**文档密级:**公司内部A

**Polaris 2.0性能测试报告**

**测试环境:**

Cpu: 2\*12\*2.4GHz

Mem: 800GB

Zk:5 companion:5 cynosure:5

**测试场景:**

订阅配置，订阅服务，注册服务

**一 订阅配置：**

5000，8000，10000个客户端同时订阅一个配置文件

把zk的数量减少为3个，结果如下图所示：

总结：

订阅配置文件回调时花费时间较多，经分析,是配置文件网络传输耗费了时间。对于多个配置文件的订阅和回调和单个配置文件的差距并不明显。

当zk节点减少为3个时10000个订阅者订阅时间增加到1648ms,回调增加到2303ms。Zk的减少对配置订阅时间存在一定的影响。

**二 服务发现订阅服务**

* **10000个订阅者订阅一个服务，该服务下存在1，10，100个实例**

过程中花费的平均时间如下：

将zk数量减少为3个时数据如下：

* **100个服务，每个服务100个服务提供者，100个服务订阅者**

SubscribeCost:订阅服务耗时

RegisterCost：注册服务耗时

Avg 平均时间

Max 最大时间

Top95 花费第95%多的时间的客户端

Top90 花费第90%多的时间的客户端

Min 最小时间

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Content  Index | subscribeCost | | | RegisterCost |
| init | cost | Callback | cost |
| avg | 750 | 6105 | 387 | 1474 |
| max | 2211 | 14145 | 512 | 2648 |
| Top95 | 1505 | 13662 | 444 | 2171 |
| Top90 | 1391 | 13495 | 401 | 2089 |
| min | 23 | 2535 | 0 | 174 |

总结：

随着1个服务下的服务提供者的增加，订阅服务所需要花费的时间呈线性增加。同时注册服务需要花费大量的时间，多服务和单一服务的订阅注册时间差距不明显。

zk减少为3个时注册所花费的时间为1623ms，订阅服务时间增加为10021ms增加较为明显，订阅服务主要压力在zk上。

**三 混合场景**

* **每个客户端同时订阅配置文件，订阅服务。测试1，10，100个服务提供者时订阅服务和订阅配置花费时间，10000个客户端。**

总结：

和单一场景下相比，混合场景下的各中操作所花费的平均时间基本上没有太多的变化。

**四 针对性测试**

针对比较费时的服务注册和，订阅服务，配置文件回调这三中操作，做了一下测试。

Companion ：5 zk ：5 cynosure 5

* 注册服务：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Providers | 4000 | 5000 | 6000 | 8000 | 10000 |
| Agv-Time | 790 | 938 | 1116 | 1434 | 1851 |
| Pvds/avg-time | 5.06 | 5.33 | 5.37 | 5.57 | 5.40 |

可以看到8000个注册者时性能达到最优，

* 订阅服务：

针对订阅服务，一个服务下存在10，30，50，100 个服务提供者时所花费时间。

总结：

1 配置文件改变时通知客户端，主要时间花费在反馈上，反馈改为异步后，时间显著减少。

2 一个服务存在大量服务提供者时订阅服务，时间主要花在zk io调用上。

3 服务配置信息改变通知提供者，这个和配置文件回调时场景一样。

4 注册服务时链路为sdk->companion->cynosure->companion->zk->sdk，链路较长，并且是同步的所以花费时间较长。

5 同时观察zk连接数，每台zk的负载大致一致，zk负载比较均衡。

**五 测试过程中zk，companion，cynosure 的cpu与内存占用**

Companion cpu 占用（%）

Companion 物理内存占用(kb)

Zk cpu 占用(%)

Zk 内存占用(kb)

Cynosure cpu 占用(%)

Cynosure 物理内存占用(kb)

**六 可优化的部分**

1配置文件的改变的feedback改为异步（已优化）。

2当一个服务存在大量的服务提供者时，订阅时间会显著增加。分析原因是因为，订阅服务时要读取该服务下每一个提供者的实例信息，造成zk的读取节点次数随着服务提供者的增加而增加。可以改为用一个节点存储该服务下所有提供者的实例信息，订阅服务时只读取一次节点就可以获取到所有提供者的信息。