

DSF 同步调用与 Flower DSF 异步调用对比测试报告

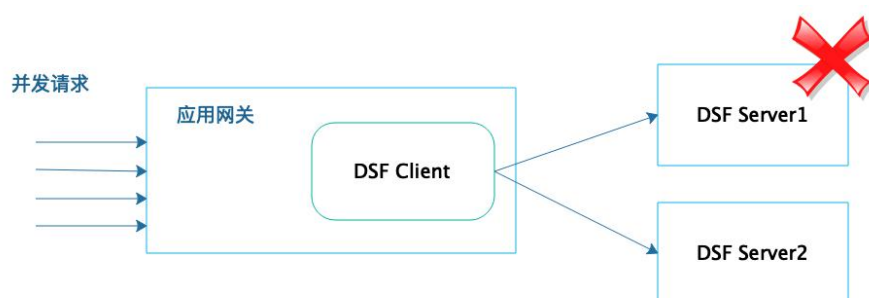
一、背景描述

因为 DSF 调用为同步阻塞式调用，当某一个 DSF Server 因为数据库异常或者其他原因而产生响应延迟的时候，会导致 DSF Client 调用者也被阻塞大量线程而运行缓慢，进而导致 DSF Client 的上级调用者也运行缓慢，也就是某一个异常 DSF Server 可能会拖累整个系统运行异常。

Flower 的异步 DSF 调用方式，可以隔离异常 DSF Server 产生的阻塞扩散，保证整个系统的正常运行。

二、测试用例

我们设计了如图所示的测试用例场景。



并发请求并发访问应用网关，应用网关通过 DSF Client 调用 DSF Server1 和 DSF Server2，DSF Server1 处于异常状态，响应时间很长。

应用网关有两种实现部署方式：

1. 传统的同步阻塞调用 DSF
2. Flower 异步非阻塞调用 DSF

我们的测试目标是证明，在 DSF Server1 响应延迟，造成阻塞的情况下，使用 1) 传统的同步阻塞调用 DSF 部署方案会导致应用网关级联崩溃，连正常运行的 DSF Server2 的请求也无法处理。而使用 2) Flower 异步非阻塞调用 DSF 部署方式的 Flower 应用网关则不受 DSF Server1 失效的影响，可以正常处理 DSF Server2 的请求。

三、环境配置

测试是在同一台机器，伪分布式部署 DSF Server1 和 DSF Server2，网关服务基于 **spring-boot 2** 进行开发。

1.

服务器配置

2.

CentOS Linux release 7.4.1708 (Core)
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 8
On-line CPU(s) list: 0-7
Thread(s) per core: 1
Core(s) per socket: 1
Socket(s): 8
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 42
Model name: Intel Xeon E312xx (Sandy Bridge)
Memory: 16G

3.

4. DSF server1 和 DSF server2 使用 DSF 默认配置信息，未做任何调整

5.

网关应用基于 **spring-boot**，对内置 **Tomcat** 参数进行调整。配置如下：

6.

```
server.tomcat.accept-count=100  
server.tomcat.max-connections=200  
server.tomcat.max-threads=100  
server.tomcat.min-spare-threads=10
```

7.

网关应用共提供 2 对接口：**a.** 一对是基于 DSF 同步阻塞调用 DSF 服务接口；**b.** 一对是基于 **Flower** 异步无阻塞调用 DSF 服务的接口。其中 **ping** 接口是正常可访问接口，**pingdead** 是慢响应(50s)接口。详情说明如下：

DSF 同步阻塞 DSF 接口:

- a. /sync/ping
- b. /sync/pingdead

Flower 异步无阻塞调用 DSF 接口:

- a. /async/ping
- b. /async/pingdead

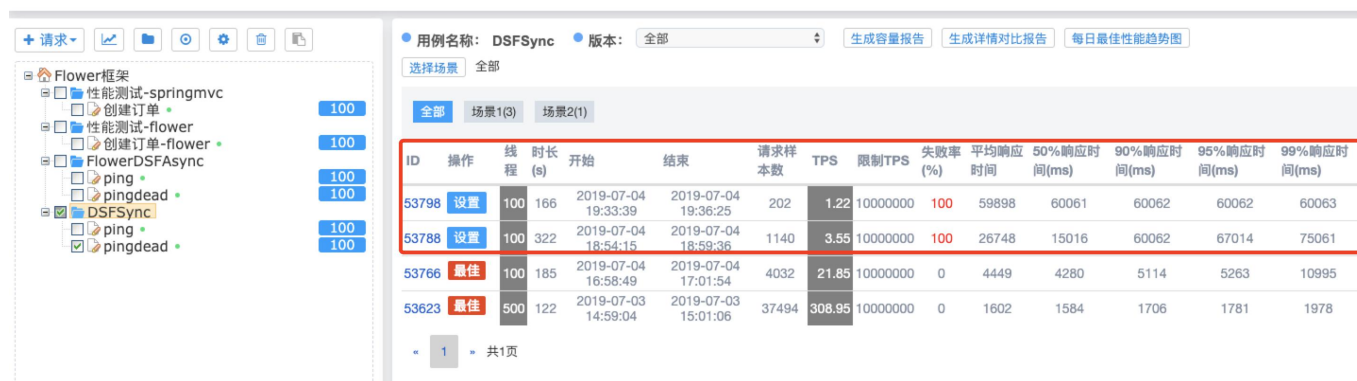
8.

9. 网关应用源代码从附件 [flower-dsf-async-showcase-master.zip](#) 下载。

四、测试结果

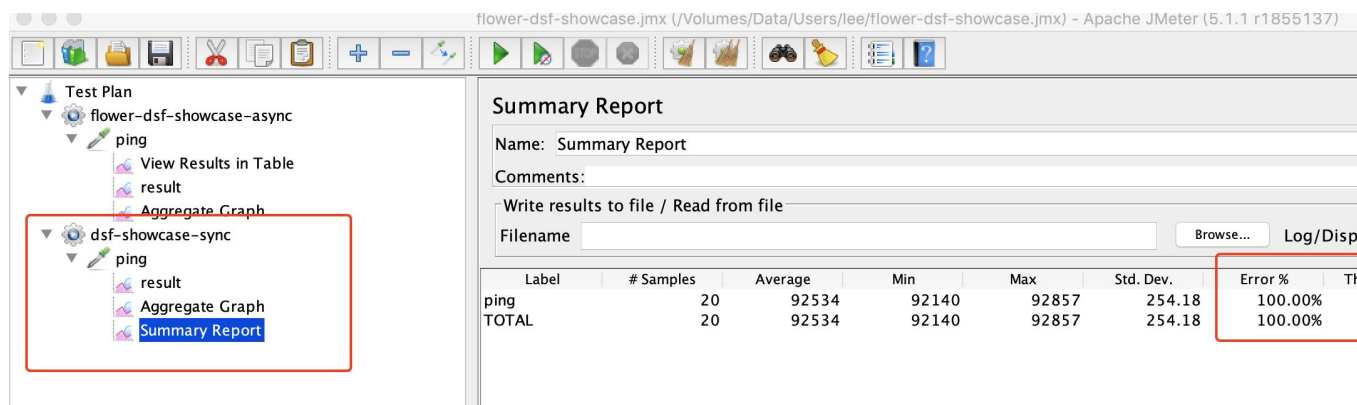
4.1 阻塞式调用 DSF

4.1.1 调用 DSF Server1, 模拟服务端无响应。客户端压测 (天锤) 结果如下图所示:



ID	操作	线程数	时长 (s)	开始	结束	请求样本数	TPS	限制TPS	失败率 (%)	平均响应时间	50%响应时间(ms)	90%响应时间(ms)	95%响应时间(ms)	99%响应时间(ms)
53798	设置	100	166	2019-07-04 19:33:39	2019-07-04 19:36:25	202	1.22	10000000	100	59898	60061	60062	60062	60063
53788	设置	100	322	2019-07-04 18:54:15	2019-07-04 18:59:36	1140	3.55	10000000	100	26748	15016	60062	67014	75061
53766	最佳	100	185	2019-07-04 16:58:49	2019-07-04 17:01:54	4032	21.85	10000000	0	4449	4280	5114	5263	10995
53623	最佳	500	122	2019-07-03 14:59:04	2019-07-03 15:01:06	37494	308.95	10000000	0	1602	1584	1706	1781	1978

4.1.2 调用 DSF Server2, 模拟请求另一个正常响应的 DSF 服务。由于调用 DSF Server1 时占用了服务容器的所有线程, 导致不能调用其他正常服务, 导致网关服务器假死。客户端压测 (Jmeter) 结果如下图所示:



Label	# Samples	Average	Min	Max	Std. Dev.	Error %
ping	20	92534	92140	92857	254.18	100.00%
TOTAL	20	92534	92140	92857	254.18	100.00%

4.2 Flower 异步调用 DSF

4.2.1 调用 DSF Server1, 模拟服务端无响应。客户端压测 (天锤) 结果如下图所示:

用例名称: FlowerDSFAsync 版本: 全部

生成容量报告 生成详情对比报告 每日最佳性能趋势图

选择场景 全部

全部 场景1(5) 场景2(1) 场景3(1) 场景4(1)

ID	操作	线程	时长 (s)	开始	结束	请求样本数	TPS	限制TPS	失败率(%)	平均响应时间	50%响应时间(ms)	90%响应时间(ms)	95%响应时间(ms)	99%响应时间(ms)
53797	设置	100	130	2019-07-04 19:29:40	2019-07-04 19:31:49	400	3.10	10000000	100	30868	30989	30998	30999	31015
53796	设置	100	129	2019-07-04 19:26:19	2019-07-04 19:28:28	400	3.11	10000000	100	30872	30990	30998	30999	31003
53795	设置	100	129	2019-07-04 19:23:01	2019-07-04 19:25:10	400	3.11	10000000	100	30869	30989	30997	30999	31001
53794	设置	100	130	2019-07-04 19:19:34	2019-07-04 19:21:43	400	3.10	10000000	100	30872	30990	30998	30999	31001
53784	设置	100	315	2019-07-04 18:46:23	2019-07-04 18:51:37	1000	3.18	10000000	100	30946	30993	30999	31000	31002
53665	最佳	100	101	2019-07-03 18:30:08	2019-07-03 18:31:48	286433	2,868.60	10000000	0	32	27	53	67	120
53618	最佳	100	61	2019-07-03 14:45:25	2019-07-03 14:46:26	13045	215.16	10000000	0	458	437	508	569	900
53608	最佳	500	206	2019-07-03 13:35:29	2019-07-03 13:38:55	1768254	8,595.52	10000000	0.0007	55	36	119	164	272

1 共1页

4.2.2 调用 DSF Server2，模拟请求另一个正常响应的 DSF 服务。结果显示是可以正常响应用户请求的。客户端压测（Jmeter）结果如下图所示：

Test Plan

- flower-dsf-showcase-async
 - ping
 - result
 - Aggregate Graph
 - Summary Report
- flower-dsf-showcase-sync
 - ping
 - result
 - Aggregate Graph
 - Summary Report

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename: Browse... Log/Disp...

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput
ping	16792	70	58	1272	33.40	0.00%	27.00
TOTAL	16792	70	58	1272	33.40	0.00%	27.00

4.3 测试结论

1. 同步阻塞调用 DSF 会造成雪崩效应，在 DSF 服务响应时间比较长的情况下，应用网关吞吐量降低，严重情况下会导致服务网关不响应，出现服务网关假死现象。
2. Flower 异步调用可以更优雅处理单个 DSF 服务响应慢/无响应的情况，不会阻塞其他服务接口对外提供服务，充分利用服务网关性能，提升系统处理能力、系统吞吐量。

五、为什么 Flower DSF 异步调用可以隔离异常、不会阻塞崩溃整个系统？

为什么会有这样一个结果，首先我们基于 Flower 进行开发，Flower 容器和 Web 容器通过异步方式调用，不会阻塞 Web 容器的线程；其次，Flower 内部也通过异步消息驱动的方式进行服务调用，不会阻塞线程；再次，通过重构 flower-dsf-client-async，使 DSF 与 Flower 无缝整合，DSF Client 可以异步调用 DSF 服务，不会阻塞线程；总之，利用 Flower 的功能特性，实现了一个无阻塞、即时响应的系统，单个 DSF 服务的异常不会级联扩散到应用系统的其他层面。

支持 DFS 异步调用的 Flower DSF Client 扩展包：

```
<dependency>
  <groupId>com.ly.train</groupId>
    <artifactId>flower-dsf-client-async</artifactId>
    <version>0.0.1-SNAPSHOT</version>
</dependency>
```

六、总结

Flower 是一个反应式微服务框架，使用 Flower 框架开发的应用系统可以得到更好的响应特性，使用 Flower 框架开发的应用系统具有更好的性能，更高的可用性，更低的服务器成本。

关于更多 Flower 功能特性，欢迎体验 [Flower 框架](https://github.com/zhihuili/flower.git)——
<https://github.com/zhihuili/flower.git>。