

分布式的那些小事

个人分享 by @Lhfcws_宸风

この番組はご覧のスポンサーの提供でお送りします

**MODERN INTERNET
APPLICATION CLUB**

Modern Internet Application club in Sun Yat-sen University.

What I gonna tell U

- 关于分布式（mainly 分布式计算）的初步认识
- 几个经典的基础分布式场景
- 离线计算&流式计算

What I gonna skip

- Strict Definition
- 比较深入全面的Overview

Story ~

- Problem: Scanning pages of the book(s).
- Subject: Lhfcws
- Resources: Xuejia, Jiahua

Single Mode

Scan by Lhfcws himself. (T.T)

```
1  for (int i = 0; i < N_BOOKS; i++) {  
2      for (int j = 0; j < bookPages[i]; j++) {  
3          scan();  
4      }  
5  }
```


Distribute Mode

Scan all together. (^.^)

```
1 void scanJob(jobList) {
2     for (int i in jobList) {
3         for (int j = 0; j < bookPages[i]; j++) {
4             scan();
5         }
6     }
7 }
8
9 void dispatch() {
10     // Assign by hash(x) = x % 3, push-mode
11     Lhfcws.assign(scanJob, [0, 3, 6, 9]);
12     Xuejia.assign(scanJob, [1, 4, 7]);
13     Jiahua.assign(scanJob, [2, 5, 8]);
14 }
```


Master Election

- Some of the members may slack off! Who is on charge ?!
- We need a master!

What may master do ?

- dispatch jobs (load balance)
- monitor workers' status
- watch the whole system

《《 请求将至，我从今开始守望，至死方休。我将不断网，不断电，不宕机。我将不插鼠标，不插键盘。我将尽忠职守，生死于斯。我是黑暗中的主机，系统中的守卫，抵御万恶的攻击，破晓时分的光线，唤醒眠者的号角，守护集群的坚盾。我将生命与荣耀献给整个集群，今夜如此，夜夜皆然。 《》

Master elected at first

- Assigned by human.
- If master dies, the whole system dies!

Master Auto Election

- Numbers of members will elect a master by themselves
- If the master dies, all the members will elect another master, so the system can continue running normally.
- Some possible election rules: max_pid, max_ID, max_machineID, highest_priority, etc.

Communication

- Of course the machines communicated with each other through Network (usually LAN).
- TCP/UDP , HTTP GET, RPC, RMI, etc. But if we use some 3rd framework, we need not worry about the protocols.
- Basically 2 models : Shared Memory, Message Passing.

Message Middleware

- Usually we call it Message Queue (MQ)
- Broadcast / Dispatch message.
- As a message buffer between sender & receiver.

Message Synchronization

- Distributed Lock / Counter
- Implements: Zookeeper, Redis, etc.

HeartBeats

- worker定时向Master回传状态消息来报告其状态（负载、CPU、内存、IO数、任务数、存活时间甚至拿job请求）以做到尽可能地实时监控

Other Scenarios

- Distributed Cache
- Distributed Queue
- Distributed Storage
- Distributed FileSystem
- Cluster
- ...

离线计算 & 流式计算

- 离线计算：非实时计算，通俗地讲，系统自己读一批数据然后就按照既定逻辑运行，不会再接受外部更新。（有点类似批处理）
- 流式计算：实时计算，对庞大连续的数据流不间断地做处理并实时生成结果。
- 两者实际有效面向的计算业务类型不一致，计算方式也不一样。

ありがとうございます！

以上Lhfcws所说的话不代表Lhfcws及其本人的意见。