Apache Samza: 领英大规模数据流处理的秘籍

(Secret Kung Fu of Massive Scale Stream Processing with Apache Samza @LinkedIn)

刘新宇
Streams Team @ LinkedIn
Committer, Apache Samza

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

StuQueue
斯达克学院

实践驱动的IT教育

斯达克学院(StuQ)， 极客邦旗下实践驱动的IT教育平台。通过线下和线上多种形式的综合学习解决方案 ， 帮助IT从业者和研发团队提升技能水平。

人工智能 | 大数据 | 前端开发 | 后端开发 | 架构设计

移动开发 | 运维设计 | 产品测试 | 产品经理 | 技术管理

10大职业技术领域课程

http://www.stuq.org

TABLE OF
CONTENTS 大纲

- 介绍Samza @ LinkedIn
- Samza大规模流处理的4大秘籍
- 总结展望

Apache Samza

- Apache Samza is a distributed stream and batch processing framework.

Apache Samza 社区

- Apache顶级项目（2014年12月起）
- 7次发布（0.7 ~ 0.13）
- 14 Committers
- 62 Contributors
- 用户：LinkedIn, Uber, MetaMarkets, Netflix, Intuit, TripAdvisor, MobileAware, Optimizely …

Samza @ LinkedIn

- **200 +** 应用
- Production : A **Trillion** Events + ／天
- 单个节点： 1.1M Messages / 秒
- 广泛应用在各个领域，数量在过去两年内指数级增长

Security

Notifications

News classification

Performance monitoring

实时推送通知系统

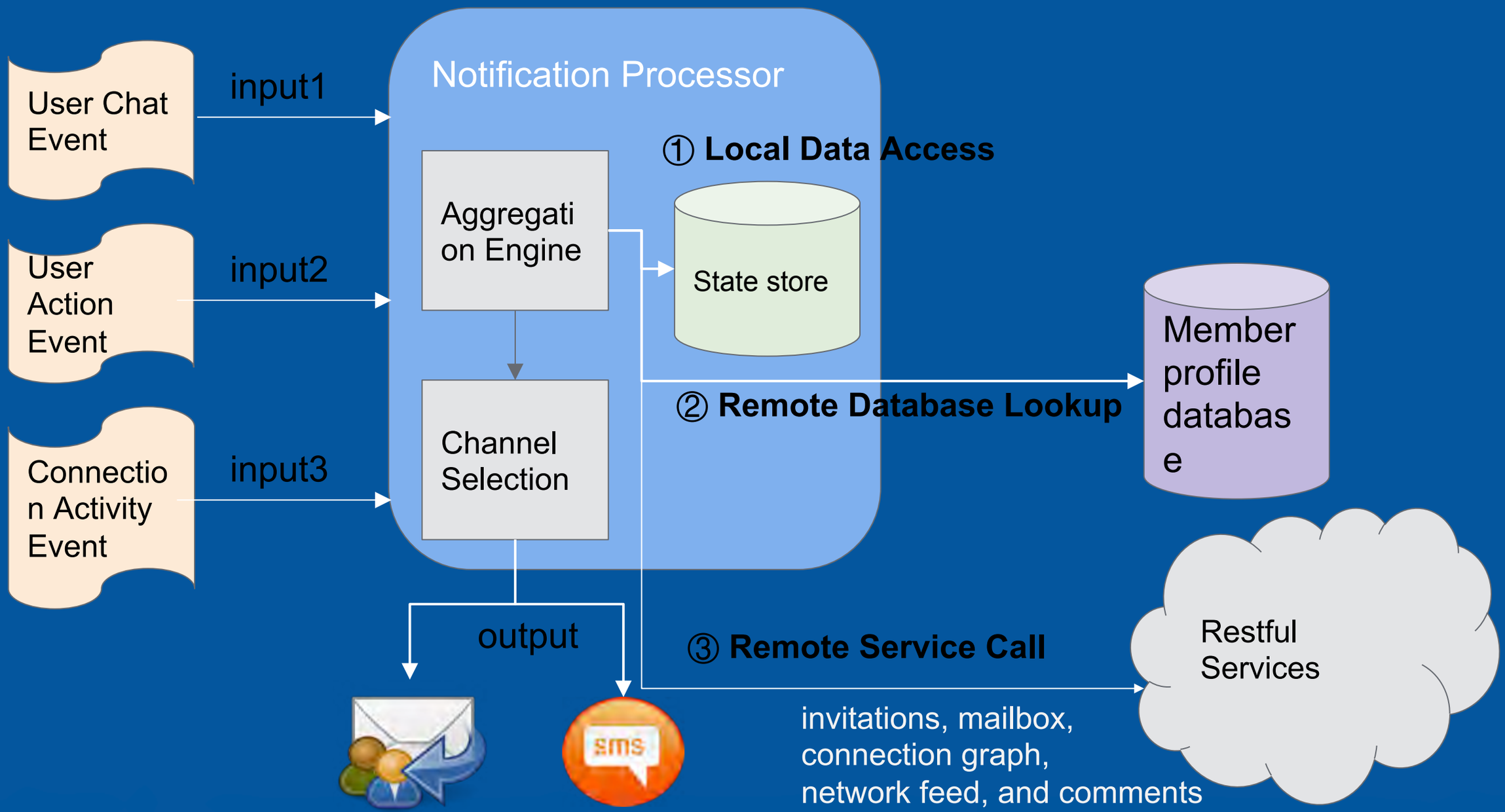


TABLE OF
CONTENTS 大纲

- 介绍Samza @ LinkedIn
- Samza大规模流处理的4大秘籍
- 总结展望

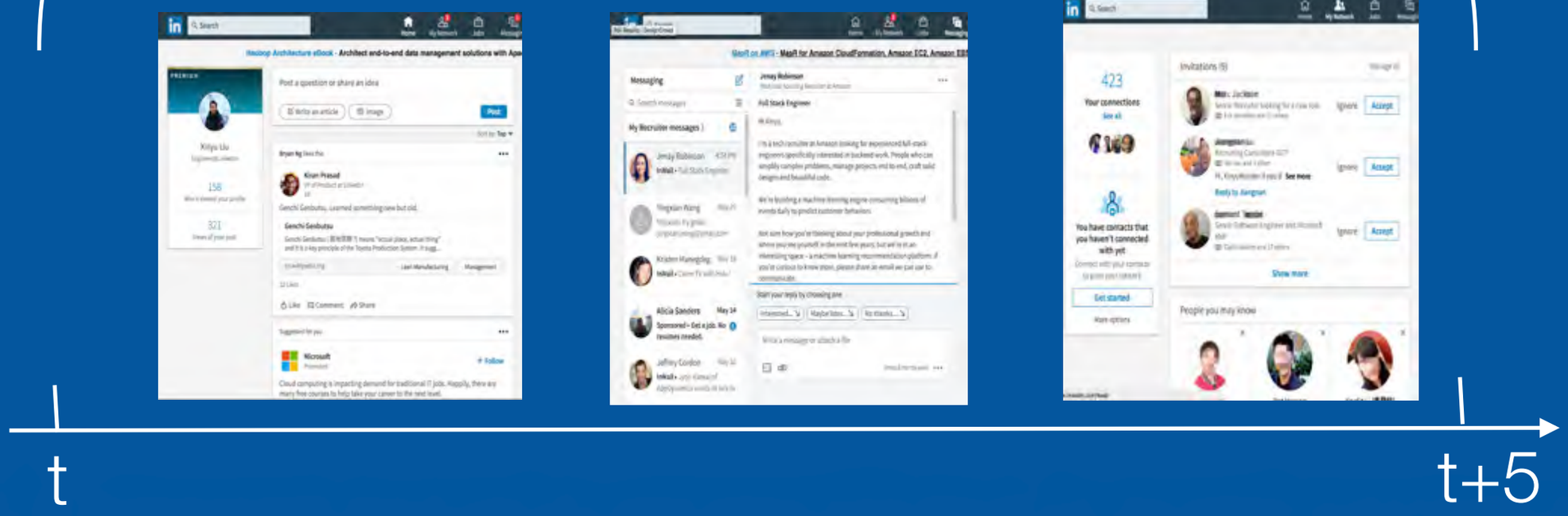
Samza 数据处理秘籍之一

面向数据流的编程模型

High-level API

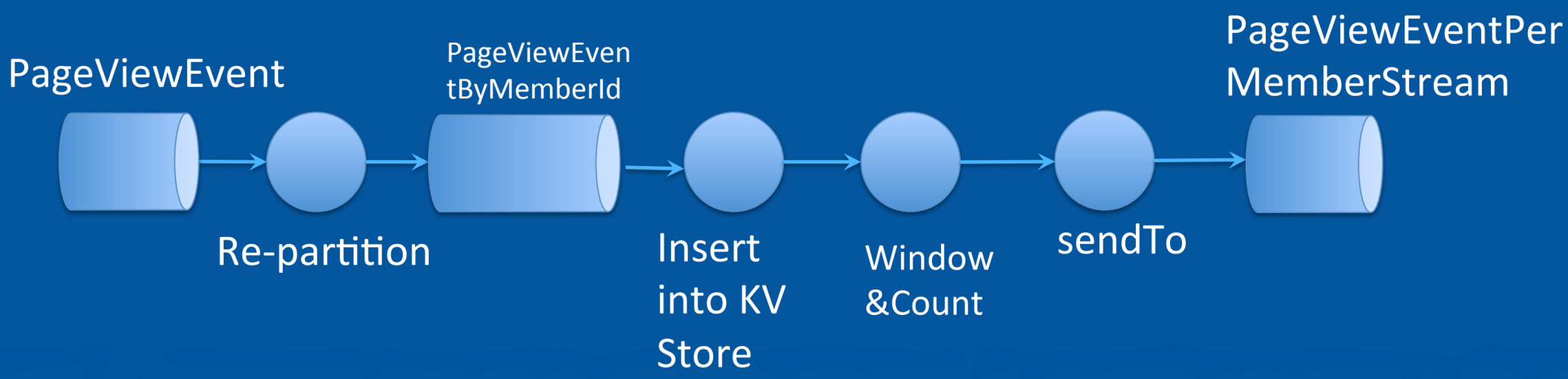
范例分析

- 对于一个PageViewEvent的Kafka数据流（Partitioned by page key），每五分钟统计一次每个用户的event数量，然后发送给另一个Kafka topic.



传统的事件处理编程模型

- 如果使用基本的Event Processing编程，对每个event都需要做如下的工作:
  - 将原PageViewEvent对用户Id进行repartition
  - 在repartition后，每个Event都根据key = (timestamp, memberId) 写入一个key-value store.
  - 当5分钟window timer到来时，对这个kv store进行过去五分钟的range query, 对这五分钟内出现的所有用户和Pageview进行统计
  - 统计结果发送到另一个Kafka topic
- 此编程模型效率低，程序冗长，容易出错，可维护性差。



数据流编程模型

```
public class RepartitionAndCounterExample implements StreamApplication {

@Override public void init(StreamGraph graph, Config config) {
 Supplier<Integer> initialValue = () -> 0;
 MessageStream<PageViewEvent> pageViewEvents =
    graph.getInputStream("pageViewEventStream", (k, m) -> (PageViewEvent) m);
 OutputStream<String, MyStreamOutput, MyStreamOutput> pageViewEventPerMemberStream = graph
    .getOutputStream("pageViewEventPerMemberStream", m -> m.memberId, m -> m);

  pageViewEvents
    .partitionBy(m -> m.memberId)
    .window(Windows.keyedTumblingWindow (m -> m.memberId, Duration.ofMinutes(5), initialValue, (m,
c) -> c + 1))
    .map(MyStreamOutput::new)
    .sendTo(pageViewEventPerMemberStream);
  }
}
```

运行可视化工具

- 实例可视化链接

Samza Operators

stateless functions	filter	select a subset of messages from the stream
	map	map one input message to an output message
	flatMap	map one input message to 0 or more output messages
	merge	union all inputs into a single output stream
I/O functions	partitionBy	re-partition the input messages based on a specific field
	sendTo	send the result to an output stream
	sink	send the result to an external system (e.g. external DB)
stateful functions	window	window aggregation on the input stream
	join	join messages from two input streams

Samza 数据处理秘籍之二

可扩展的数据存取

scalable data access

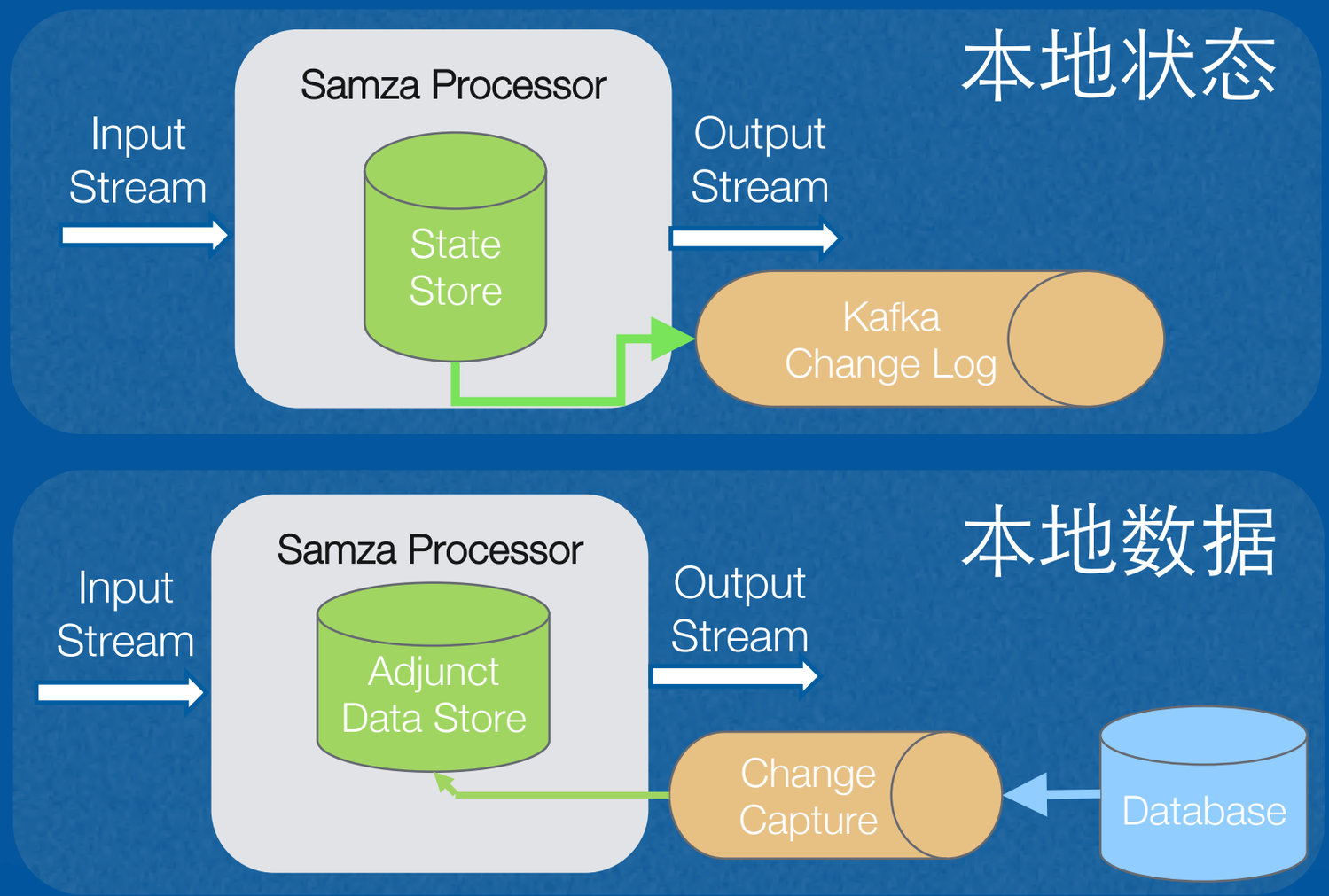
范例分析

- 在上面的PageViewEvent统计实例中，用户需要

1. 保存统计的中间结果以便故障恢复
2. 读取远程的用户数据信息

本地数据存取

- Samza提供基于内存或RocksDb的Key-Value Store 用于高速本地数据存取



1.2TB
State

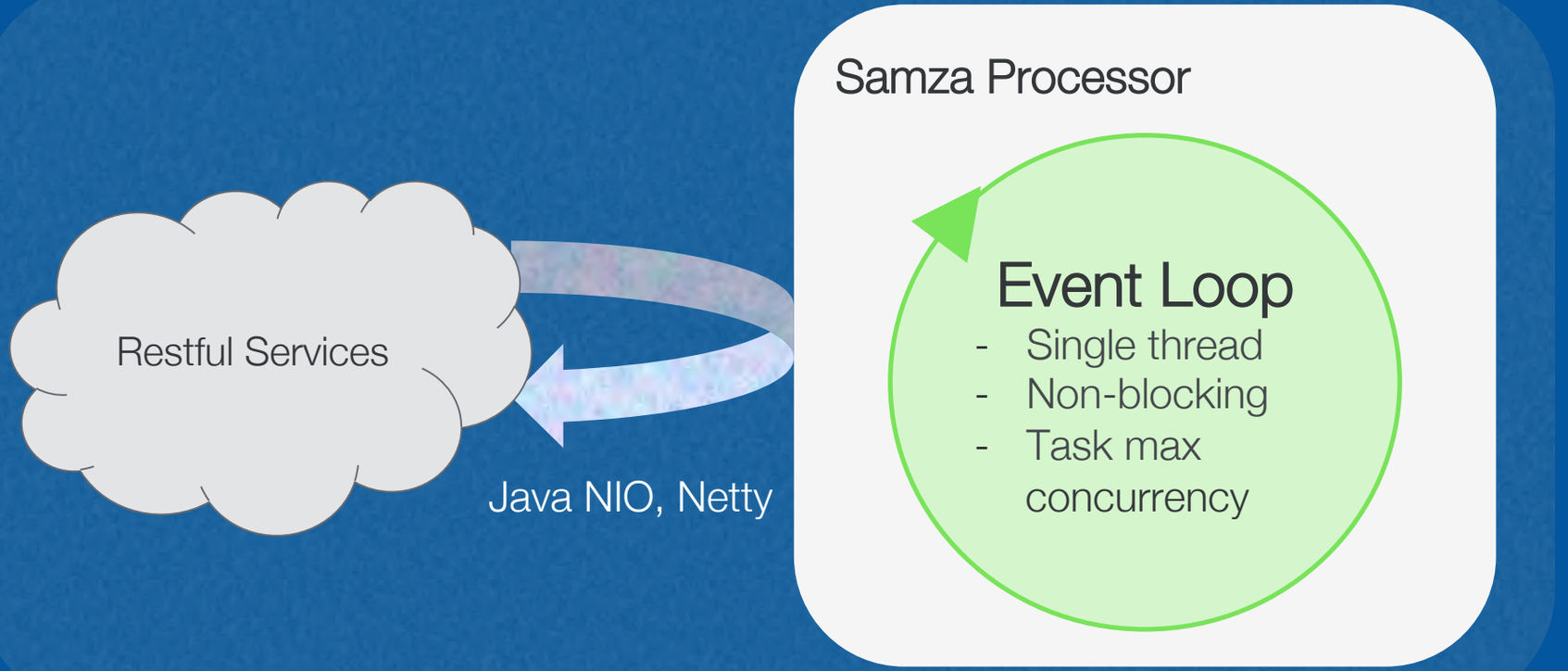
100x
Faster

1.1M
TPS
On a single machine

60x
Faster
than bootstrap

远程数据存取

- Samza支持Native异步数据处理
- Samza提供multi-threaded同步数据处理



100x
Parallelism

Samza 数据处理秘籍之三

统一的流处理和批处理

Unified Stream & Batch Processing

范例分析

- Kafka提供实时的PageViewEvent，同样的数据也被存储到了Hadoop HDFS。用户需要随时选择不同的数据源进行处理，那需要两套完全不同的处理流程吗?

统一的流处理和批处理

- Samza应用不需要任何程序流程的改变，只需要在Configuration里修改数据源。

Kafka 数据源

```
streams.pageViewEventStream.system=kafka
streams.pageViewEventStream.physical.name=PageViewEvent
systems.kafka.samza.factory=org.apache.samza.system.kafka.KafkaSystemFactory
systems.kafka.consumer.zookeeper.connect=localhost:2181/
systems.kafka.producer.bootstrap.servers=localhost:9092
```

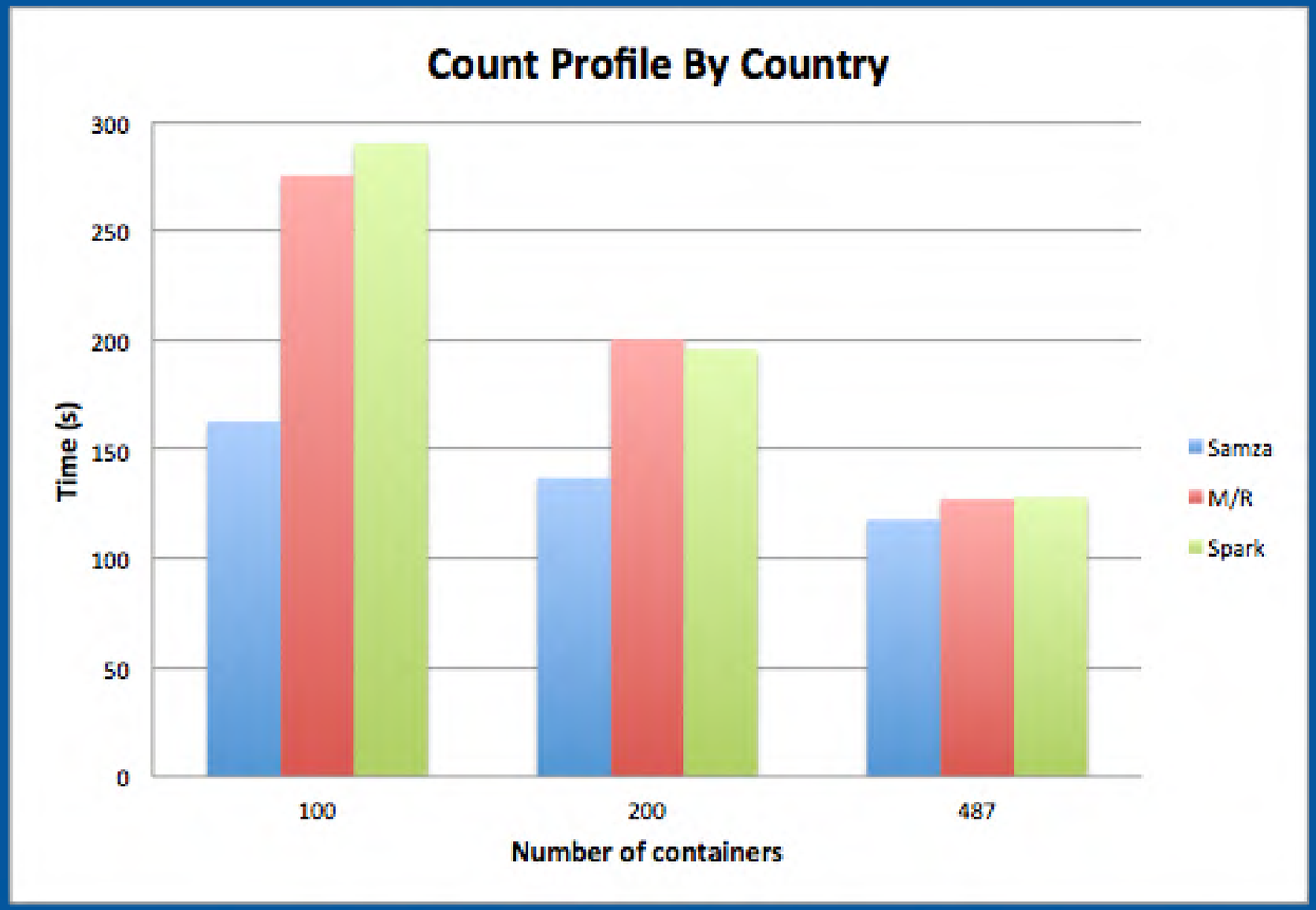
HDFS 数据源

```
streams.pageViewEventStream.system=hdfs
streams.pageViewEventStream.physical.name=hdfs://mydbsnapshot/PageViewEvent/
systems.hdfs.samza.factory=org.apache.samza.system.hdfs.HdfsSystemFactory
```

性能比较

- 统计各国用户人数：通过统计，得到用户人数最多的N个国家

Test data: member profile (242 GB, 487 files, around 450 million records)



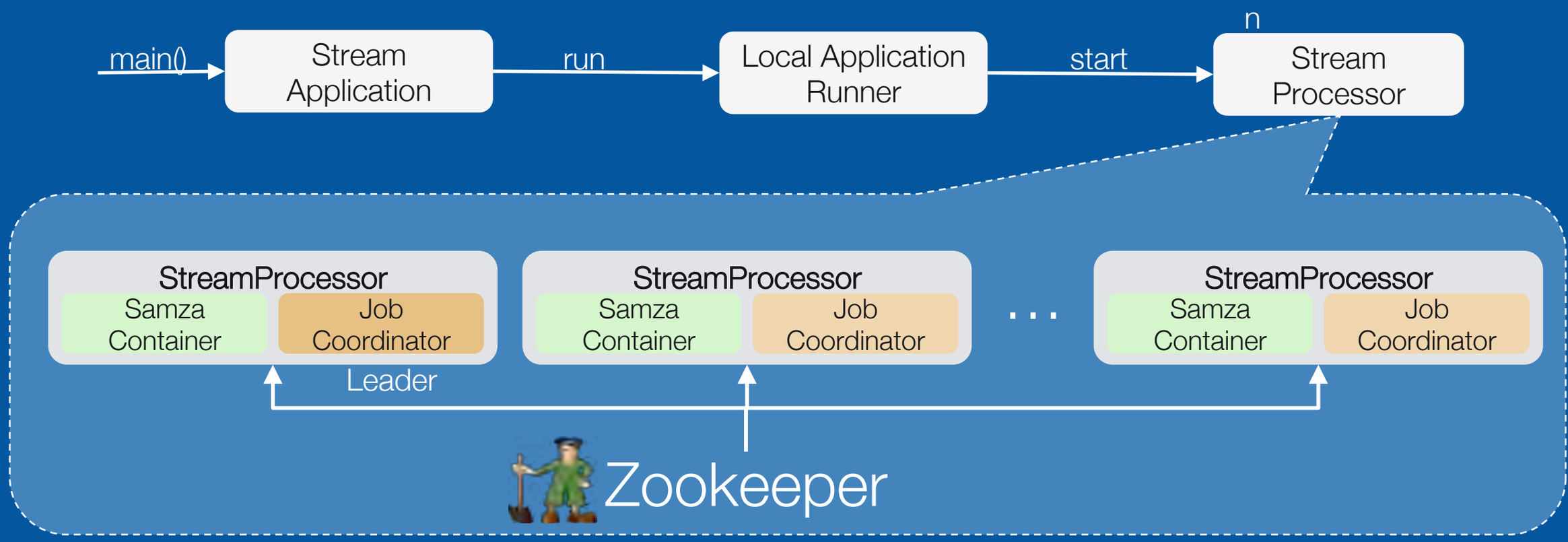
Samza 数据处理秘籍之四
灵活的部署方式
Flexible deployment models

范例分析

- 在实际中，用户运行数据处理的机群是多种多样的，怎么才能够在不同的机群结构里运行同样的Samza应用?

Samza-as-a-library 部署

- Samza集成在用户程序当中，用户完全掌握自己的流处理，如应用的生态周期和资源的分配和管理



Samza in a Cluster 部署

- Samza可以Native运行在主流的Cluster上， 如Yarn和Mesos (in progress)
- 使用Cluster进行部署，协调和资源管理，自动的故障恢复

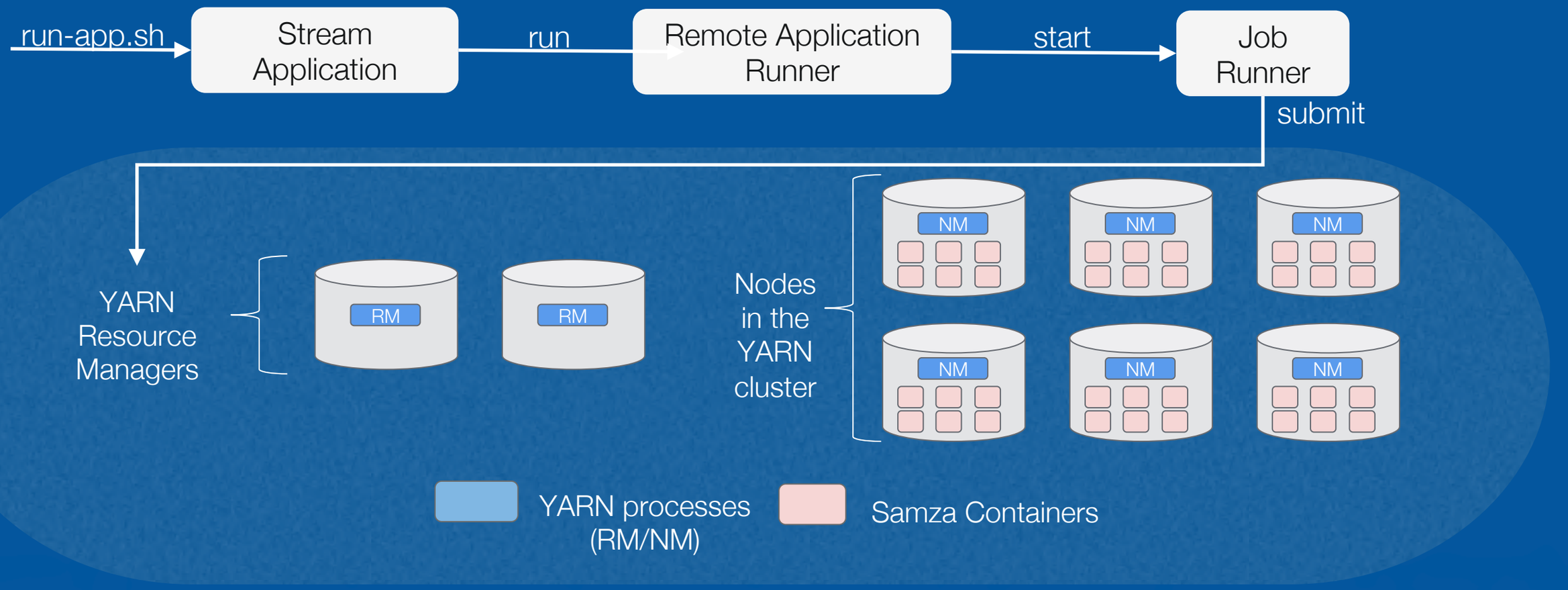
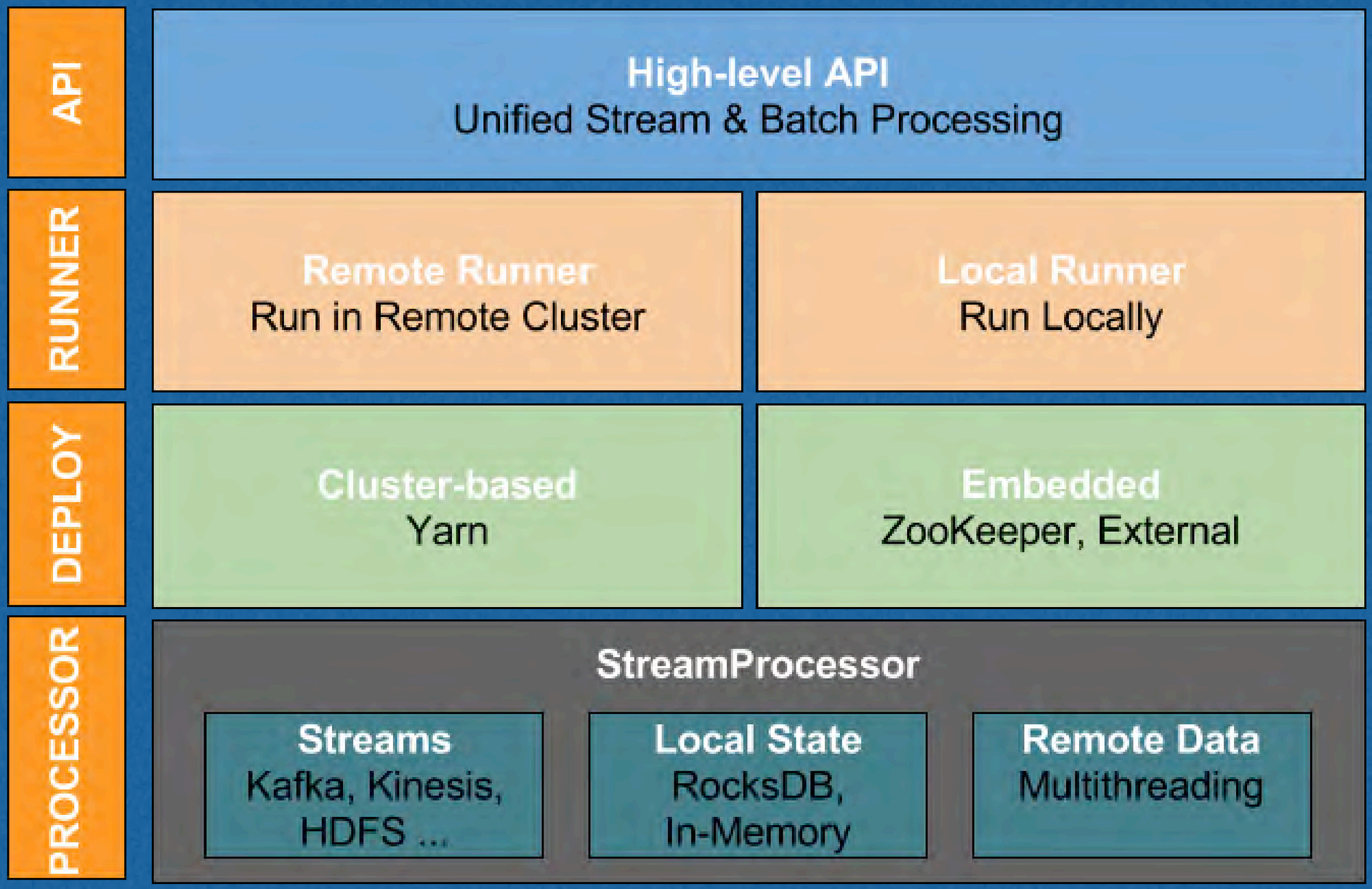


TABLE OF
CONTENTS 大纲

- 介绍Samza @ LinkedIn
- Samza大规模流处理的4大秘籍
- 总结展望

Samza系统结构


API
High-level API
Unified Stream & Batch Processing
RUNNER
Remote Runner
Run in Remote Cluster
Local Runner
Run Locally
DEPLOY
Cluster-based
Yarn
Embedded
ZooKeeper, External
PROCESSOR
StreamProcessor
Streams
Kafka, Kinesis, HDFS ...
Local State
RocksDB, In-Memory
Remote Data
Multithreading

Samza未来展望

- Samza runner for Apache Beam
- Event-time Processing
- Exactly-once Processing
- Table API with Adjunct Dataset
- SQL

THANKS!

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