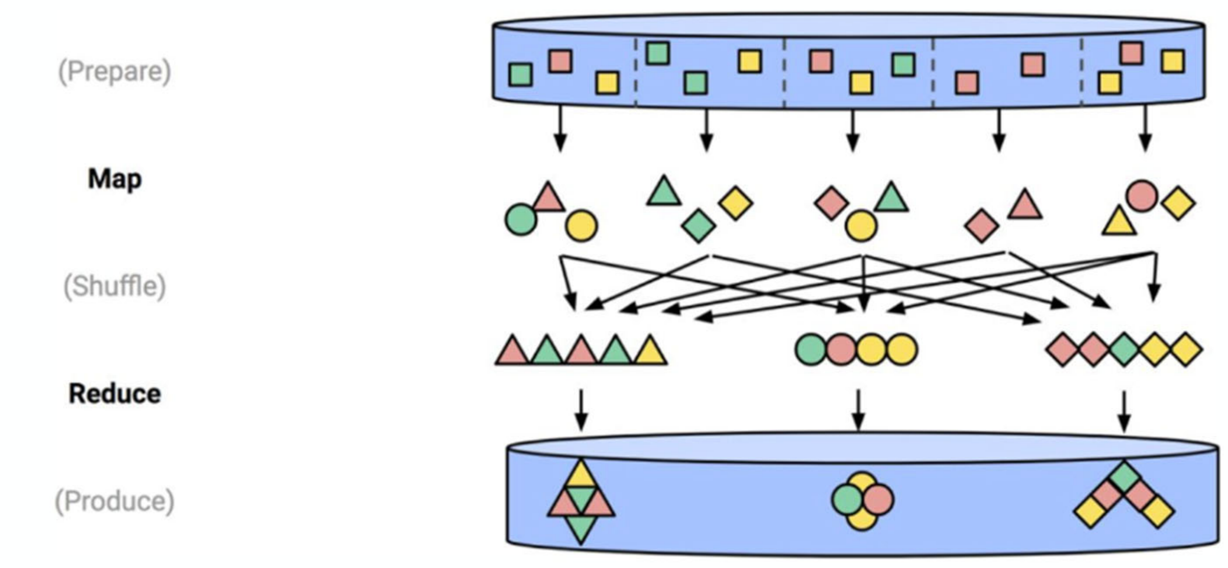
Micro-Batch, Streaming and Serveless

@Luminous Moonlight

Presented by KONY128

2020-11-21

Map Reduce Review



Google MapReduce https://research.google/pubs/pub62/

RDD Review

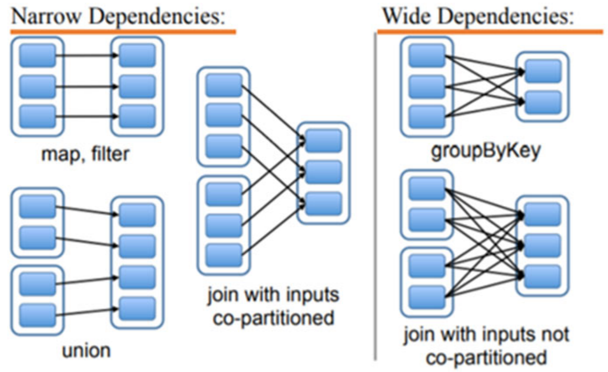
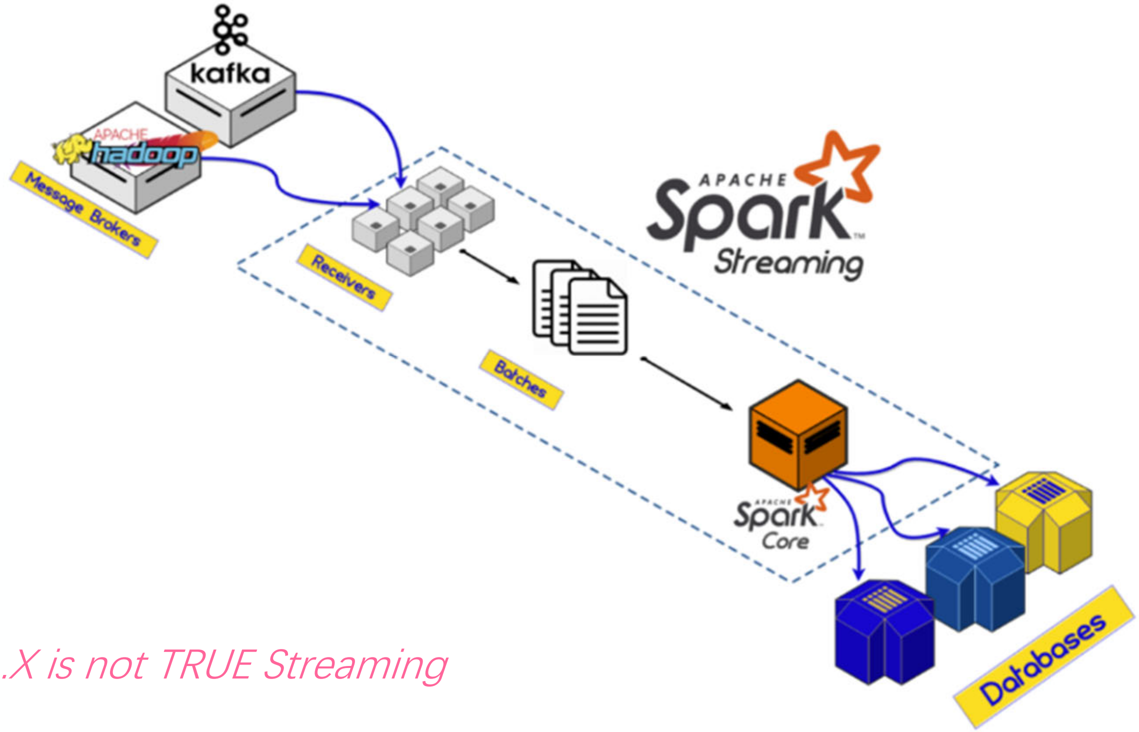

Narrow Dependencies:
map, filter
union
join with inputs co-partitioned
Wide Dependencies:
groupByKey
join with inputs not co-partitioned

Figure 4: Examples of narrow and wide dependencies. Each box is an RDD, with partitions shown as shaded rectangles.

No Scheduler Fault-Tolerance

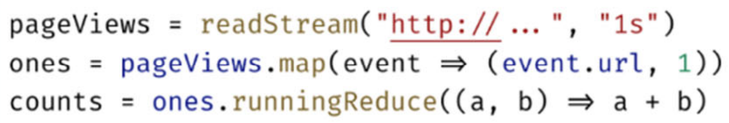
Resilient Distributed Datasets:
https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf

Spark Streaming

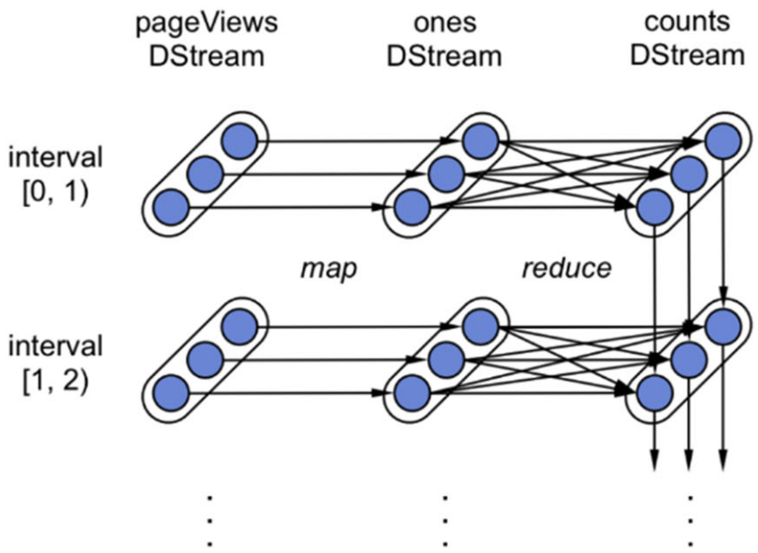


Spark Streaming 1.X is not TRUE Streaming

Discretized Streams



```
pageViews = readStream("http:// ... ", "1s")
ones = pageViews.map(event ⇒ (event.url, 1))
counts = ones.runningReduce((a, b) ⇒ a + b)
```



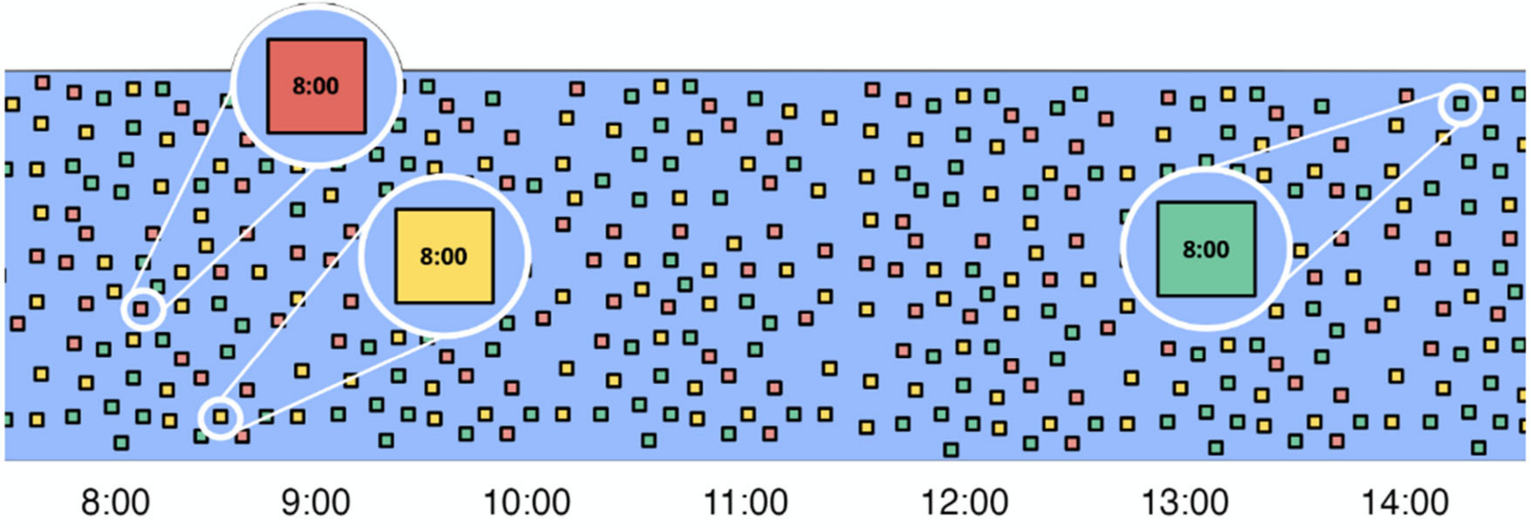
https://dl. acm.org/doi/pdf/10.1145/2517349.2522737

The Dataflow Model

A Practical Approach to Balancing Correctness, Latency, and Cost in Massive- scale, Unbounded, Out-of-order Data Processing

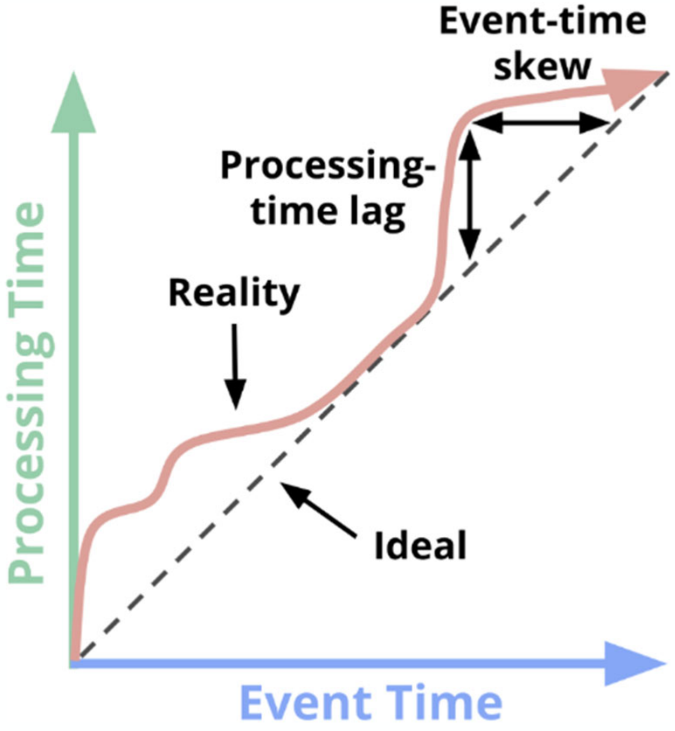
https://research.google/pubs/pub43864/

Data Distribution Of Streaming

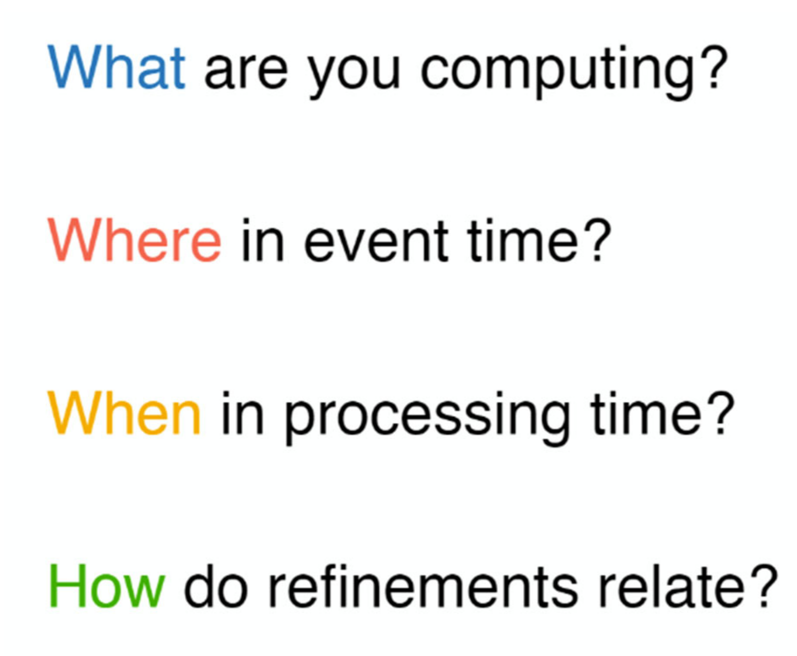


Data can be infinitely big with unknown delays.

Time Distribution Of Events



Focus



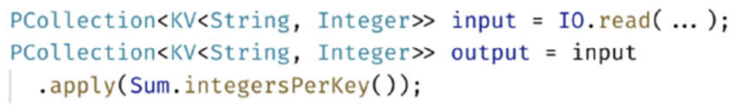
What are you computing?

Where in event time?

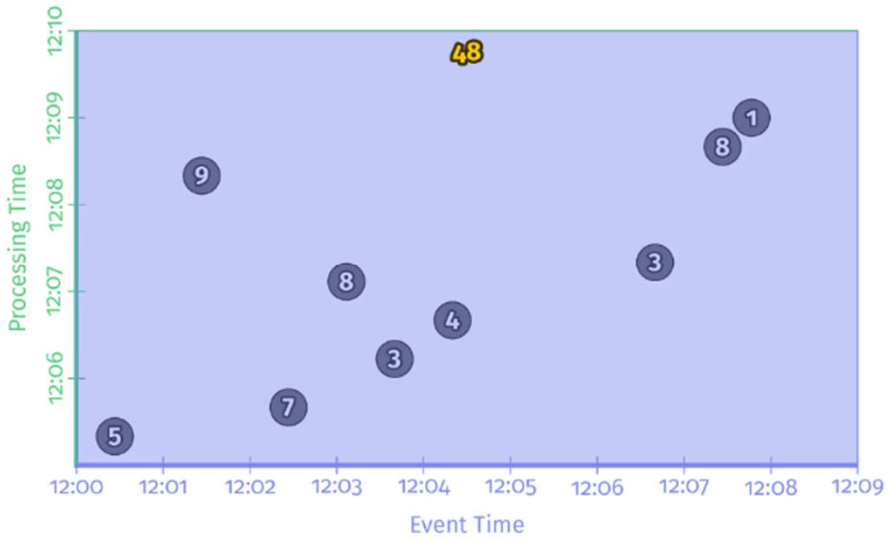
When in processing time?

How do refinements relate?

What Are You Computing

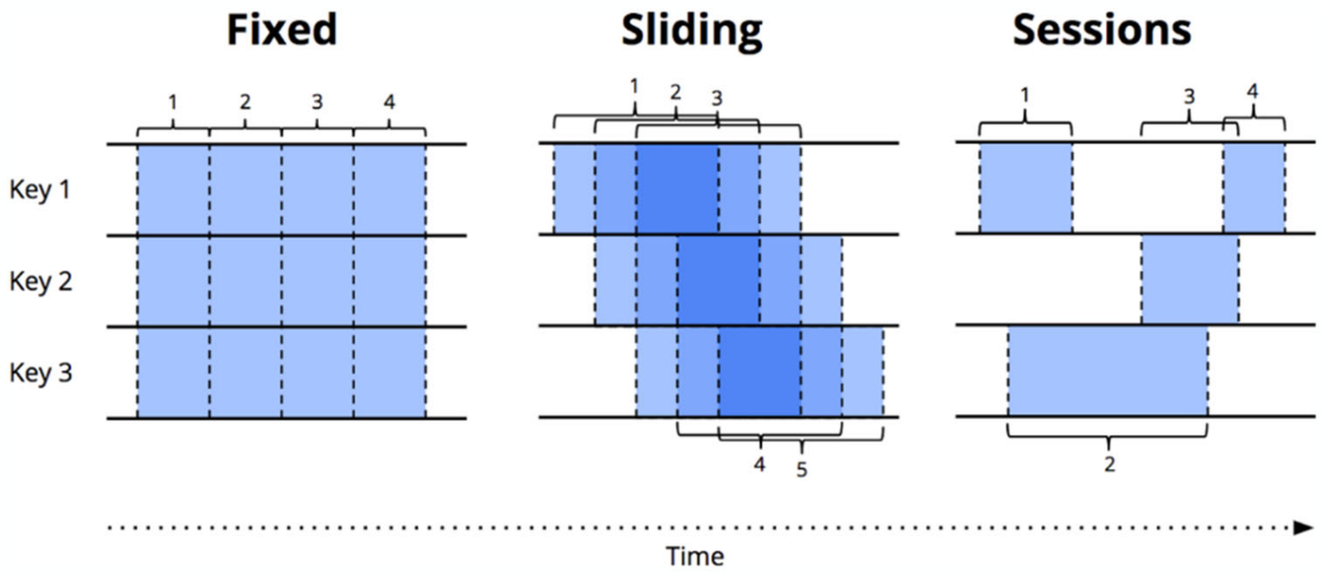


```
PCollection<KV<String, Integer>> input = IO.read( ... );
PCollection<KV<String, Integer>> output = input
  .apply(Sum.integersPerKey());
```

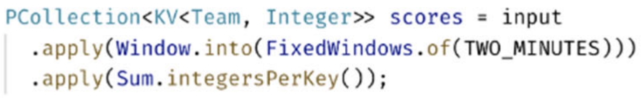


Where in Event Time

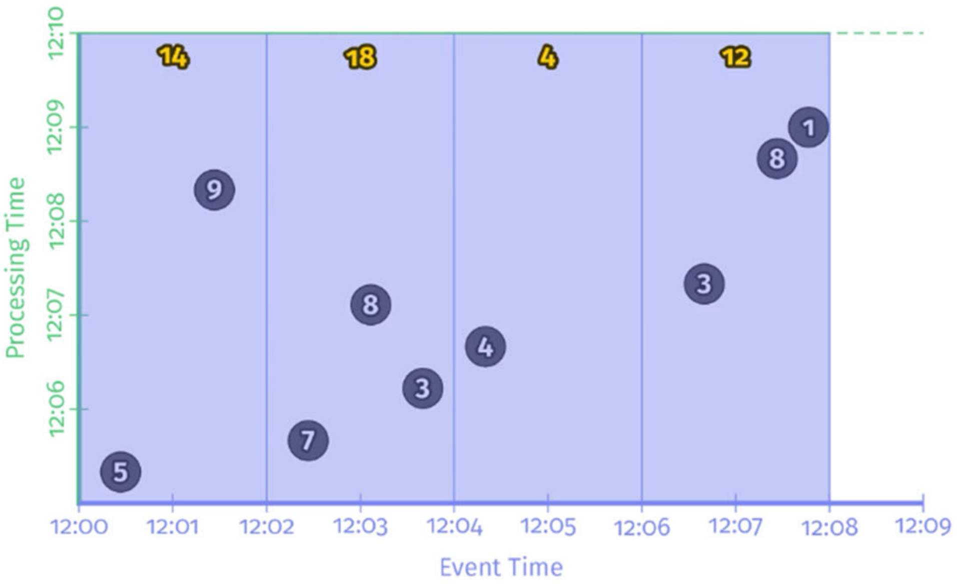
Windowing



Where in Event Time



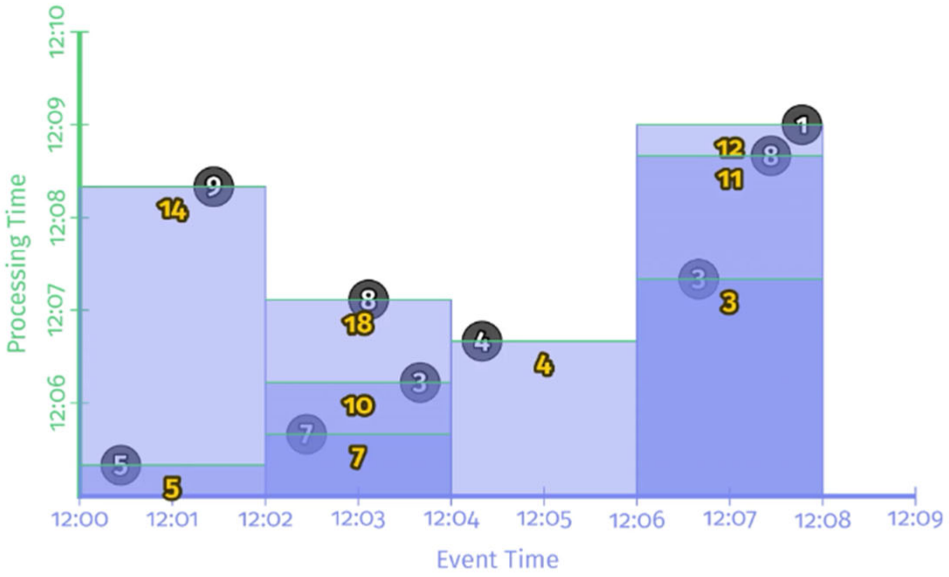
```
PCollection<KV<Team, Integer>> scores = input
  .apply(Window.into(FixedWindows.of(TWO_MINUTES)))
  .apply(Sum.integersPerKey());
```



When in Processing Time

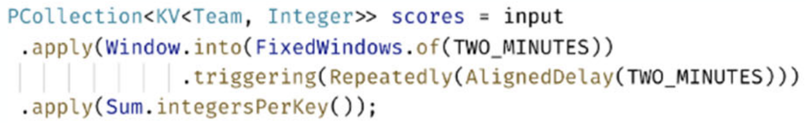
Repeated Update Triggers

```
PCollection<KV<Team, Integer>> scores = input
 .apply(Window.into(FixedWindows.of(TWO_MINUTES))
               .triggering(Repeatedly(AfterCount(1))));
 .apply(Sum.integersPerKey());
```

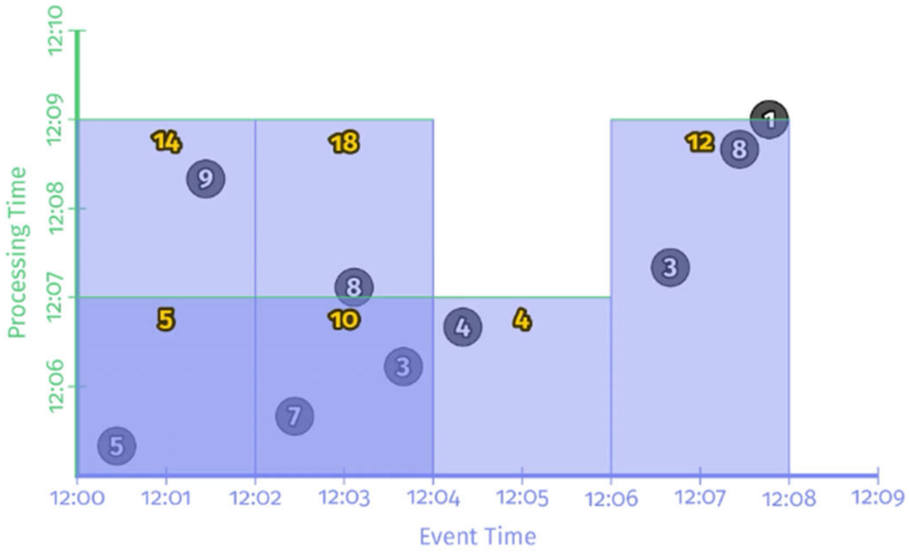


When in Processing Time

Repeated Update Triggers

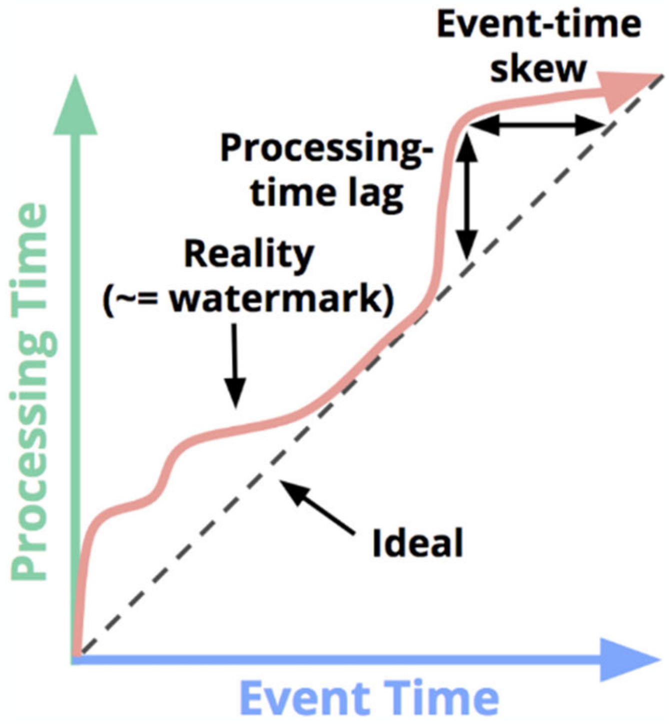


```
PCollection<KV<Team, Integer>> scores = input
 .apply(Window.into(FixedWindows.of(TWO_MINUTES))
               .triggering(Repeatedly(AlignedDelay(TWO_MINUTES)))
 .apply(Sum.integersPerKey());
```



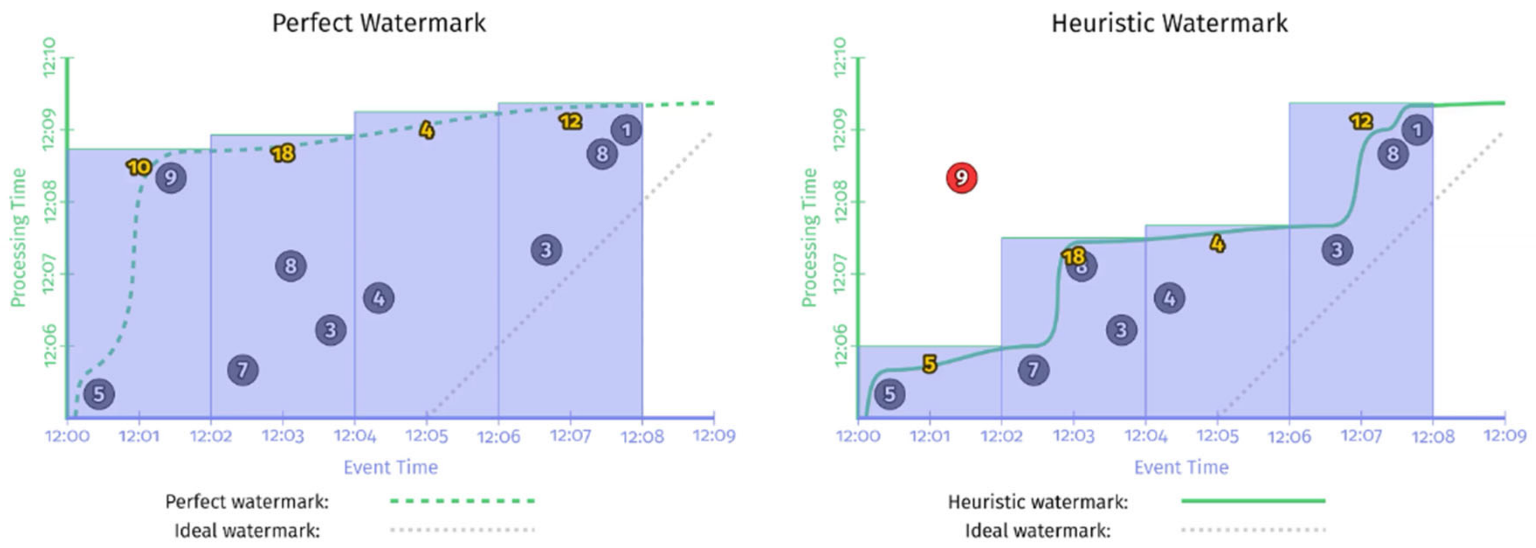
When in Processing Time

Watermarks — Completeness Triggers



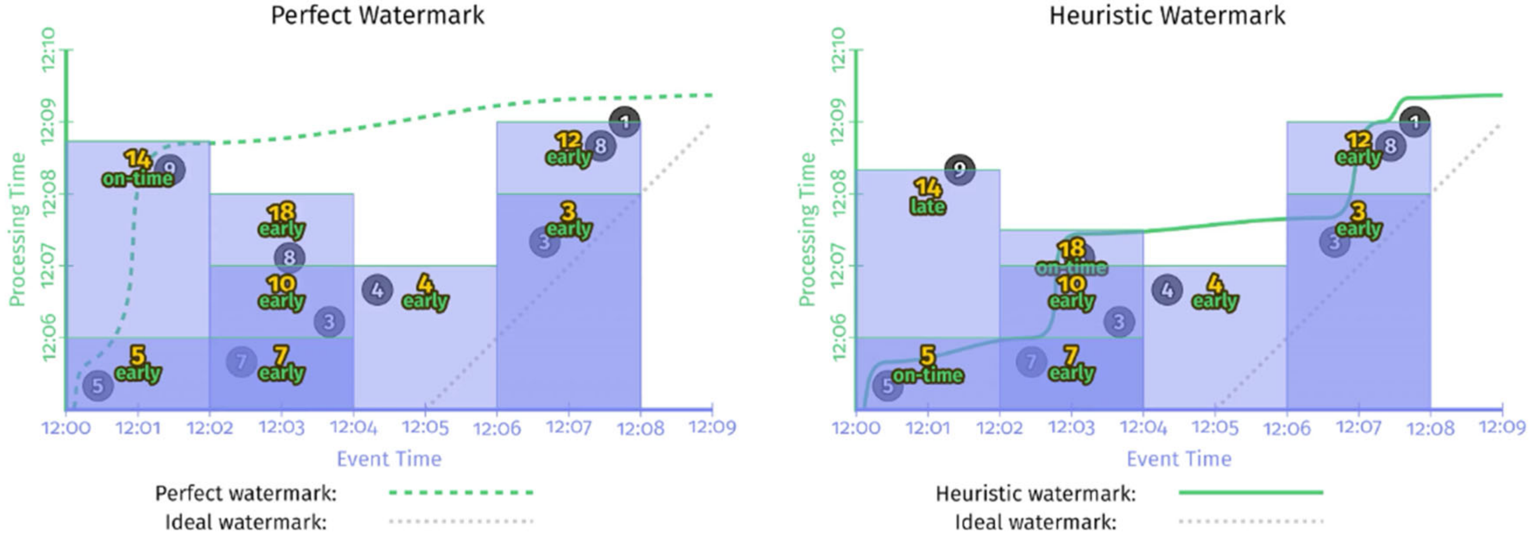
When in Processing Time

Watermarks — Completeness Triggers



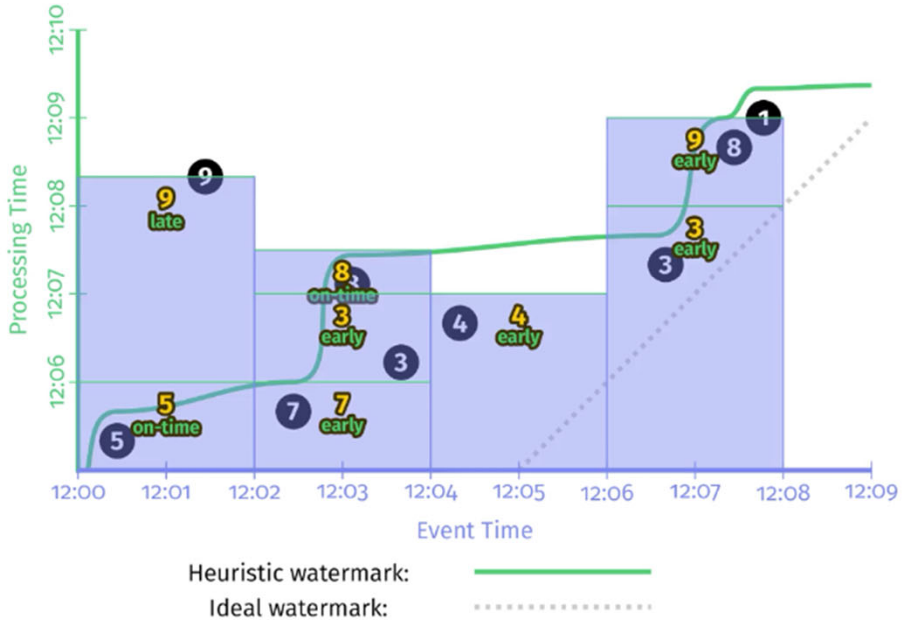
When in Processing Time

Early/On-time/Late Trigger



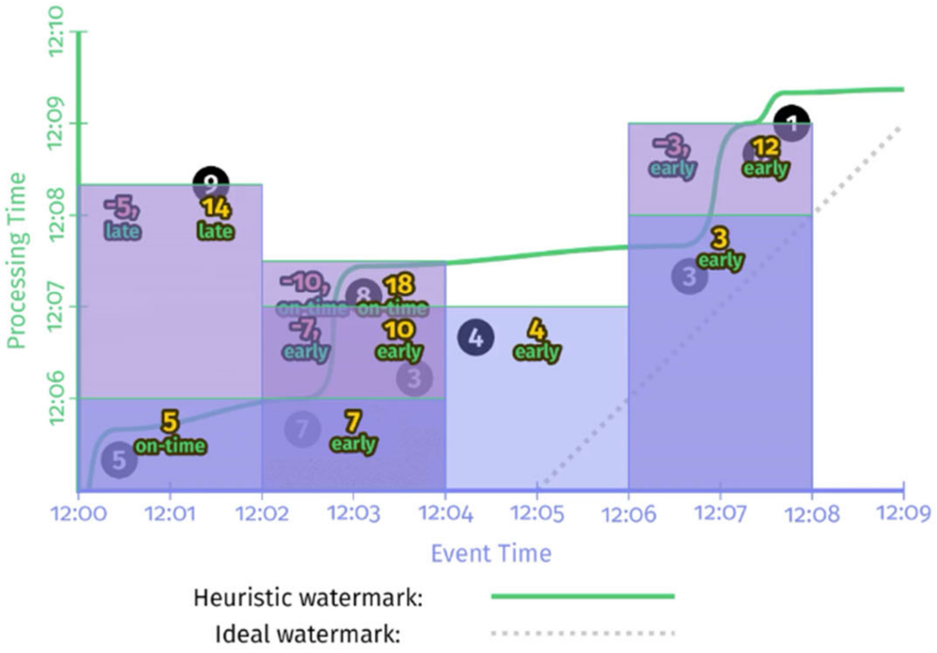
How do refinements relate

Discarding

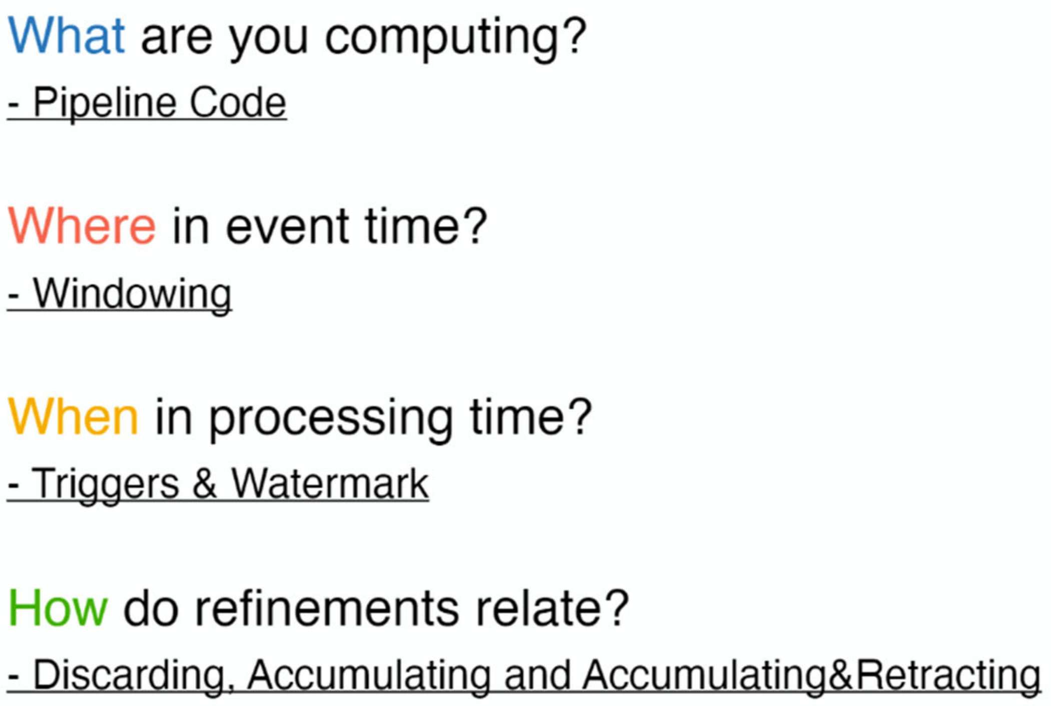


How do refinements relate

Accumulating & Retracting



The Dataflow Model Summary



What are you computing?

- Pipeline Code

Where in event time?

- Windowing

When in processing time?

- Triggers & Watermark

How do refinements relate?

- Discarding, Accumulating and Accumulating&Retracting

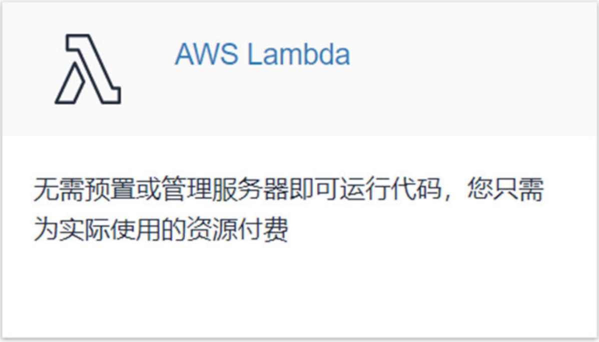
Serveless - FaaS

docker

kubernetes

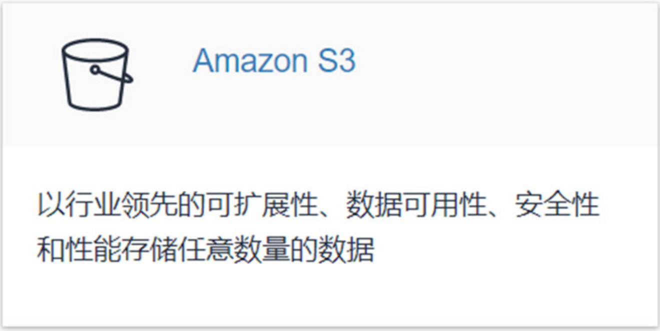
....

aws



AWS Lambda

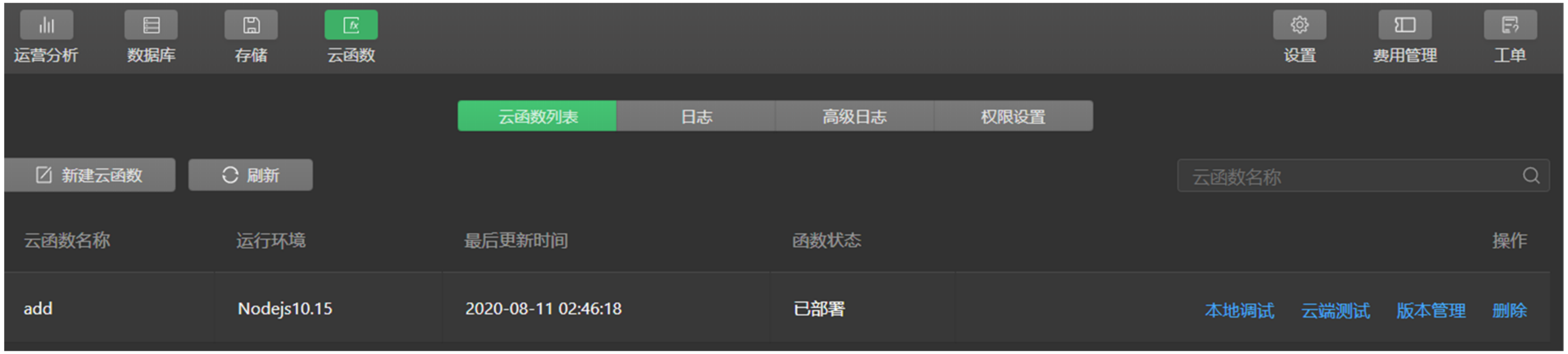
无需预置或管理服务器即可运行代码，您只需为实际使用的资源付费



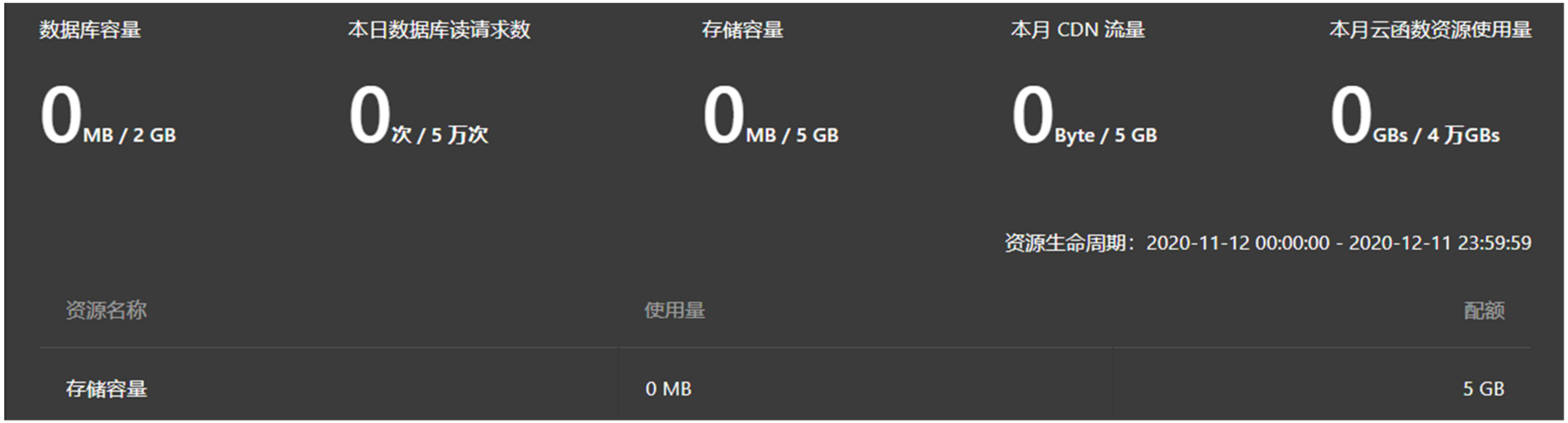
Amazon S3

以行业领先的可扩展性、数据可用性、安全性和性能存储任意数量的数据

Exp. Wechat Mini Program Serveless



云函数名称	运行环境	最后更新时间	函数状态	操作
add	Nodejs10.15	2020-08-11 02:46:18	已部署	本地调试　云端测试　版本管理　删除

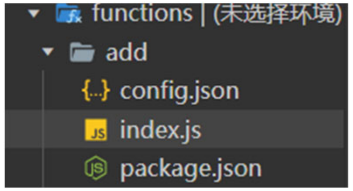


数据库容量	本日数据库读请求数	存储容量	本月 CDN 流量	本月云函数资源使用量
0 MB / 2 GB	0 次 / 5 万次	0 MB / 5 GB	0 Byte / 5 GB	0 GBs / 4 万GBs

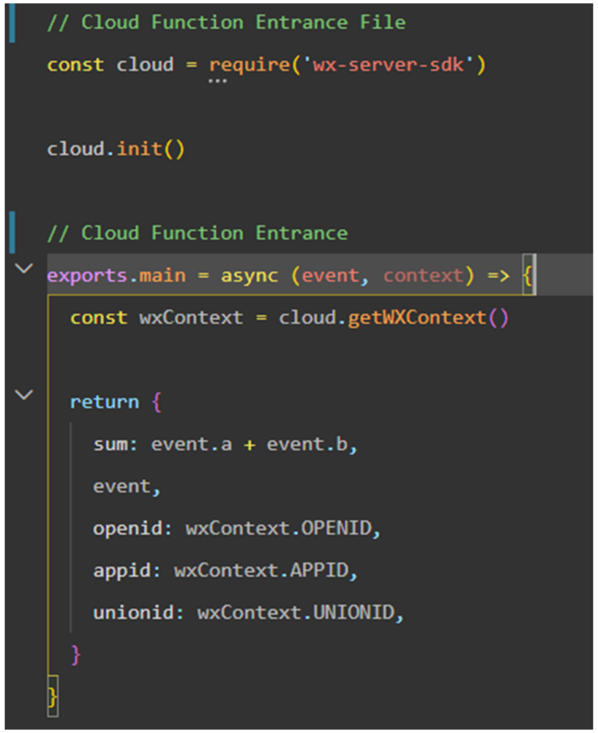
资源生命周期：2020-11-12 00:00:00 - 2020-12-11 23:59:59

资源名称	使用量	配额
存储容量	0 MB	5 GB

Exp. Serveless Function: add



```
▾ functions | (未选择环境)
  ▾ add
      config.json
      index.js
      package.json
```



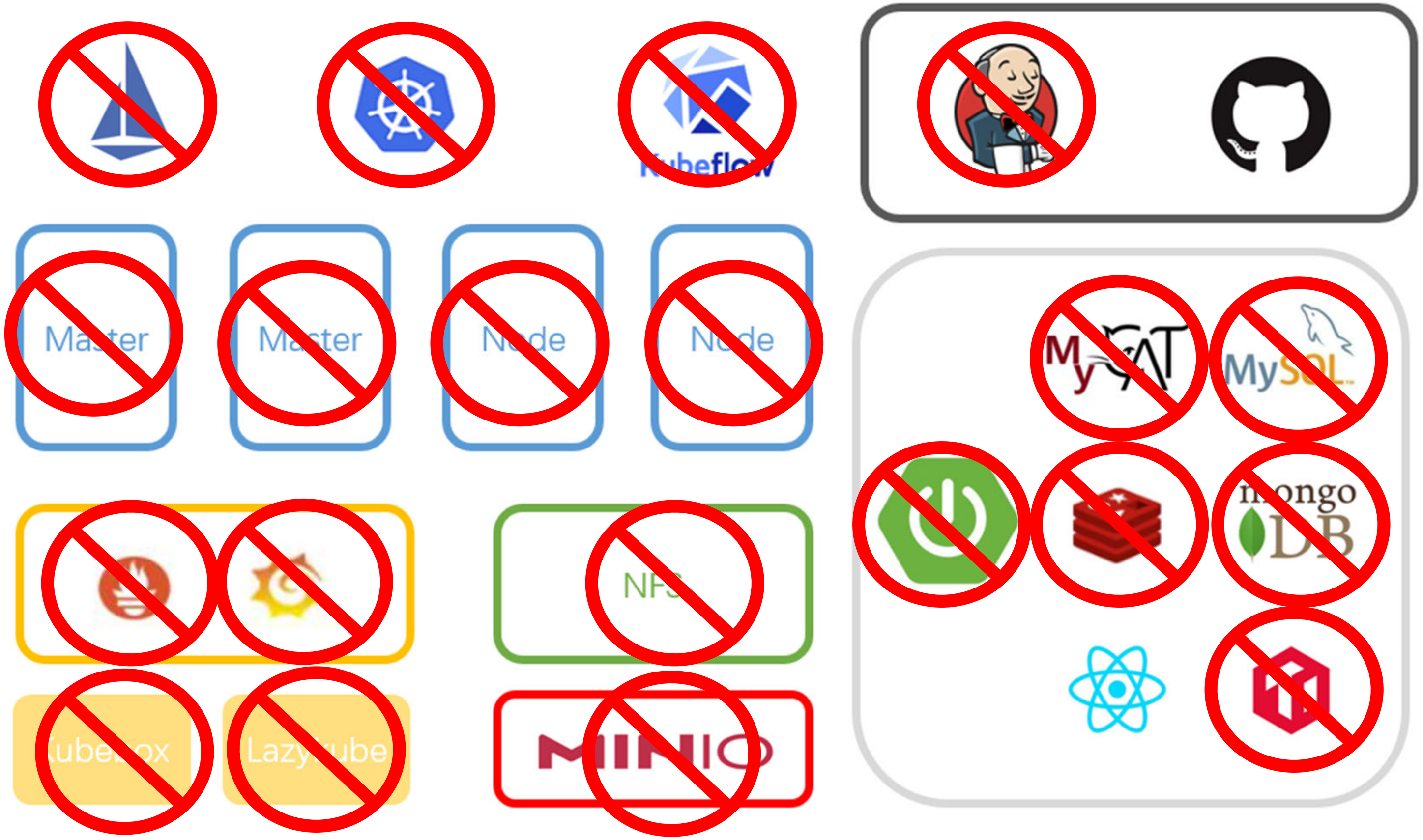
```
// Cloud Function Entrance File
const cloud = require('wx-server-sdk')

cloud.init()

// Cloud Function Entrance
exports.main = async (event, context) => {
  const wxContext = cloud.getWXContext()

  return {
    sum: event.a + event.b,
    event,
    openid: wxContext.OPENID,
    appid: wxContext.APPID,
    unionid: wxContext.UNIONID,
  }
}
```

The Pro of Serveless From A Project Exp.



The Challenge of Serveless