Sloth: 网易流计算服务化平台架构实践

金晓军

网易大数据
Netease Big Data

CNUTCon 2017
全球运维技术大会

上海·光大会展中心大酒店 | 2017.9.10–11

智能时代的新运维

大数据运维

DevOps

安全

SRE

Kubernetes

Serverless

游戏运维

AIOps

智能化运维

基础架构

监控

互联网金融

主办方 Geekbang. 极客邦科技 InfoQ

个人介绍

金晓军

网易杭州研究院 大数据技术负责人&首席架构师

2010年毕业加入阿里，高级技术专家。Aliyun StreamCompute(galaxy)架构师兼核心开发。一直从事分布式计算平台设计与研发工作。

现负责网易大数据平台团队建设，人才培养。整体架构设计，自研系统研发与开源组件功能扩展与集成，网易大数据产品输出。

网易大数据
Netease Big Data

01

数据平台

网易大数据
Netease Big Data

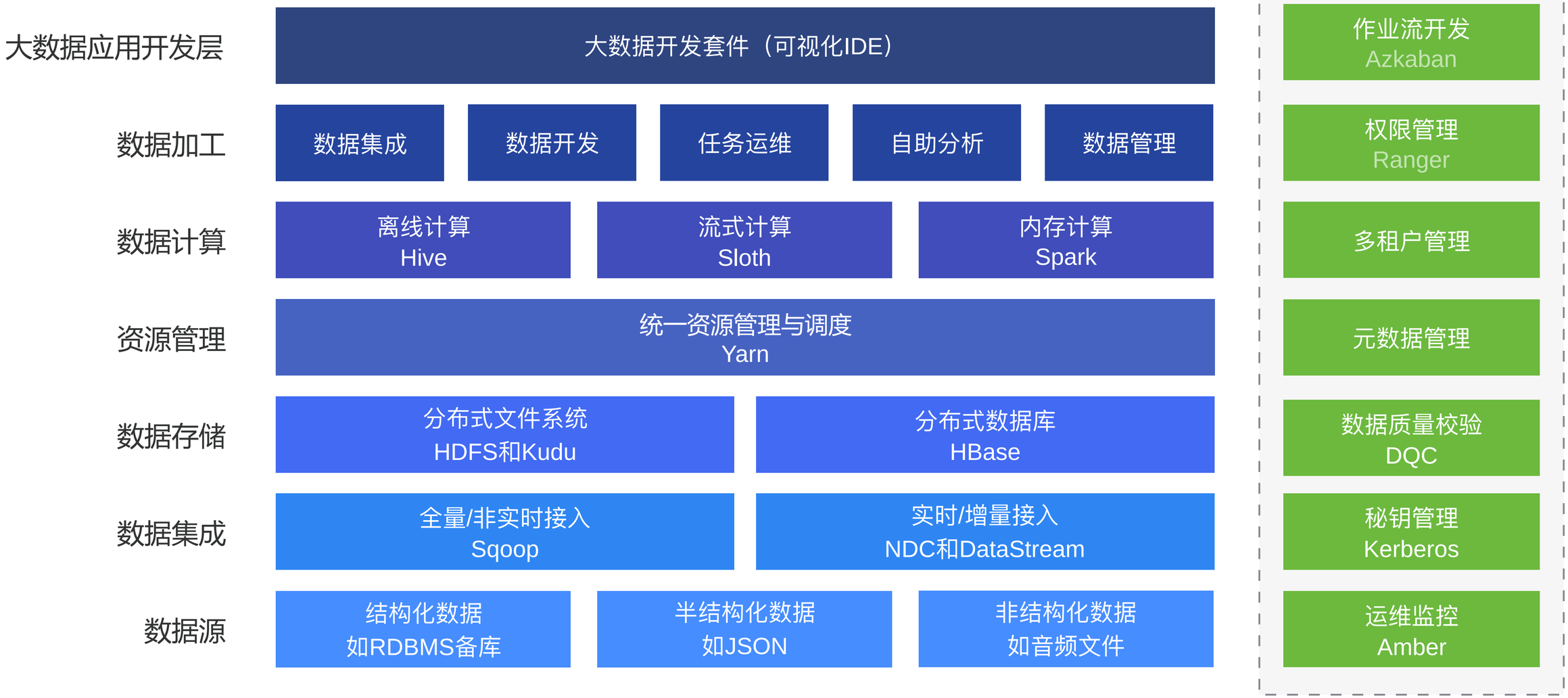
数据业务架构

网易大数据
Netease Big Data

数据应用	网易新闻	网易云音乐	网易考拉	网易云课堂	……
数据平台	网易有数 / 网易猛犸		数据存储计算服务		
数据采集	日志	Database	直接导入	其他仓库	……

网易大数据体系

网易大数据
Netease Big Data



自研系统与开源组件的结合

网易大数据
Netease Big Data

统一元数据服务

- Hive, spark, impala,hbase元数据打通
- 数仓体系内，用户无需在不同的系统一之间做元数据同步
- 不同组件组件之前，数据全增量同步

流计算服务

- Sloth流计算服务化平台
- 通过增量计算的方式，来完成流计算任务
- 使用SQL作为开发方式，完全与离线SQL兼容，支持window/join/subquery/having/retracting等复杂SQL功能

数据安全与权限

- HDFS/Hive/Impala/Spark等组件自动 权限同步
- 支持到列级别的权限控制，支持数据自动加密，即使被拖库，也不用担心敏感数据泄露

一站式

- 一站式的数据平台，数据地图
- 基于ambari开发的一站式的统一部署，监控，运维体系

02

流计算服务化平台

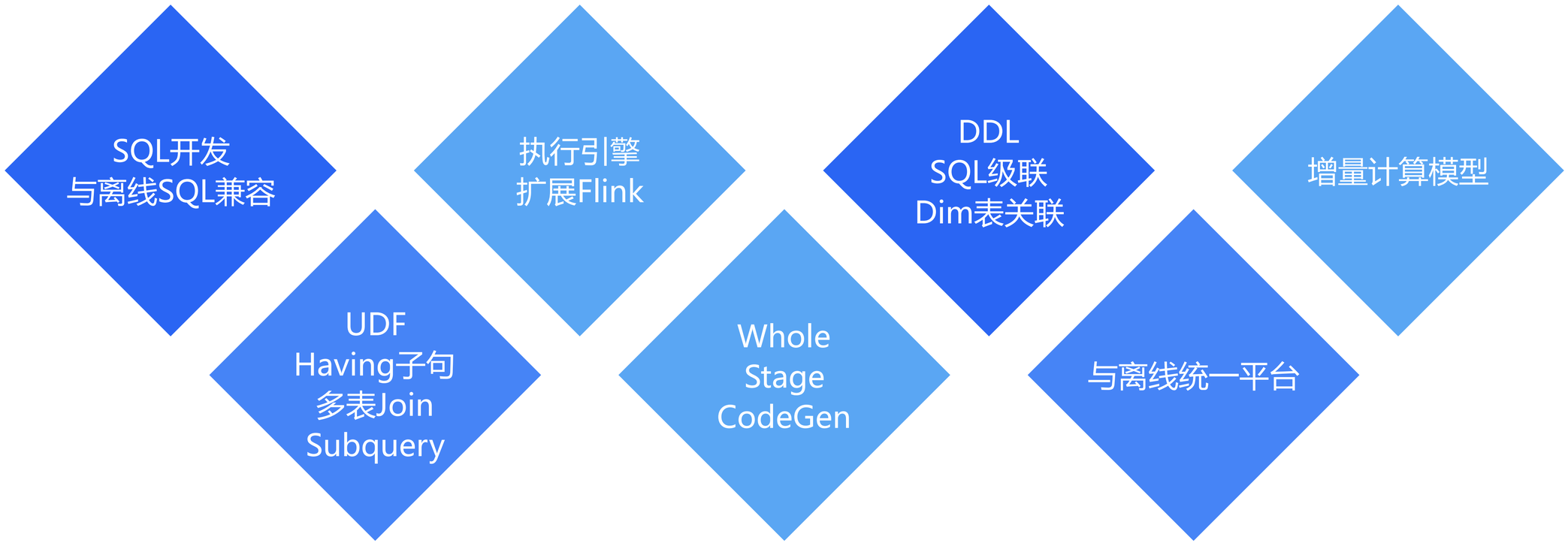
网易大数据
Netease Big Data

Sloth——流计算服务化平台

网易大数据
Netease Big Data

Sloth 特点

网易大数据
Netease Big Data



Sloth 增量计算

网易大数据
Netease Big Data

考拉要对所有商家按销售额做分类统计，销售额在[0,100]区间内的归为一类，[100,200]区间的的归为一类，以此类推，通过计算输出每个区间内的商家个数。

这个任务可以用SQL定义为：

```
-- stage1：计算每个商家的销售总额

INSERT INTO tmp

SELECT seller_id, sum(payment) as total FROM source GROUP BY seller_id;

-- stage2: 计算每个销售额区间内的商家个数

SELECT count(seller_id) as num, total/100 as range FROM tmp GROUP BY (total/100);
```

Sloth 增量计算

网易大数据
Netease Big Data

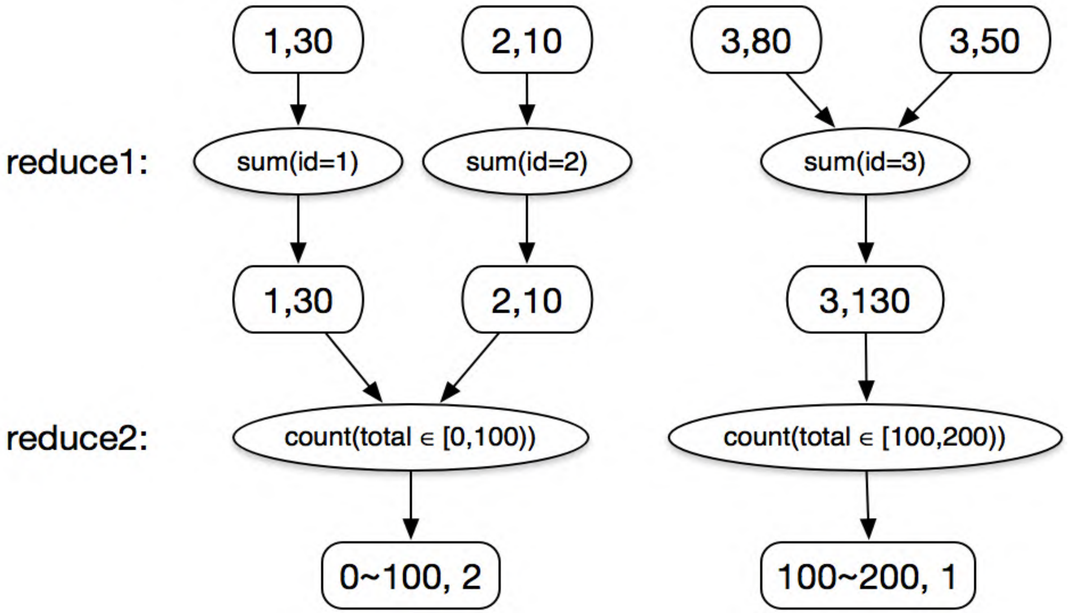
输入数据 | 离线计算 | 流式计算 | 增量计算

1,30 2,10 3,80 3,50

Sloth 增量计算

网易大数据
Netease Big Data

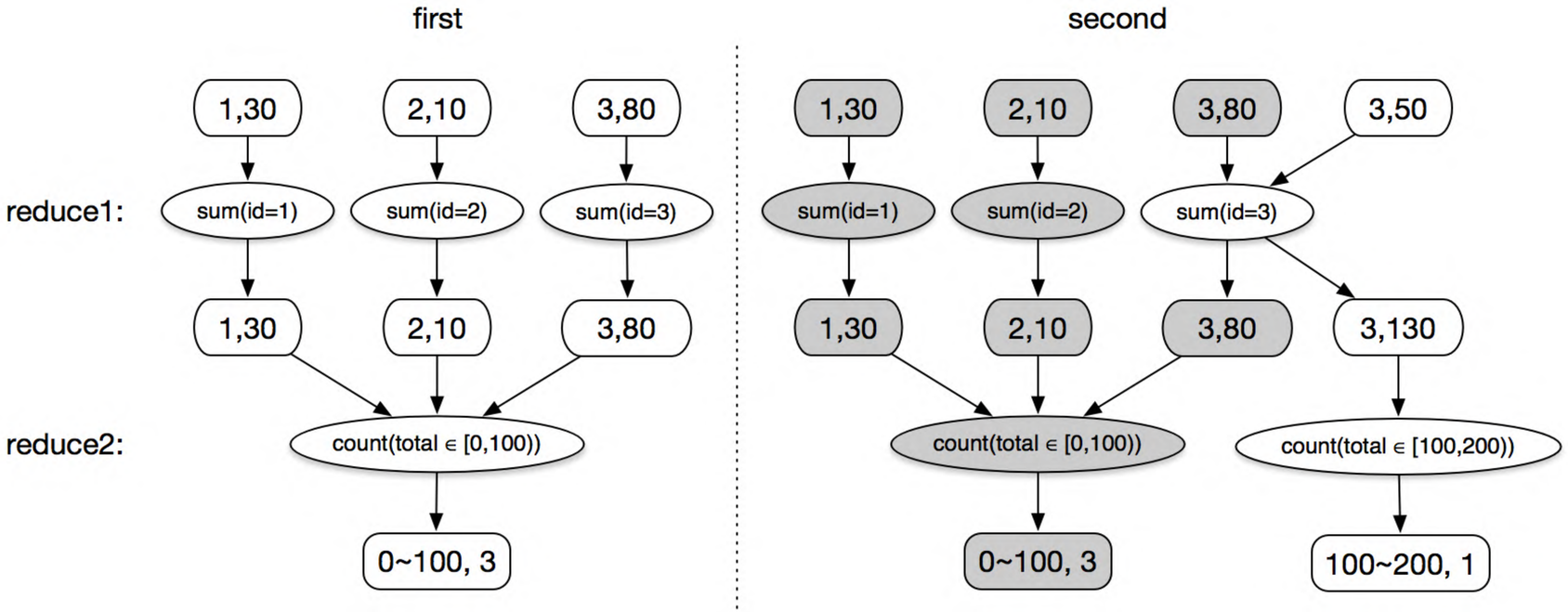
输入数据 | 离线计算 | 流式计算 | 增量计算



Sloth 增量计算

网易大数据
Netease Big Data

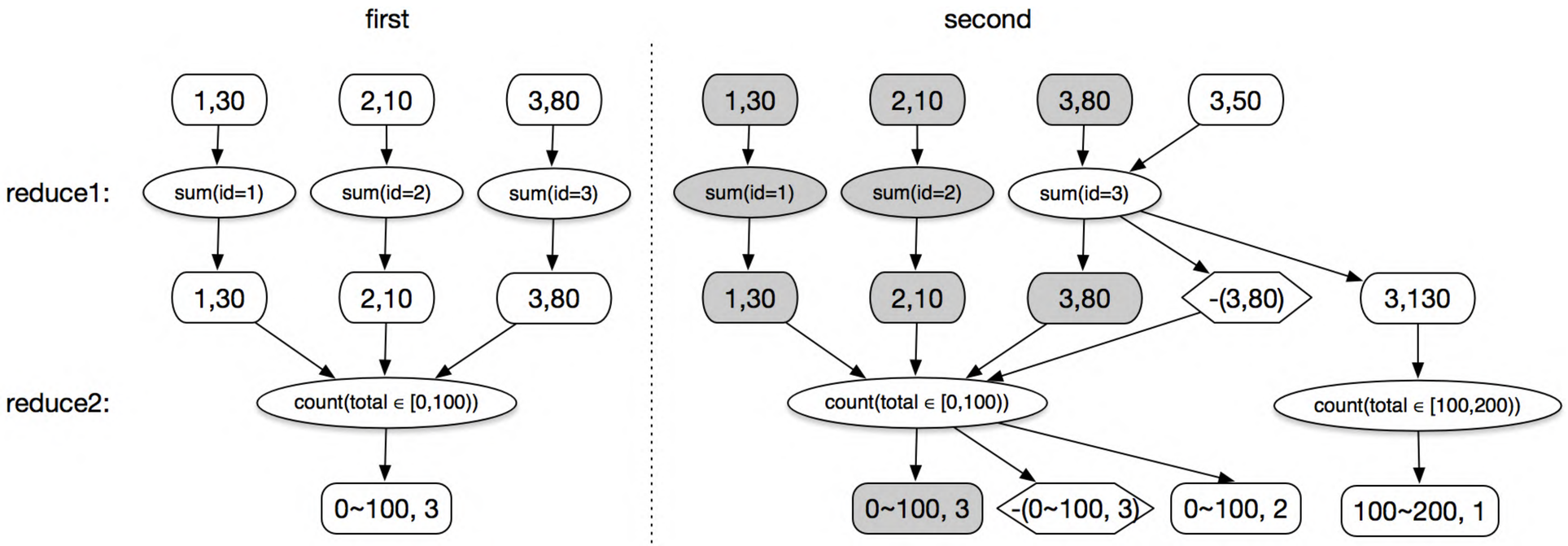
输入数据 | 离线计算 | 流式计算 | 增量计算



Sloth 增量计算

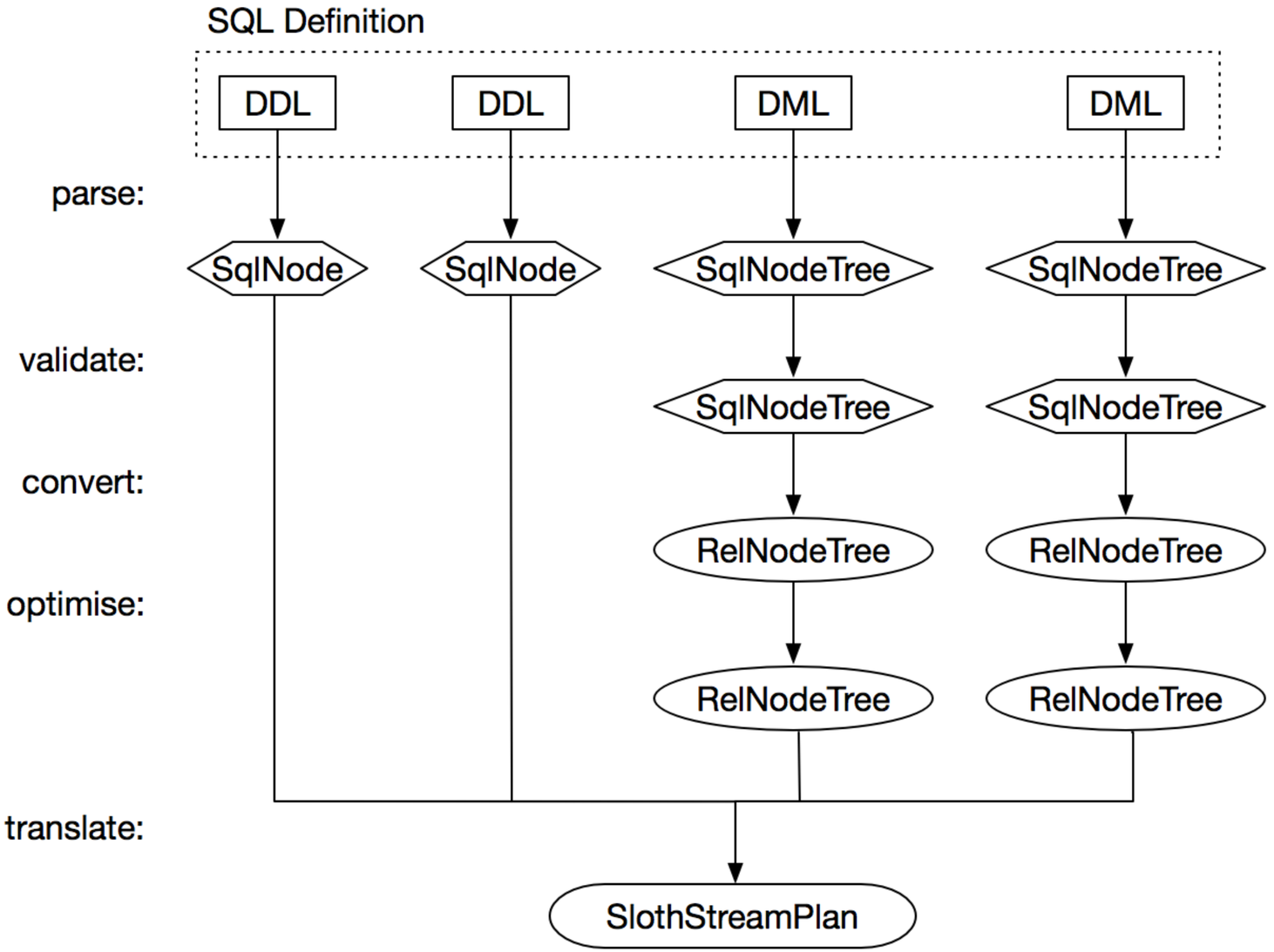
网易大数据
Netease Big Data

输入数据 | 离线计算 | 流式计算 | **增量计算**



SlothPyhsicalPlan

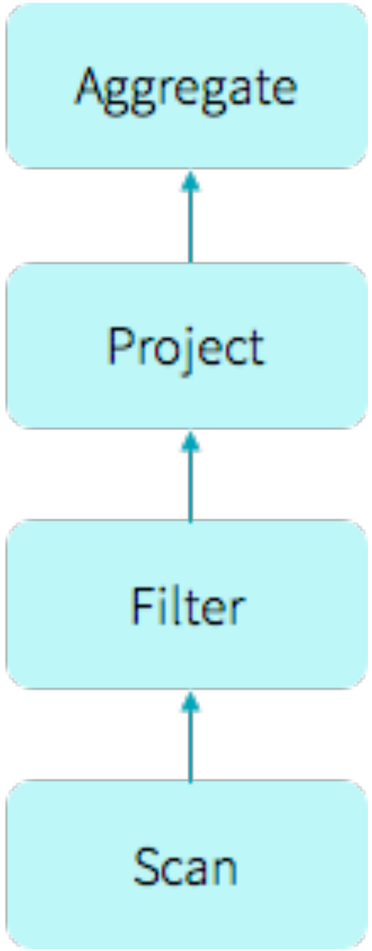
网易大数据
Netease Big Data



Whole Stage Codegen

网易大数据
Netease Big Data

```
select count(*) from store_sales
where ss_item_sk = 1000
```



Whole Stage Codegen

网易大数据
Netease Big Data

```
class Filter(child: Operator, predicate: (Row => Boolean))
  extends Operator {
  def next(): Row = {
    var current = child.next()
    while (current == null || predicate(current)) {
      current = child.next()
    }
    return current
  }
}
```

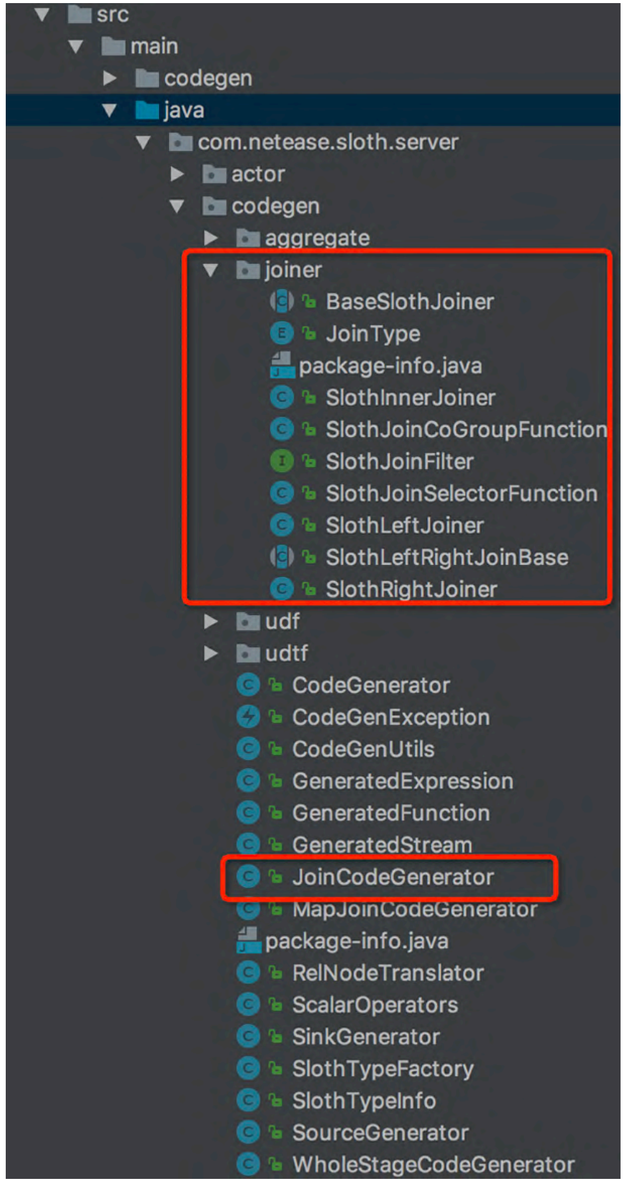
Whole Stage Codegen

网易大数据
Netease Big Data

```
var count = 0

for (ss_item_sk in store_sales) {

  if (ss_item_sk == 1000) {

    count += 1

  }

}
```

Whole Stage Codegen

网易大数据
Netease Big Data




```
@Override
public GeneratedExpression visitCall(RexCall call) {
    List<GeneratedExpression> operands = Lists.newArrayList();
    SqlOperator operator = call.getOperator();
    for (RexNode rexNode : call.getOperands()) {
        if (operator.equals(SqlStdOperatorTable.EQUALS)) {
            RexNode rexNode1 = call.getOperands().get(0);
            RexNode rexNode2 = call.getOperands().get(1);

            if (rexNode1 instanceof RexInputRef && rexNode2 instanceof RexInputRef) {
                //get pk list here
                Map.Entry<Integer, Integer> inputIndex1 = this.getInputIndex((RexInputRef)
                Map.Entry<Integer, Integer> inputIndex2 = this.getInputIndex((RexInputRef)

                if (Objects.equals(inputIndex1.getValue(), inputI
                    throw new CodeGenException("join condition er
                }

                if (!Objects.equals(rexNode1.getType(), rexNode2.
                    throw new CodeGenException("join condition er
                }

                if (inputIndex1.getValue() == 0) {
                    leftJKIndex.add(inputIndex1.getKey());
                    rightJKIndex.add(inputIndex2.getKey());
                } else {
                    leftJKIndex.add(inputIndex2.getKey());
                    rightJKIndex.add(inputIndex1.getKey());
                }
            } else { // there is exactly one rexNode tree of call.getOperands() contai

                RexInputRef filterRexNode;

                if (rexNode1 instanceof RexInputRef) {
                    if (containRexInputRef(rexNode2)) {
                        throw new CodeGenException("syntax error.");
                    }
                    filterRexNode = (RexInputRef) rexNode1;
                } else if (rexNode2 instanceof RexInputRef) {
                    if (containRexInputRef(rexNode1)) {
                        throw new CodeGenException("syntax error.");
                    }
                    filterRexNode = (RexInputRef) rexNode2;
                } else {
                    throw new CodeGenException("syntax error.");
                }

                if (filterRexNode.getIndex() < input1.getArity()) {
                    GeneratedExpression exp = codeGen1.visitCall(call);
                    leftStreamFilter.append(String.format("%s"
                                    + "if(!%s) {"
                                    + "return false;"
                                    + "}\n",
                            exp.getCode(),
                            exp.getResultTerm()));
                } else {
                    GeneratedExpression exp = codeGen2.visitCall(call);
                    rightStreamFilter.append(String.format("%s"
                                    + "if(!%s) {"
                                    + "return false;"
                                    + "}\n",
                            exp.getCode(),
                            exp.getResultTerm()));
                }
            }
```

Registration

UDF—与Hive完全兼容

网易大数据
Netease Big Data

```java
public class UDFAsin implements UDF {

    /**
     * Take Arc Sine of a in radians.
     */
    public Double evaluate(Double data) {
        if (data == null || data > 1 || data < -1) {
            return null;
        }

        return Math.asin(data)
    }
}
```

```java
public class UDFYear implements UDF {
    private final SimpleDateFormat formatter = new SimpleDateFormat( pattern: "yyyy-MM-dd");
    private final Calendar calendar = Calendar.getInstance();

    /**
     * Get the year from a date string.
     *
     * @param dateStr the dateStr in the format of "yyyy-MM-dd HH:mm:ss" or
     *                "yyyy-MM-dd".
     * @return an int from 1 to 12. null if the dateStr is not a valid date
     * string.
     */
    public Integer evaluate(String dateStr) throws ParseException {...}

    public Integer evaluate(Date date) {...}

    public Integer evaluate(Timestamp timestamp) {...}
}
```

UDAF—增量UDAF

网易大数据
Netease Big Data

```java
public interface Aggregate<T> extends Serializable {

    /**...*/
    void prepare(Object value, SlothRecord intermediate);

    /**...*/
    void initiate(SlothRecord intermediate);

    /**
     * Merge intermediate aggregate data into aggregate buffer.
     *
     * @param intermediate The intermediate aggregate row to merge.
     * @param buffer       The aggregate buffer into which the intermedidate is merged.
     */
    void merge(SlothRecord intermediate, SlothRecord buffer);

    /**
     * Calculate the final aggregated result based on aggregate buffer.
     *
     * @param buffer The aggregate buffer from which the final aggregate is computed
     * @return The final result of the aggregate
     */
    T evaluate(SlothRecord buffer);

    /**...*/
    TypeInformation intermediateDataType();

    void setAggOffsetInRow(int offsetInRow);

    /**...*/
    boolean supportPartial();
}
```

Registration

自定义函数注册

网易大数据
Netease Big Data

```
1 CREATE TEMPORARY function 'ERRORUDF' AS 'com.netease.sloth.server.udf.UDFError';
2 CREATE TEMPORARY function 'EXTRACT_CLICK_LOG' AS 'com.netease.sloth.server.codegen.udtf.sample.SampleClickParser';
3 CREATE TEMPORARY function 'EXTRACT_PV_LOG' AS 'com.netease.sloth.server.codegen.udtf.sample.SamplePvParser';
```

流计算开发平台

网易大数据
Netease Big Data

新建Notebook

考拉

考拉

streaming-test-job1

streaming-test-job1

```
1  INSERT INTO level_1_join_result_stream
2  SELECT
3    a.logdate, a.hour_id, a.advertiser_id, a.campaign_id, a.adgroup_id, a.creative_id,
4    sum(case when b.logdate is not null then 1 else 0 end) as num,
5    sum(case when b.logdate is not null then 1 else 0 end) as charge
6  FROM level_1_LEFT_stream a left OUTER join level_1_right_stream b
7  ON a.logdate = b.logdate and a.hour_id = b.hour_id and a.advertiser_id = b.advertiser
8    and a.campaign_id = b.campaign_id and a.adgroup_id = b.adgroup_id and a.creative_id
9  GROUP BY
10   a.logdate, a.hour_id, a.advertiser_id, a.campaign_id, a.adgroup_id, a.creative_id;
11
12 INSERT INTO level_2_join_result_stream
13 SELECT
14   a.logdate, a.hour_id, a.advertiser_id, a.campaign_id, a.adgroup_id,
15   sum(case when b.logdate is not null then 1 else 0 end) as num,
16   sum(case when b.logdate is not null then 1 else 0 end) as charge
17 FROM level_1_join_result_stream a LEFT OUTER join level_1_right_stream b
18 on a.logdate = b.logdate and a.hour_id = b.hour_id and a.advertiser_id = b.advertiser
19   and a.campaign_id = b.campaign_id and a.adgroup_id = b.adgroup_id and a.creative_id
20 GROUP BY
21   a.logdate, a.hour_id, a.advertiser_id, a.campaign_id, a.adgroup_id;
22
```

Sloth与开源的结合

网易大数据
Netease Big Data

为什么不直接使用FlinkSQL?

FlinkSQL支持的功能还是非常简单且没经过大规模生产实践应用。
Sloth是通用的StreamingSQL框架，Codegen生成runtime，支持各类SQL的功能，与底层的执行框架无关，可以对接flink/spark/beam。

将新feature及修复的bug贡献回flink&calcite社区

Sloth开源，预计年底

03

未来技术规划

网易大数据
Netease Big Data

网易大数据平台未来规划

网易大数据
Netease Big Data

Thanks!

金晓军
hzjinxiaojun@corp.netease.com

网易BDMS出品，必属精品

网易大数据
Netease Big Data

让创新技术推动社会进步

HELP TO BUILD A BETTER SOCIETY WITH
INNOVATIVE TECHNOLOGIES

Geekbang.

极客邦科技

InfoQ

专注中高端技术人员的技术媒体

EGO EXTRA GEEKS' ORGANIZATION NETWORKS

高端技术人员学习型社交平台

StuQ 斯达克学院

实践驱动的 IT 教育平台

地址：北京市朝阳区洛娃大厦C座8层1801室　　网址：www.geekbang.org