



Flink China

Flink在车联网的应用与实践

张皓

G7 PaaS平台系统架构师

2018-09-01

G7介绍



Flink China



70万+
车辆



3T+
日均数据量



5万+
客户



22万+
运输线路



1,200+
员工数量



覆盖90%+
行业领袖



Flink China

IT大咖说
知识共享平台

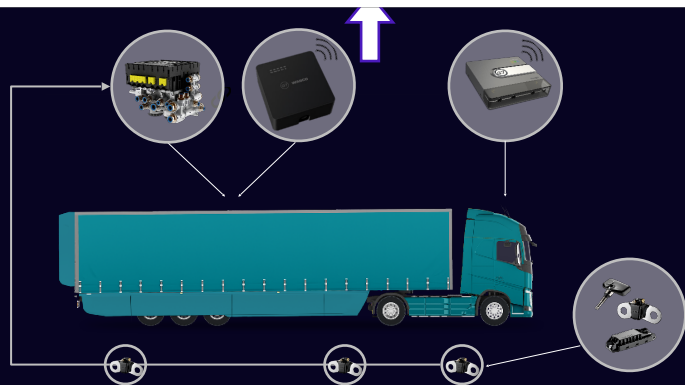
G7业务



Flink China

IoT场景特点

- 传感器数据
- 数据种类多
- 数据质量差
- 数据低延迟
- 硬件设备100+
- 业务数量200+
- 每天产生20亿+轨迹点，
100亿+条数据



目录 CONCENTS



Flink China



实时计算选型



G7业务应用案例



实时计算平台
开发和现状



未来规划



Flink China



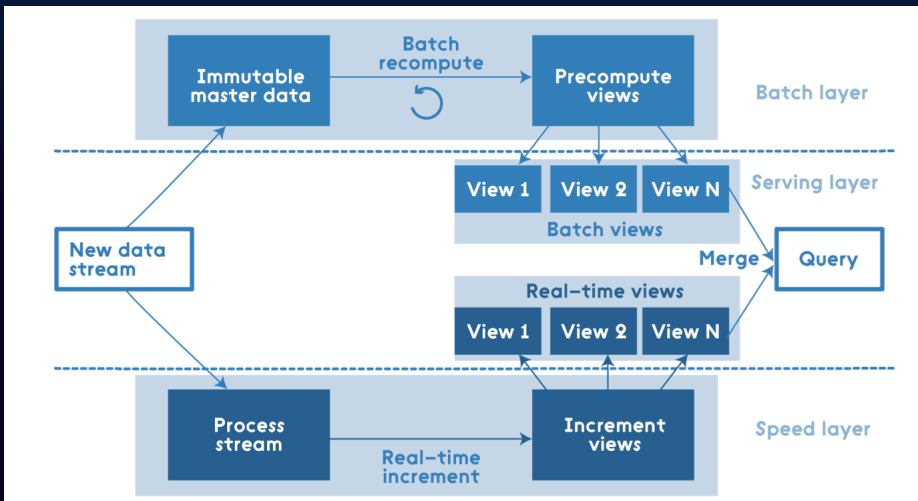
实时计算选型

IoT架构: Fast, 从Lambda到Kappa

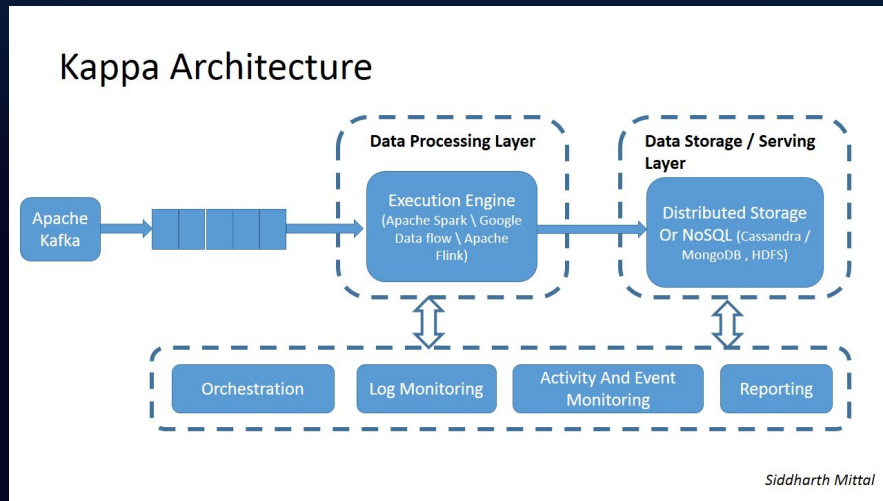


Flink China

Lambda Architecture



Kappa Architecture



Siddharth Mittal

实时流计算框架选型

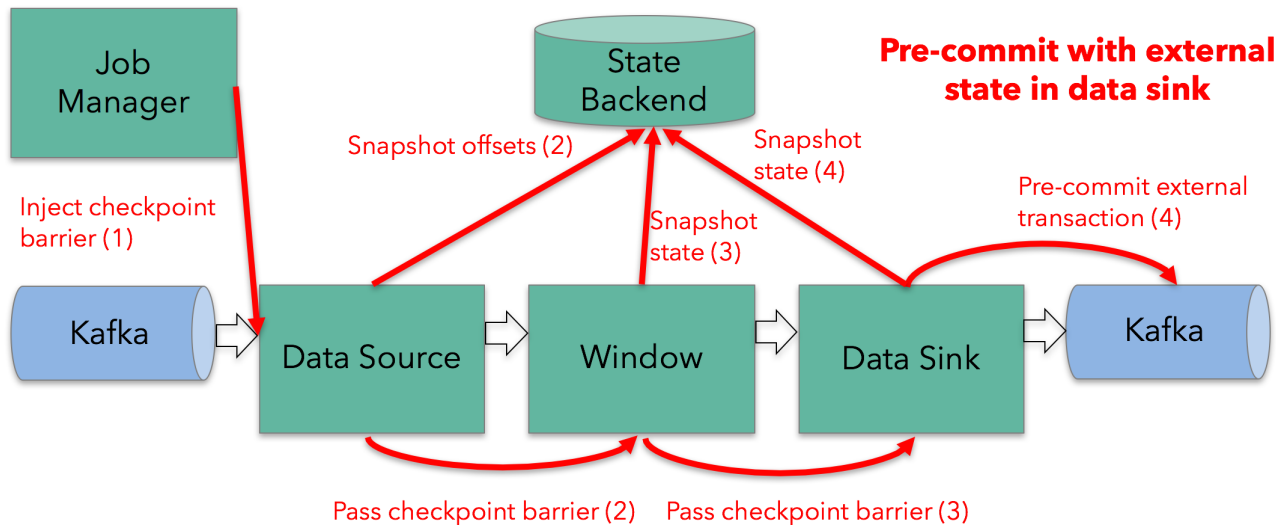


Flink China

	Storm Record Acks	Spark Streaming, Trident Micro-Batching	Google Cloud Dataflow Transactional updates	Flink Distributed snapshots
数据保证	At least once	Exactly once	Exactly once	Exactly once
延迟	very low	High	low	very low
吞吐量	Low	High	Medium to High (由分布式的事务存储决定的)	High
计算模型	Streaming	Micro-batch	Streaming	Streaming
容错机制的开销	High	Low	由分布式事务存储决定	Low
流控	Problematic	Problematic	Natural	Natural
分离	Partially	NO (微批量的大小会影响语义)	Yes	Yes



Exactly-once two-phase commit



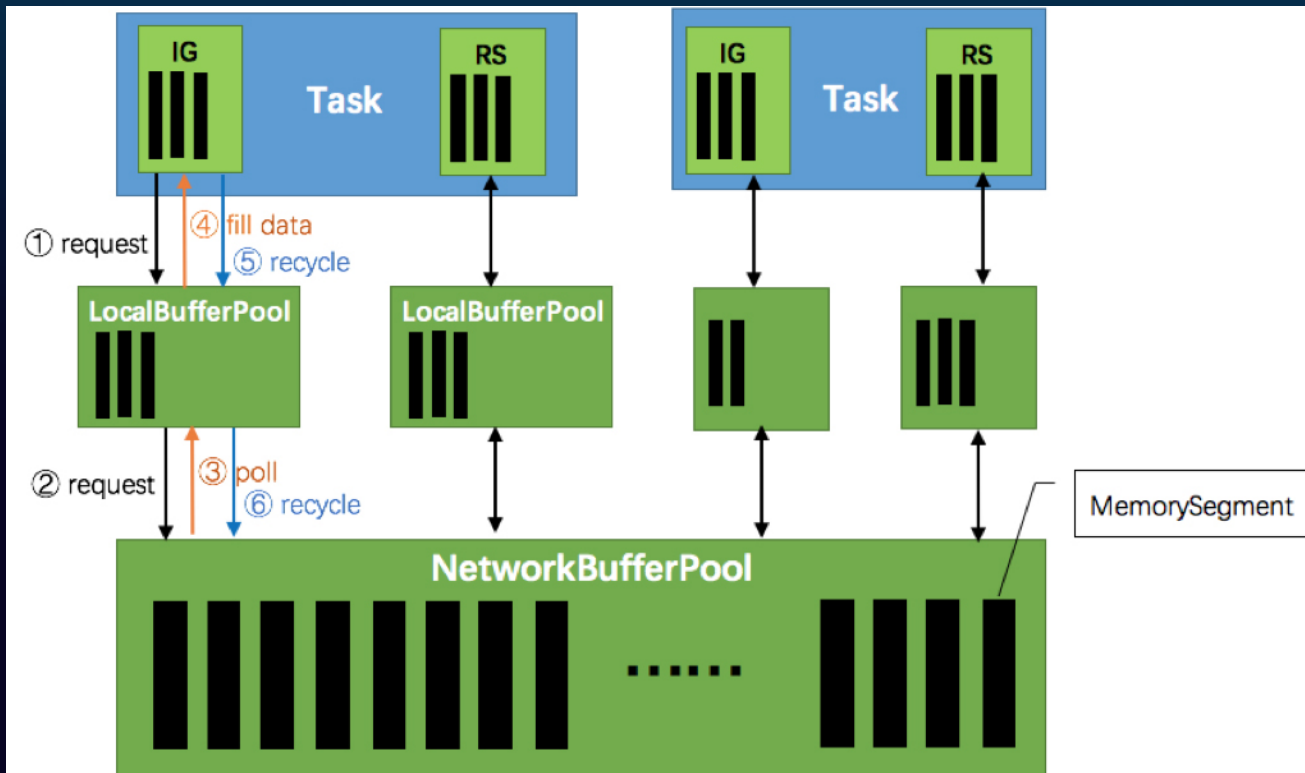
- 内部计算一致性
- end-to-end一致性

背压 Back Pressure



Flink China

- 优雅
 - MemorySegment
 - LocalBufferPool
 - NetworkBufferPool





Flink China



G7业务应用案例



Flink China

Flink目前在G7的应用场景

实时计算



实时ETL



统计分析





Flink China

实时计算业务

疲劳报警事件计算

怠速事件计算



进出区域事件计算



超速事件计算



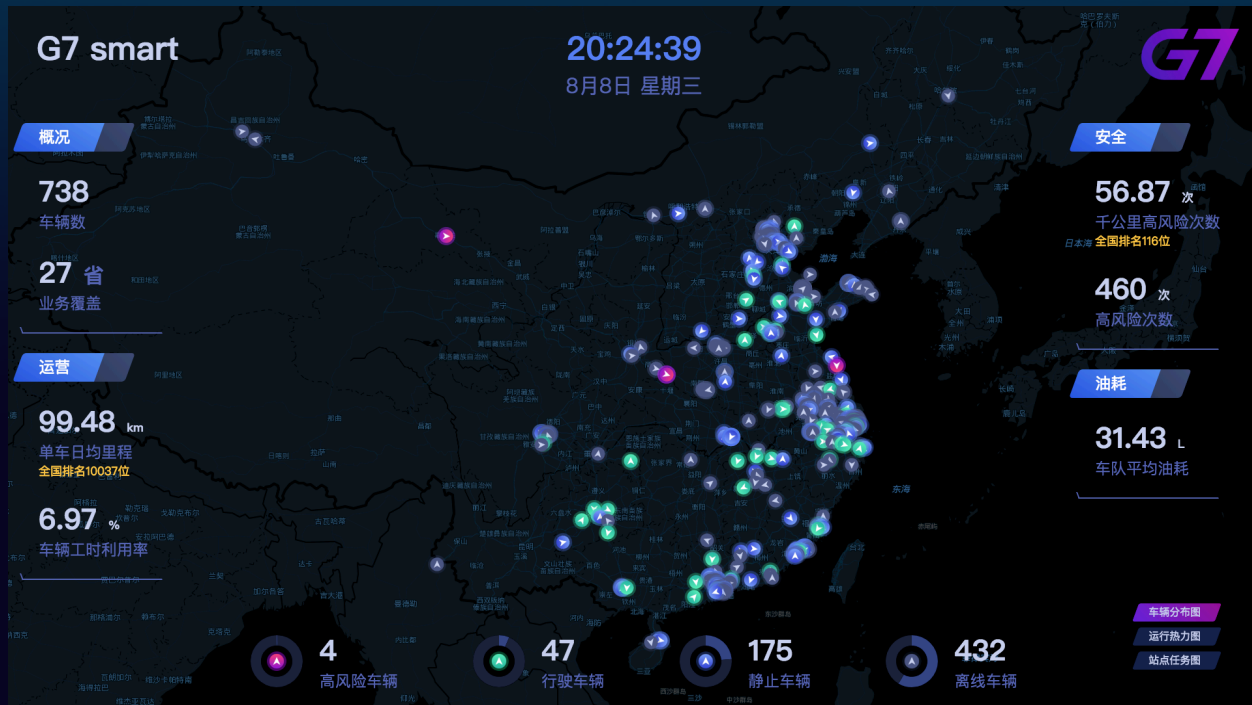
油耗计算

疲劳报警业务模型



Flink China

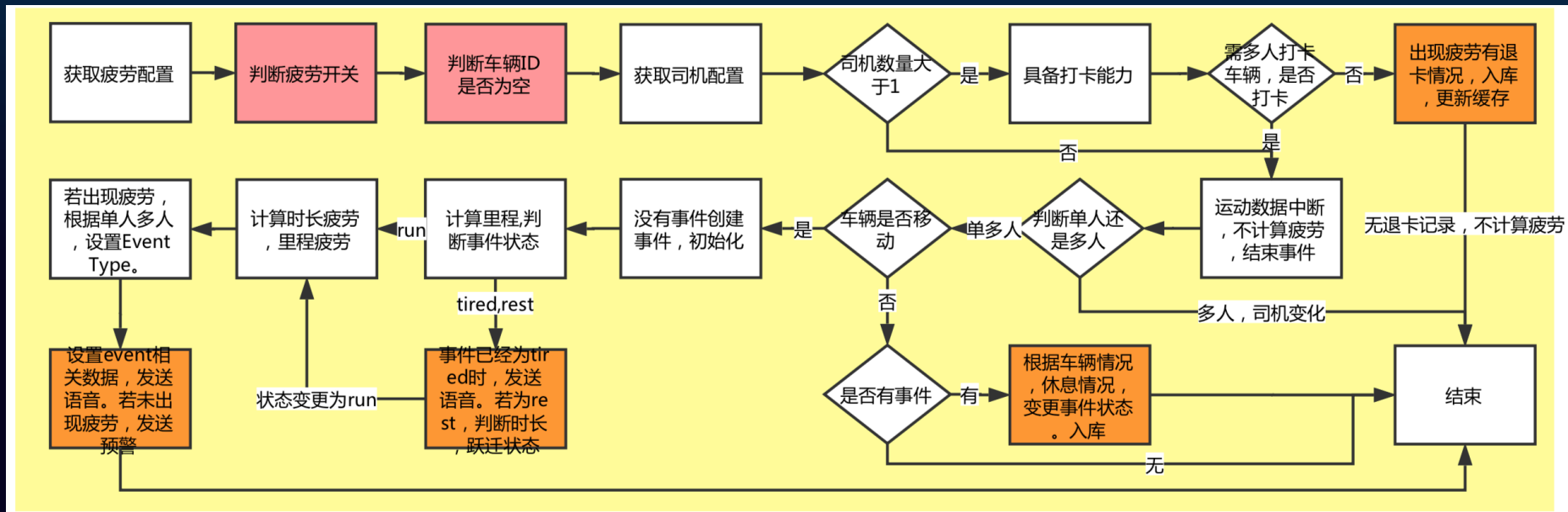
- 根据G7的大数据计算，因为疲劳驾驶造成货车事故的比重占到整个事故的20%。根据场景判断司机是否在疲劳驾驶。
- 根据车辆行驶的里程，驾驶员行驶的里程，驾驶时长，判断是否存在疲劳驾驶。如果存在则报警，如果不存在根据情况提前预警。
- 报警和预警都是下发语音到货车驾驶室。
- 挑战：实时性，稳定性非常高



疲劳报警业务流程



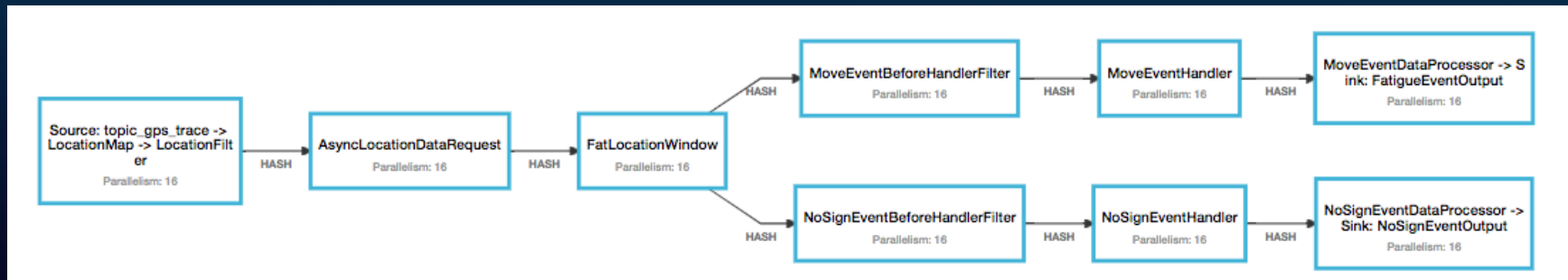
Flink China



疲劳报警Streaming模型



Flink China



Tips

- 算子表达尽量单一
- 每个算子尽量内聚，算子间尽量低耦合
- 算子打散，异步+多线程发挥更好
- 单独设置每个算子单元的并行度，性能更优
- hash和balance根据情况选择

/O 调用dubbo接口, zuul, db, hbase等外部接口





- Kafka to HDFS
- Kafka to DB
(MySQL)
- Kafka to HBase
- Kafka to ES
- Kafka to Kafka

```
FlinkKafkaConsumer09<String> consumer09 = new FlinkKafkaConsumer09<>( topic: "flink-test",
    new SimpleStringSchema(), getKafkaConsumerProperties());

JDBCAppendTableSink sink = new JDBCAppendTableSinkBuilder().setDrivername(DRIVER_CLASS)
    .setDBUrl(DB_URL)
    .setQuery("insert into event (imei, type, lat, lng, time) values (?,?,?,?,?)")
    .setParameterTypes(new int[]{Types.VARCHAR, Types.INTEGER, Types.DOUBLE, Types.D
    .build();

SingleOutputStreamOperator<Row> stream = env.addSource(consumer09).map(new MapFunction<S

    public Row map(String value) {
        //map string to row object
        return new Row(arity: 5);
    }

});
sink.emitDataStream(stream);
```

- Source -> Transformation -> Sink



- 最近一小时全国各城市，车辆总数，司机总数，疲劳事件，进出区域事件，打卡次数，点熄火事件....
- 设备硬件上传报文，根据设备类型，设备数量...
- 实时数仓
- SQL处理

```
FlinkKafkaConsumer09 kafkaTableSource = new FlinkKafkaConsumer09(kafkaTopic,
    new SimpleStringSchema(), initProps());
DataStream<CTBoxDO> stream = env.addSource(kafkaTableSource).map(new CTBoxDOMapper())
    .name("map_" + kafkaTopic);

tableEnv.registerDataStream(kafkaTopic, stream, fields: "imei, truckno, proctime.proctime");
Table table = tableEnv
    .sql( query: "SELECT TUMBLE_START(proctime, INTERVAL '1' MINUTE) as procTime, imei FROM "
        + kafkaTopic + " GROUP BY TUMBLE(proctime, INTERVAL '1' MINUTE) ");

table.writeToSink(
    new CsvTableSink(filePath, fieldDelim: "|", numFiles: 1, FileSystem.WriteMode.OVERWRITE));
```





Flink China



实时计算平台 开发与现状

实时计算平台架构



Flink China

Glink任务管控平台

Flink App

Flink App

Flink App

性能
监控
APM

Cat

集群
监控
管理

Zabbix
/
Ambari

Yarn (计算资源平台)

HDFS

任务
监控
报警

Influxd
b

诊断
调优

ELK

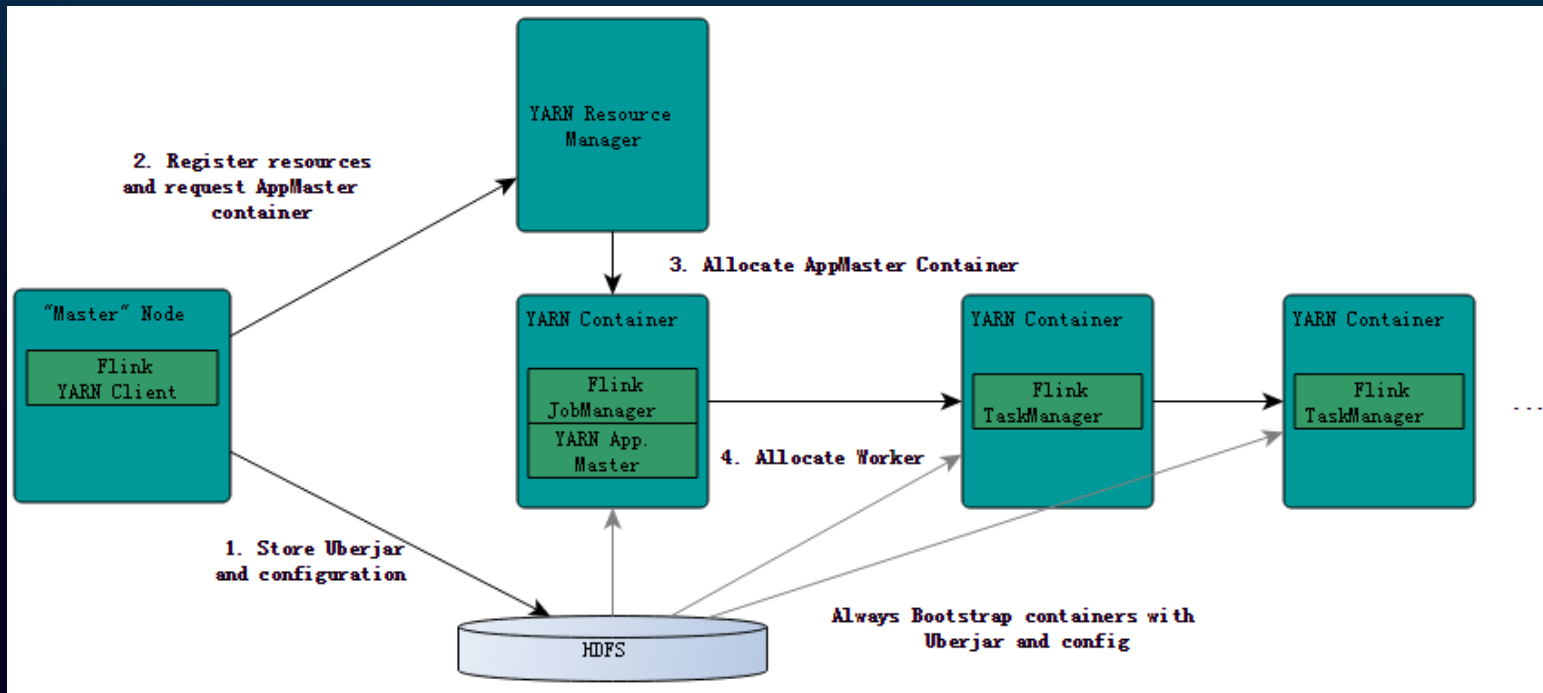


Yarn集群模式



Flink China

- Yarn Single 模式



实时计算平台展示-任务提交



Flink China

- 用户自己申请：容器数，容器大小，slot等
- 用户上传task，指定入口函数
- Flink版本：1.3, 1.5可自选
- 管理员审核

任务管理 / 任务列表

新建任务

* 任务名称:

* 部门:

* 每个container槽位数: 2 默认值2

* Container数量: 1 默认值1

* TaskManager大小(MB): 1000 默认值1000

* JobManager大小(MB): 1000 默认值1000

总资源大小(MB):

槽位总数:

* 任务描述:

* 版本号:

* Flink版本号:

* 入口类路径:

上传任务包:

实时计算平台展示-任务管理



Flink China

任务管理 / 任务列表

程序名称: 状态: 运行 发布者:

程序名称	发布时间	状态	运行时长	发布者	操作
[REDACTED]	2018-08-23 10:26:13	运行	9小时59分5秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-08-08 14:22:12	运行	15天6小时3分6秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-08-17 13:32:40	运行	6天6小时52分38秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-08-16 14:43:05	运行	7天5小时42分13秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-06-20 17:46:17	运行	64天2小时39分1秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-08-13 09:41:47	运行	10天10小时43分31秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-06-13 16:46:38	运行	71天3小时38分40秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-08-20 14:48:04	运行	3天5小时37分14秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-08-02 16:25:10	运行	21天4小时8秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>
[REDACTED]	2018-06-13 16:16:05	运行	71天4小时9分13秒	[REDACTED]	<input type="button" value="查看"/> <input type="button" value="下载"/> <input type="button" value="停止"/> <input type="button" value="升级"/> <input type="button" value="关闭"/> <input type="button" value="启动日志"/>



实时计算平台展示-集群管理



Flink China

Summary No alerts

- MapReduce2
- ZooKeeper
- Ambari Metrics
- SmartSense

Actions

Standby NameNode Started **No alerts**

ZKFailoverController Started **No alerts**

Active NameNode Started **No alerts**

ZKFailoverController Started **No alerts**

DataNodes [redacted]

DataNodes Status [redacted]

JournalNodes [redacted]

NFSGateways [redacted]

NameNode Uptime [redacted]

NameNode Heap [redacted]

Disk Usage (DFS Used) [redacted]

Disk Usage (Non DFS Used) [redacted]

Disk Remaining [redacted]

Blocks (total) [redacted]

Block Errors 0 corrupt replica / 0 missing / 0 under replicated

Total Files + Directories [redacted]

Upgrade Status No pending upgrade

Safe Mode Status Not in safe mode

Metrics

NameNode GC count: 1

NameNode GC time: 1 ms

NN Connection Load: [line graph]

NameNode Heap: 1000 MB

NameNode RPC: 5 ms

Failed disk volumes: 0

Blocks With Corrupted Replicas: 0

Under Replicated Blocks: 0

Group: All-the-machines | Host: [redacted] | Network interface: All

System uptime: 2.16 year

CPU 1 load: 0.9

Free space on /data(%): 35%

Available memory: 2.2 GiB

▼ CPU

CPU

System load

▼ MEMORY

Memory used percent

Linux tcp_con

实时计算平台展示-日志和性能监控

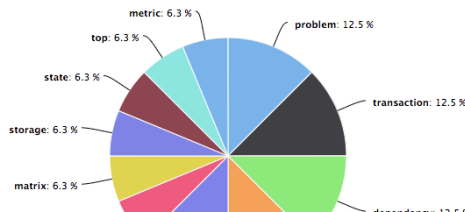


Flink China

The screenshot displays the Kibana interface for monitoring a Flink application. The top navigation bar includes 'Discover', 'Visualize', 'Dashboard', 'Timelion', 'APM', 'Dev Tools', 'Monitoring', and 'Management'. The main area is divided into several sections:

- Discover:** Shows a search query for 'application_id: application_153422580936'. The left sidebar lists 'Selected Fields' (e.g., @timestamp, caller_file_name) and 'Available Fields'.
- Visualize:** A bar chart titled 'ptsjb-logs-*' showing log counts over time. The x-axis represents time from 20:58:30 to 21:00:30, and the y-axis represents the count, peaking at 200.
- Log Viewer:** Displays log entries with their source information. Two entries are visible, both from 'org.apache.flink'.
- Transaction Table:** A table summarizing transaction metrics. The columns include Name, Total, Failure, Failure%, Sample Link, Min(ms), Max(ms), Avg(ms), 95Line(ms), and 99.9Line(ms).

Transaction	Name	Total	Failure	Failure%	Sample Link	Min(ms)	Max(ms)	Avg(ms)	95Line(ms)	99.9Line(ms)
TOTAL	TOTAL	16	0	0.0000%	Log View	6	2,932.3	782.3	-	-
problem	problem	2	0	0.0000%	Log View	711.7	2,932.3	1,822.0	2,500.0	2,500.0
transaction	transaction	2	0	0.0000%	Log View	824.4	1,505.8	1,165.1	1,500.0	1,500.0
heartbeat	heartbeat	1	0	0.0000%	Log View	1,035	1,035	1,035.0	1,000.0	1,000.0
matrix	matrix	1	0	0.0000%	Log View	866.5	866.5	866.5	850.0	850.0
dependency	dependency	2	0	0.0000%	Log View	469.4	1,079	774.2	1,050.0	1,050.0
cross	cross	2	0	0.0000%	Log View	402.3	851.8	627.0	850.0	850.0
storage	storage	1	0	0.0000%	Log View	593.1	593.1	593.1	550.0	550.0
event	event	2	0	0.0000%	Log View	443	557.6	500.3	550.0	550.0
state	state	1	0	0.0000%	Log View	222.3	222.3	222.3	200.0	200.0
top	top	1	0	0.0000%	Log View	16.4	16.4	16.4	16.0	16.0
metric	metric	1	0	0.0000%	Log View	6	6	6.1	6.0	6.0



Glink-Framework框架



Flink China

- 代码框架Glink-Framework

- 简化pom文件
- 三方调用集成：dubbo, zuul
- 三方数据库集成：mysql, redis
- 多环境管理
- 依赖版本管理
- 代码监测工具：checkstyle, pmd, findbugs

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <project xmlns="http://maven.apache.org/POM/4.0.0"
3.         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4.         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http
5.         <modelVersion>4.0.0</modelVersion>
6.
7.         <groupId>glink-framework-demo</groupId>
8.         <artifactId>glink-framework-demo</artifactId>
9.         <version>1.0-SNAPSHOT</version>
0.
1.         <parent>
2.             <groupId>glink-parent</groupId>
3.             <artifactId>glink-parent</artifactId>
4.             <version>0.0.1-SNAPSHOT</version>
5.         </parent>
6.
7.         <dependencies>
8.             <dependency>
9.                 <groupId>glink-core</groupId>
0.                 <artifactId>glink-core</artifactId>
1.                 <version>0.0.1-SNAPSHOT</version>
2.             </dependency>
```



平台化业务BP合作方式



Flink China

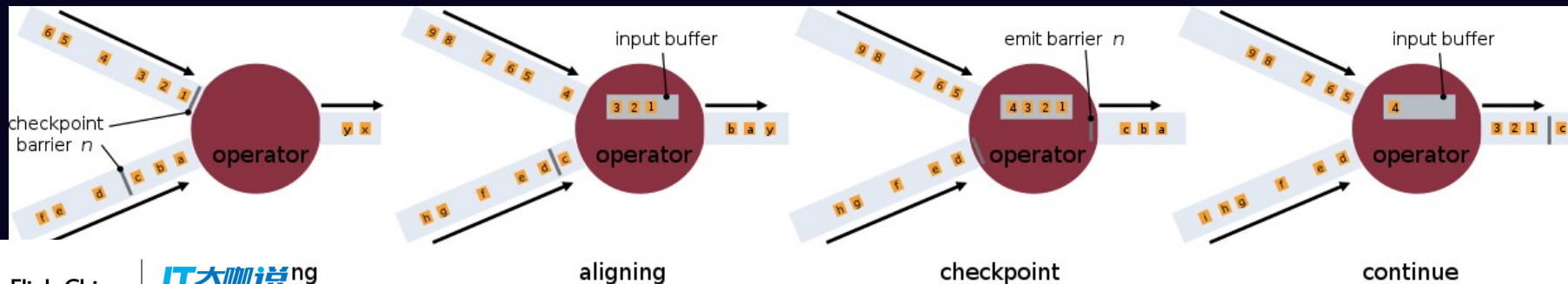
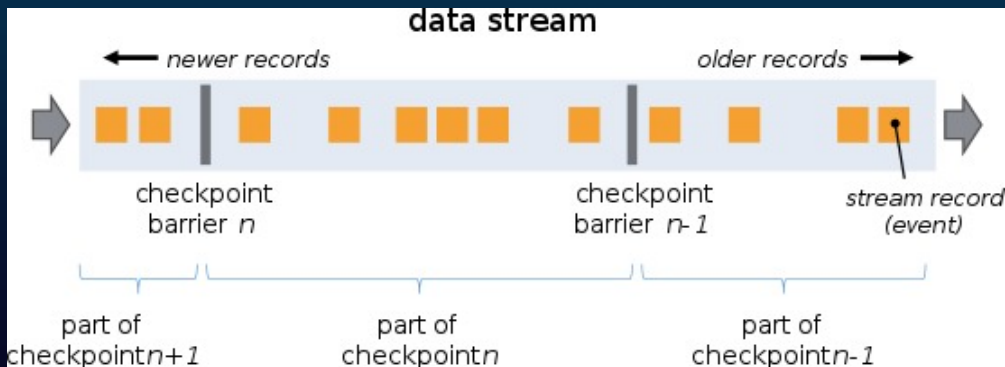


踩坑：barrier对齐造成checkpoint耗时



Flink China

- 有些任务checkpoint的alignment时间需要60s以上





踩过的坑



- 并行度太多造成barrier对齐的花费时间更长，有个并行度28的子任务的对齐时间超过50s



- Valuestate不能跨算子共享



- flink1.3 kafka connector不支持partition增加



- 与spring整合，出现handler匹配的问题
- hadoop的包冲突造成，程序无法正常启动的问题



平台收益

资源利用率提高

集群cpu利用率20%

数据处理量

平台每天处理数据量

80亿条以上



开发效率提升

ETL采集程序从1天开发，
提高到1小时

业务覆盖面广

- 平台上线业务30+，预计年内突破100+
- 服务于公司各条业务线，IoT平台，EMS，FMS，挂车，企业解决方案，SaaS，硬件部门等



Flink China



未来规划



拥抱Flink，降本增效



资源隔离更彻底

CGroup+Node Label



平台易用性

代码发布，debug，监控，一站式解决问题



减少code

Streaming SQL
CEP



通用的脚手架

更多的source、sink，工具，业务封装，简化开发



Flink China

THANKS



zachary

四川 成都



扫一扫上面的二维码图案，加我微信



Flink China

IT大咖说
知识共享平台