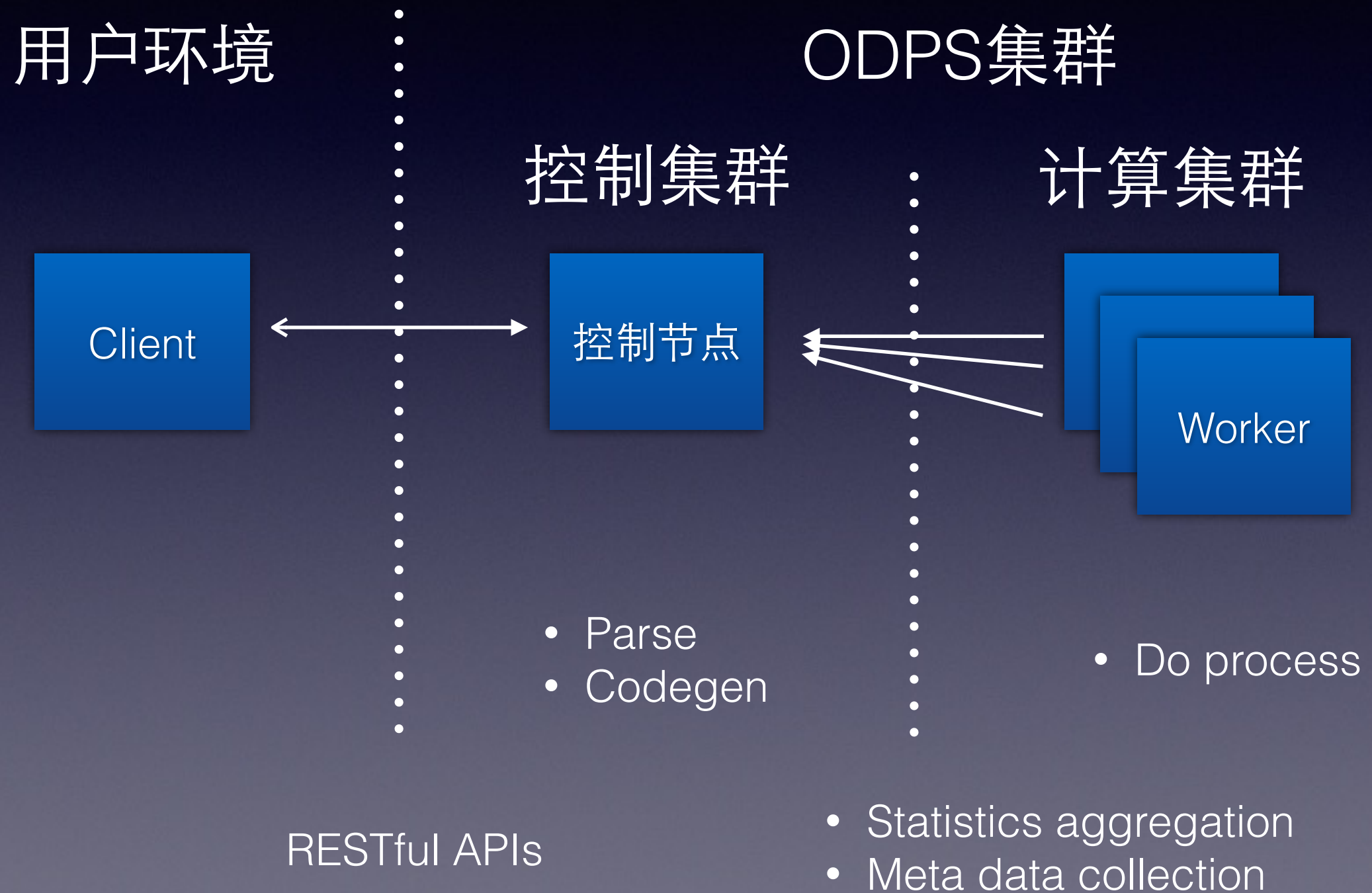
安全



多租户

- 模型
 - user
 - project
 - group
- 权限控制
 - 控制单元: column, table, project
 - package
- Quota
- 隔离

网络隔离



服务案例

- 天池大数据算法竞赛
 - ETL：归一化、直方图、去噪.....
 - 特征提取：访问转化率、（关联分析）支持度.....
- 生产级应用
 - 快的打车
 - 政府、公众事业项目

面临的挑战

- 高效的开发平台
 - IDE
 - 用户终端
- 更方便灵活的用户接口

欢迎试用



开放数据处理服务ODPS

开放数据处理服务（Open Data Processing Service，简称ODPS）由阿里云自主研发，提供针对TB/PB级数据、实时性要求不高的分布式处理能力，应用于数据分析、挖掘、商业智能等领域。阿里巴巴的数据业务都运行在ODPS上

立即开通

价格总览

🔔 ODPS已开放功能查阅



<http://www.aliyun.com/product/odps>

谢谢!

徐冬 shaojie@alibaba-inc.com