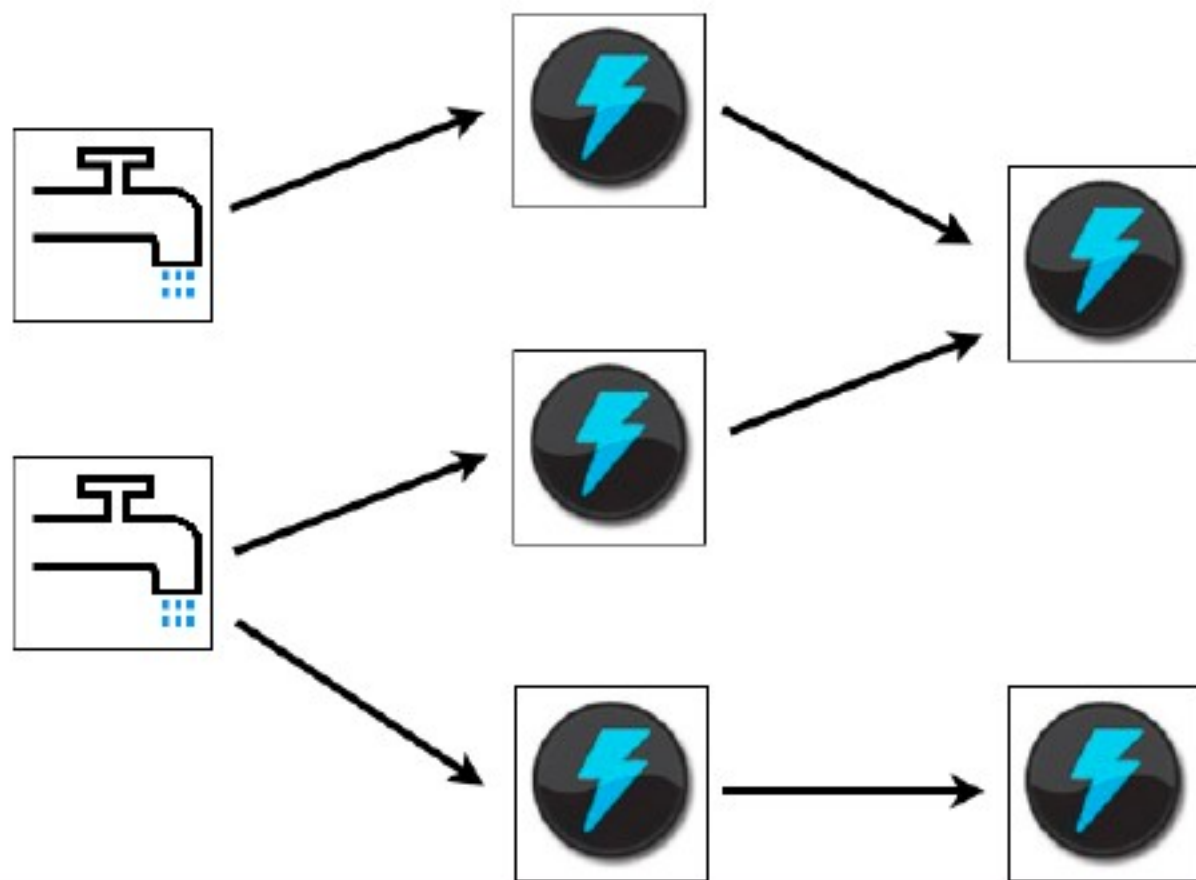


Storm基本概念和组件介绍

Storm是什么？

Apache Storm (<http://storm.apache.org/>) 是由Twitter开源的分布式实时计算系统。Storm可以非常容易的可靠地处理无限的数据流。对比Hadoop的批处理，Storm是个实时的、分布式以及具备高容错的计算系统。Storm可以使用任何编程语言来进行开发



Storm基本概念

- Storm的集群表面上看和Hadoop的集群非常像。但是在Hadoop上面你运行的是MapReduce的Job，而在Storm上面你运行的是Topology。Storm和Hadoop一个非常关键的区别是Hadoop的MapReduce Job最终会结束，而Storm的Topology会一直运行（除非你显式地杀掉它）。

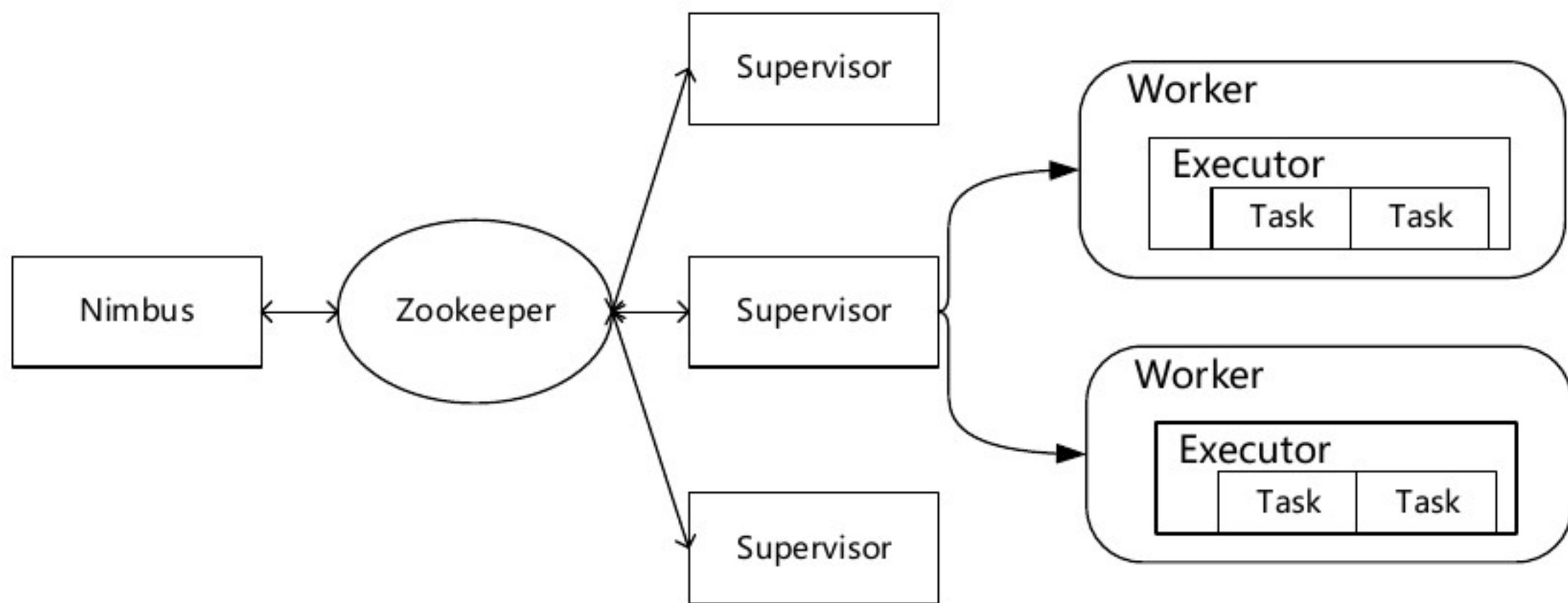
	Storm	Hadoop
角色	Nimbus	JobTracker/Resourcemanager
	Supervisor	TaskTracker/Nodemanager
	Worker	Child/Task
应用名称	Topology	Job
编程接口	Spout/Bolt	Mapper/Reducer

Storm基本概念

Topology	一个实时计算应用程序逻辑被封装在Topology对象中，类似Hadoop中的job，Topology会一直运行直到你显式杀死它。
Nimbus	负责资源分配和任务调度，类似Hadoop中的JobTracker。
Supervisor	负责接受Nimbus分配的任务，启动和停止自己管理的Worker进程
Worker	运行具体处理组件逻辑的进程。
Executor	Executor为Worker进程中的具体的物理线程，同一个Spout/Bolt的Task可能会共享一个物理线程
Task	每一个Spout/Bolt具体要干的活，各个节点之间进行Grouping的单位。
Tuple	消息传递的基本单元。
Stream	源源不断传递的Tuple就组成了Stream。

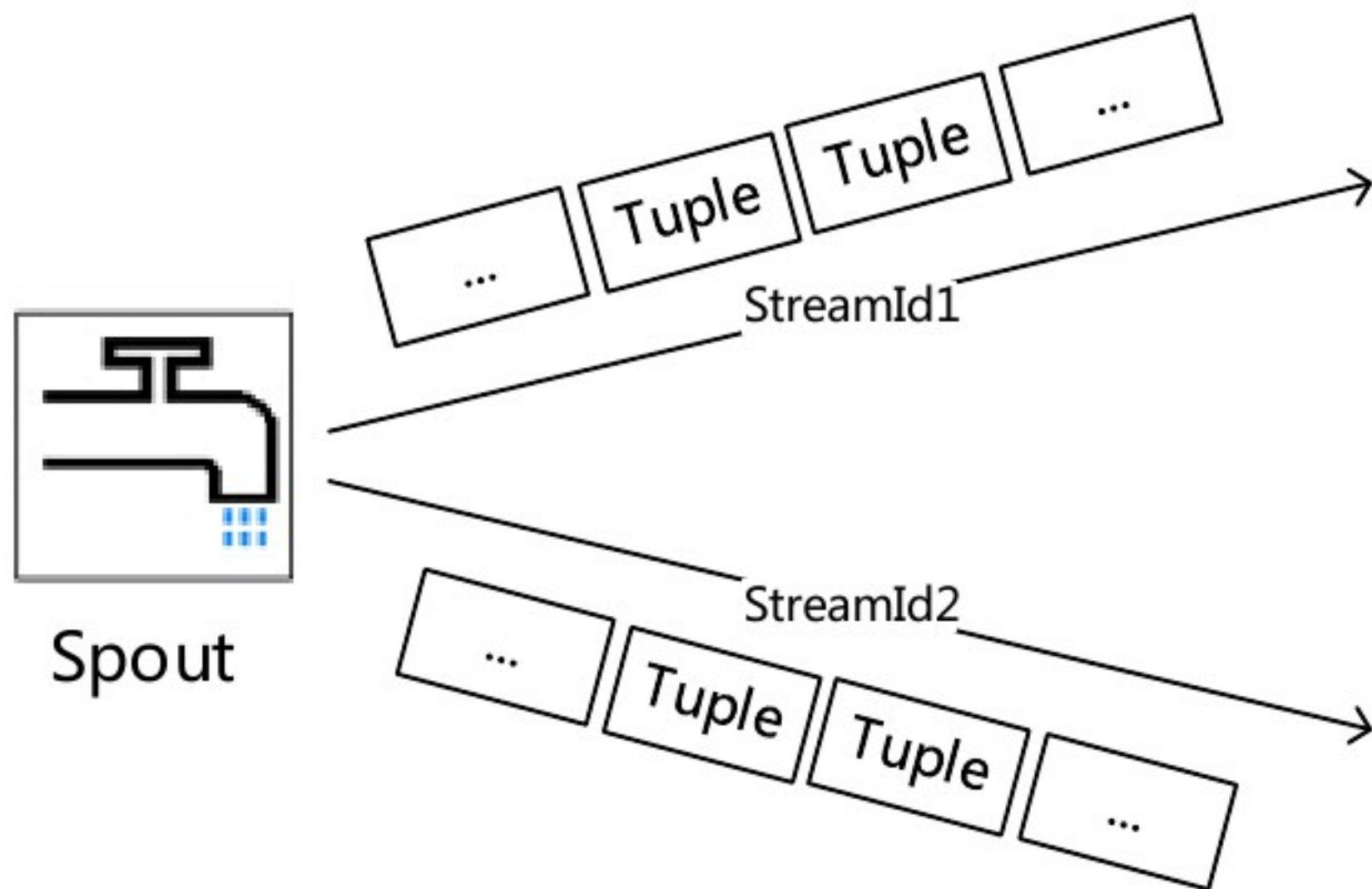
Storm体系架构

- Nimbus和Supervisor之间的通讯是依靠ZooKeeper来完成。并且Nimbus进程和Supervisor都是快速失败 (fail-fast)和无状态的。可以用kill -9来杀死Nimbus和Supervisor进程，然后再重启它们，它们可以继续工作



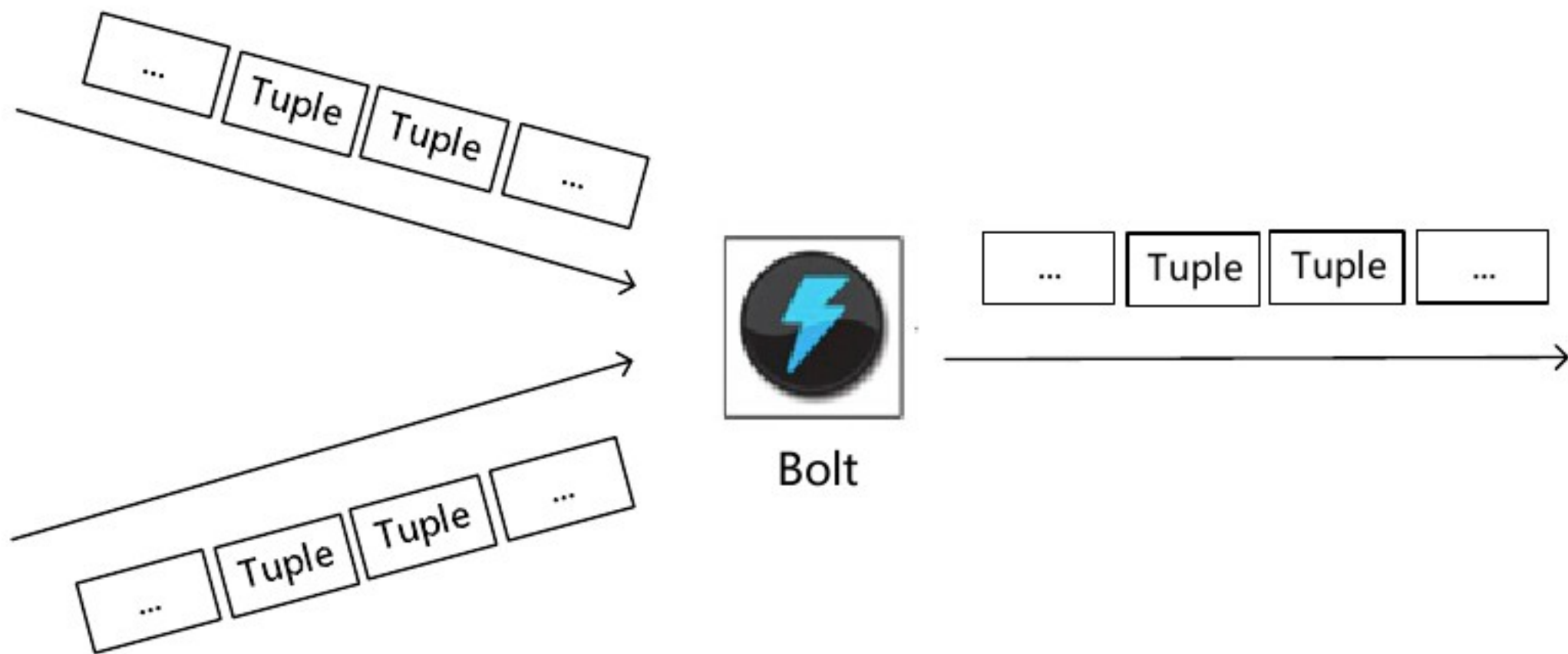
Storm基本概念-Spout

Spout: 在Topology中产生源数据流的组件。通常Spout获取从Kafka、MQ等的的数据，然后调用nextTuple函数，发射数据出去供Bolt消费。



Storm基本概念-Bolt

Bolt: 在Topology中接受Spout的数据然后执行处理的组件。
Bolt在接收到消息后会调用execute函数，用户可以在其中执行自己想要的操作。



Storm基本概念- Grouping

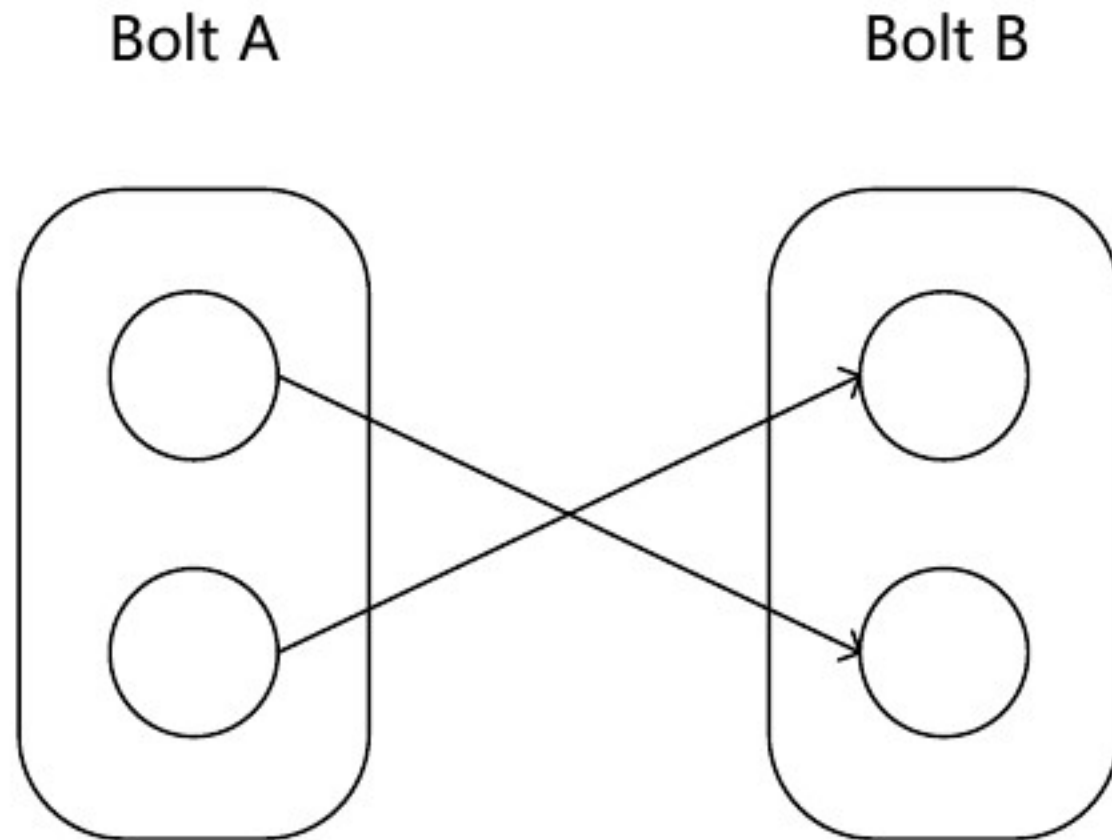
Grouping

- 即消息的partition方法，包括：shuffle、fields、all、global、none、direct和localOrShuffle等

Storm基本概念- Grouping

Shuffle grouping

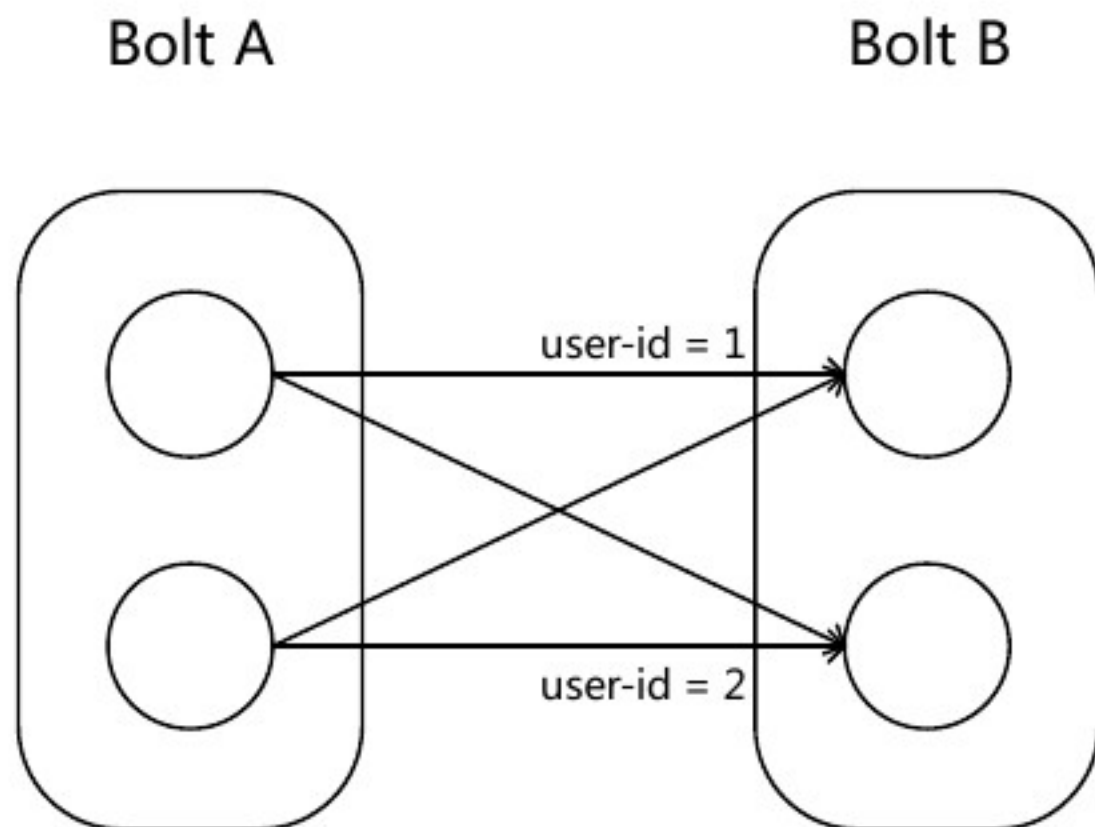
- Task中的数据随机分配，可以保证每个同一级的Bolt上的每个Task处理的Tuple数量一致



Storm基本概念- Grouping

Fields grouping

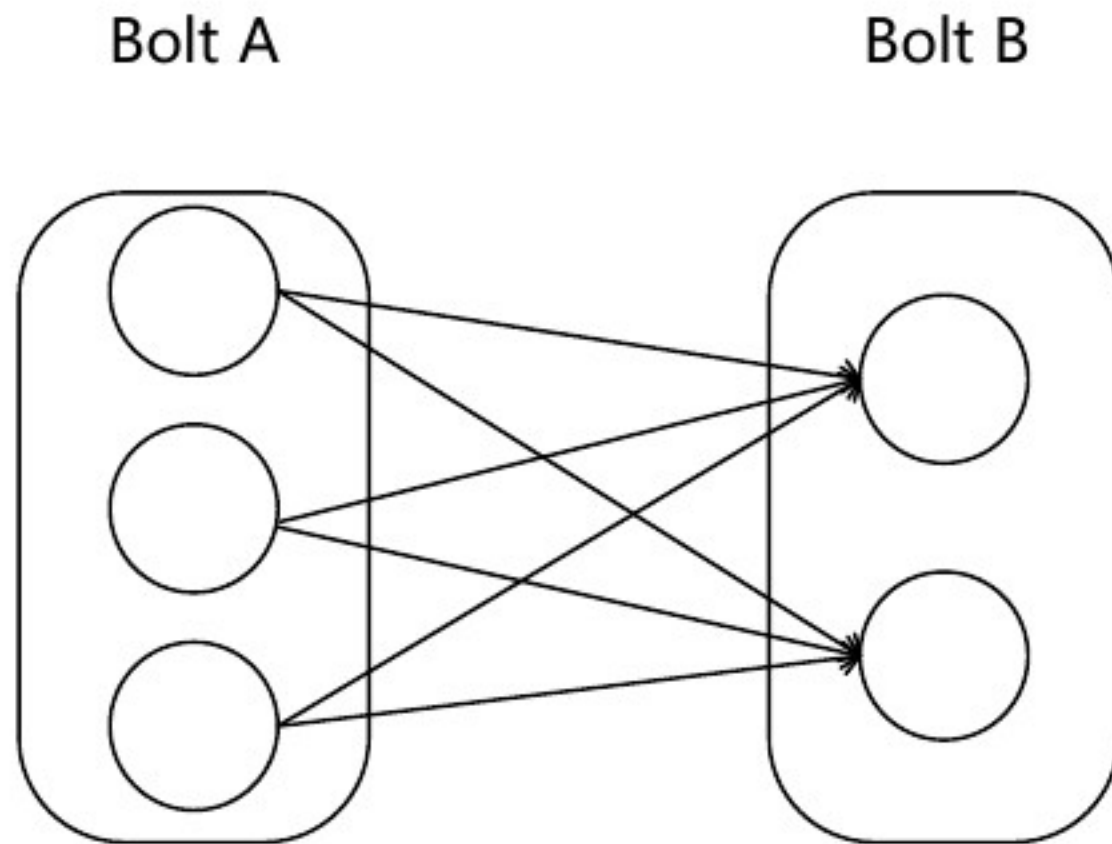
- 根据Tuple中的某一个Field或者多个Field的值来划分。比如Stream根据Field为user-id来grouping，相同user-id值的Tuple会被分发到相同的Task中



Storm基本概念- Grouping

All grouping:

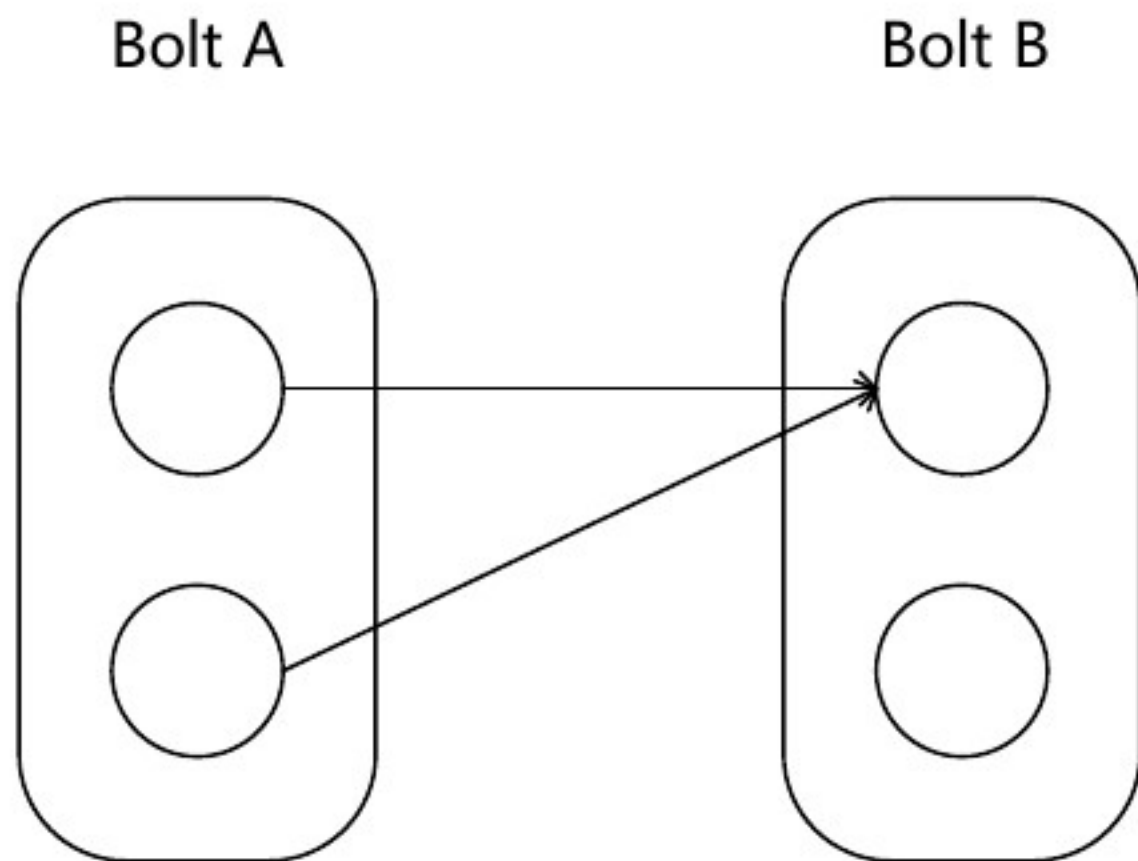
- 所有的Tuple都会到分发到所有的Task上



Storm基本概念- Grouping

Global grouping

- 整个Stream会选择一个Task作为分发的目的地，通常是最新的那个id的Task



Storm基本概念- Grouping

None grouping

- 也就是你不关心如何在Task中做Stream的分发，目前等同于 Shuffle grouping

Direct grouping

- 产生数据的Spout/Bolt自己明确决定这个Tuple被Bolt的那些Task所消费

Local or shuffle grouping

- 如果目标Bolt中的一个或者多个Task和当前产生数据的Task在同一个Worker进程里面，那么就走内部的线程间通信，将Tuple直接发给在当前Worker进程的的目的Task。否则，同Shuffle grouping

Storm优点

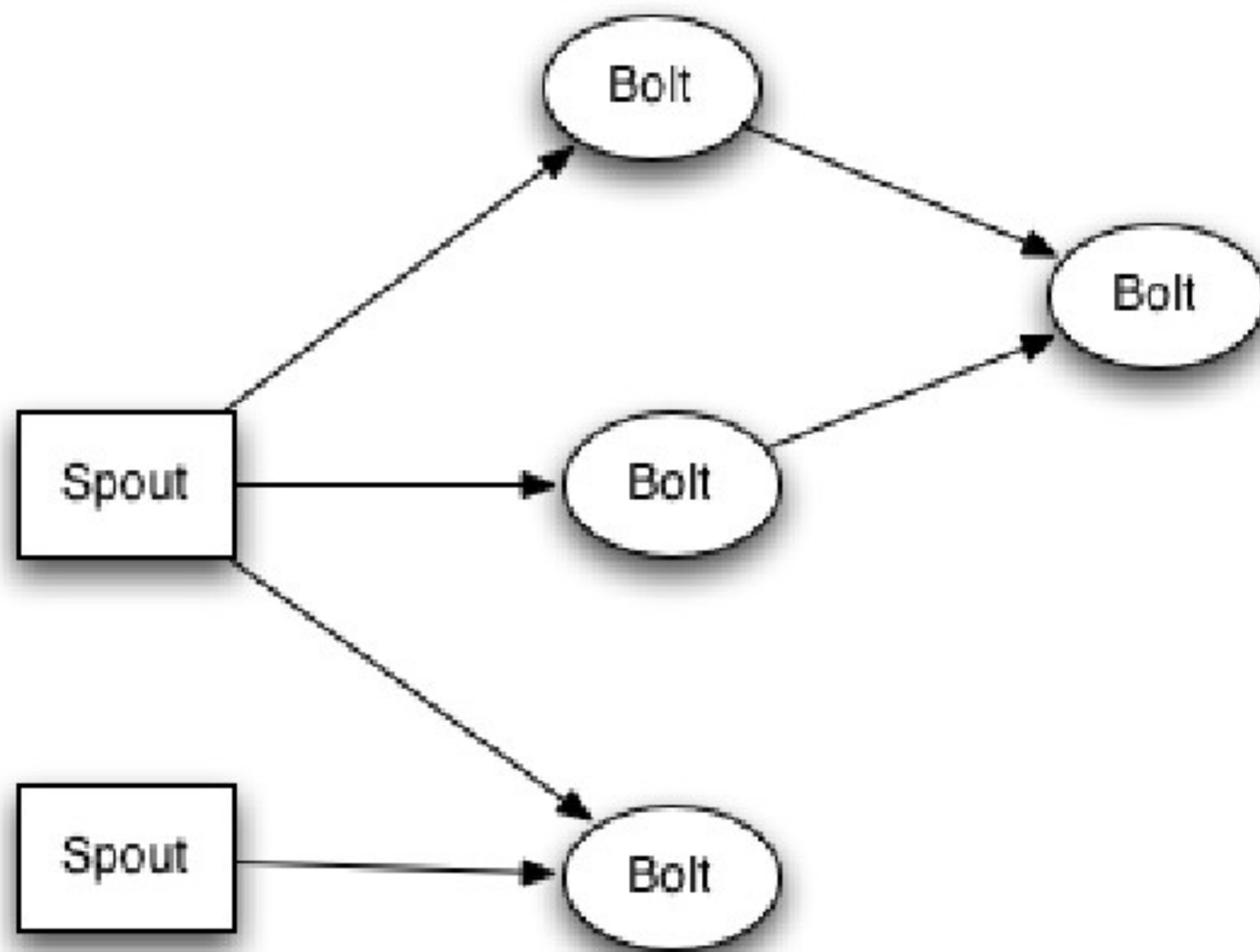
易用性

- 只要遵守Topology, Spout, Bolt的编程规范即可开发出一个扩展性极好的应用，底层RPC, Worker之间冗余，数据分流之类的操作，开发者完全不用考虑。

Storm优点

扩展性

- 当某一级处理单元速度不够，直接配置一下并发数，即可线性扩展性能



Storm优点

健壮性

- 当Worker失效或机器出现故障时，自动分配新的Worker替换失效Worker

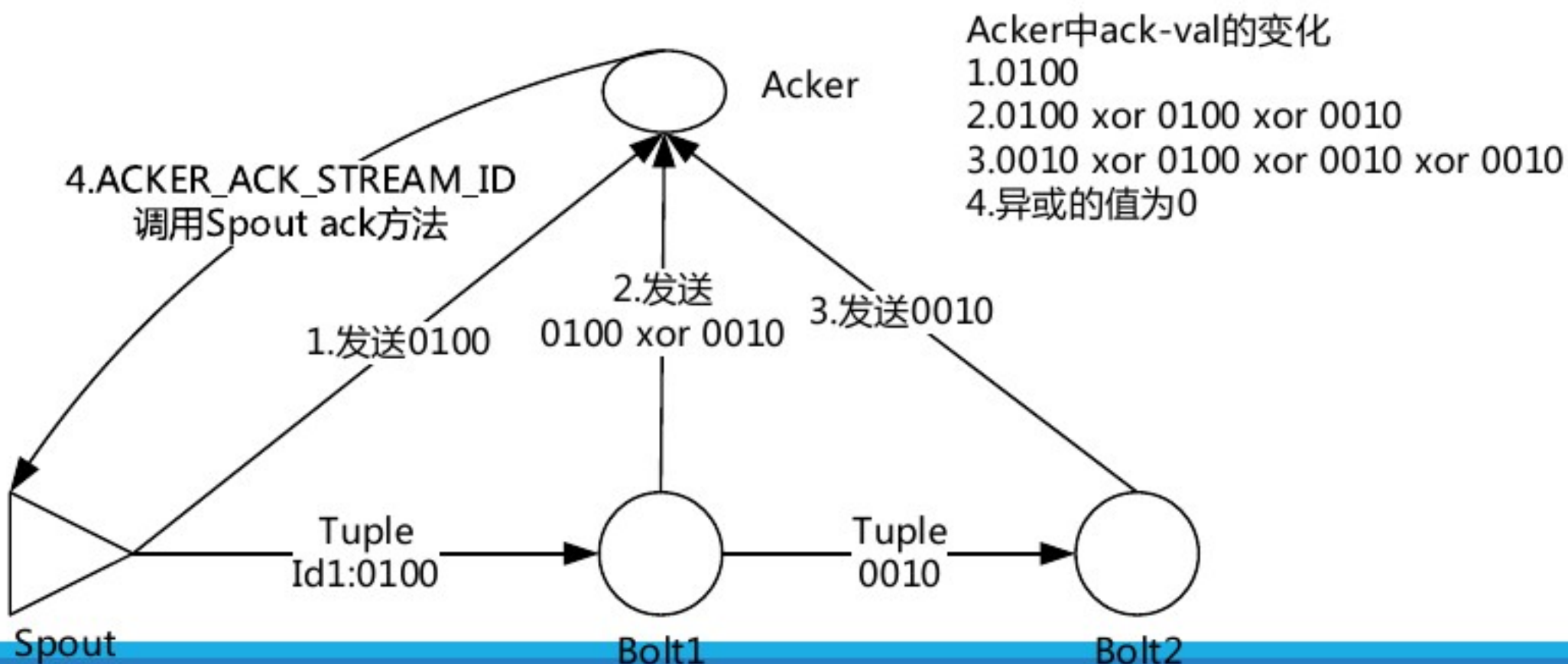
准确性

- 采用Acker机制，保证数据不丢失
- 采用事务机制，保证数据准确性

Ack机制

Storm Ack框架的一个非常大的创新

- 在工作过程中，不保存整棵Tuple树的映射，对于任意大的一个Tuple树，它只需要恒定的20字节就可以进行跟踪，这样大大节省了内存



Storm架构回顾

