

压测报告

一、软件组成

1. Gatling——负责发送 http 请求，输出服务器的响应情况。

GatlingTestSimulation.scala:

同时发送 400 个 <http://localhost:8083/infolist> 请求

```
31 class GatlingTestSimulation extends Simulation {  
32  
33     val httpProtocol = http  
34         .baseUrl("http://localhost:8083")  
35  
36     val infoes = exec(http("informations").get("/infolist"))  
37     // val files = exec(http("files").get("/files/360p_0 Grace_Trim.mp4"))  
38  
39     val scn = scenario("TEST WEB-SERVER").exec(infoes)  
40     // val scn = scenario("TEST WEB-SERVER").exec(files)  
41  
42     setUp(scn.inject(atOnceUsers(400)).protocols(httpProtocol))  
43     // setUp(scn.inject(rampUsers(600) over (3 seconds)).protocols(httpProtocol))  
44  
45 }
```

2. prometheus——负责性能监控，并根据一定的规则发出警报。

prometheus.yml:

alerting 的配置表示 prometheus 与 alertmanager 的通信地址为 localhost:9093, rule_files 的配置表示 prometheus 报警规则在 prometheus.rules.yml 中说明, scrape_configs 的配置表示 prometheus 的监控界面所处地址为 localhost:9090, 它监控的目标是 web-server, 该目标的性能数据所处的地址是 localhost:8083/actuator/prometheus

```
7 # Alertmanager configuration  
8 alerting:  
9   alertmanagers:  
10     - static_configs:  
11       - targets:  
12         - localhost:9093  
13  
14 # Load rules once and periodically evaluate them according to the global  
15 rule_files:  
16   - "prometheus.rules.yml"  
17   # - "first_rules.yml"  
18   # - "second_rules.yml"  
19  
20 # A scrape configuration containing exactly one endpoint to scrape:  
21 # Here it's Prometheus itself.  
22 scrape_configs:  
23   # The job name is added as a label `job=<job_name>` to any timeseries  
24   - job_name: 'prometheus'
```

```

20 # A scrape configuration containing exactly one endpoint to scrape:
21 # Here it's Prometheus itself.
22 scrape_configs:
23   # The job name is added as a label `job=<job_name>` to any timeseries scraped
24   - job_name: 'prometheus'
25
26     # metrics_path defaults to '/metrics'
27     # scheme defaults to 'http'.
28
29     static_configs:
30     - targets: ['localhost:9090']
31
32   - job_name: web-server
33     metrics_path: /actuator/prometheus
34     static_configs:
35     - targets: ['localhost:8083']

```

Prometheus Alerts Graph Status ▾ Help

Enable query history

http_server_requests_seconds_max

Execute - insert metric at cursor - ▾

Graph Console

◀ Moment ▶

Element

http_server_requests_seconds_max{exception="None",instance="localhost:8083",job="web-server",method="GET",outcome="SUCCESS",status="200",uri="/files/{name}"}
http_server_requests_seconds_max{exception="None",instance="localhost:8083",job="web-server",method="GET",outcome="SUCCESS",status="200",uri="/infolist"}
http_server_requests_seconds_max{exception="None",instance="localhost:8083",job="web-server",method="OPTIONS",outcome="SUCCESS",status="200",uri="/upload"}
http_server_requests_seconds_max{exception="None",instance="localhost:8083",job="web-server",method="POST",outcome="SUCCESS",status="200",uri="/upload"}
http_server_requests_seconds_max{exception="ClientAbortException",instance="localhost:8083",job="web-server",method="GET",outcome="SUCCESS",status="200",uri="/files/{name}"}
http_server_requests_seconds_max{exception="None",instance="localhost:8083",job="web-server",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus"}

Add Graph

prometheus.rules.yml:

如果 http 请求的最大响应时间大于 0.1 秒，就发送一个警报给 alertmanager。

```

1 groups:
2   - name: example
3     rules:
4       - alert: LongLatencyTime
5         expr: http_server_requests_seconds_max{exception="None",method="GET",status="200"} > 0.1
6         for: 5s
7         labels:
8           severity: page
9         annotations:
10          summary: long latency time, the system need to be extended

```

3. alertmanager——负责获取 prometheus 发出的警报，然后将警报发到个人邮箱。

```

21 global:
22     #超时时间
23     resolve_timeout: 5m
24     #smtp地址需要加端口
25     smtp_smarthost: 'smtp.qq.com:465'
26     smtp_from: '443160587@qq.com'
27     #发件人邮箱账号
28     smtp_auth_username: '443160587@qq.com'
29     #账号对应的授权码（不是密码）
30     smtp_auth_password: 'fthvkejwmgubieh'
31     smtp_require_tls: false
32
33 route:
34     group_by: ['alertname']
35     group_wait: 10s
36     group_interval: 1m
37     repeat_interval: 4h
38     receiver: 'mail'
39 receivers:
40 - name: 'mail'
41     email_configs:
42 - to: '443160587@qq.com'

```

1 alert for alertname=LongLatencyTime

[View In AlertManager](#)

[1] Firing

Labels

alertname = LongLatencyTime
 exception = None
 instance = localhost:8083
 job = web-server
 method = GET
 outcome = SUCCESS
 severity = page
 status = 200
 uri = /infolist

Annotations

summary = long latency time, the system need to be extended

[Source](#)

二、实验过程及结果

2.1 1 个服务器

```
Simulation gatlingtest.GatlingTestSimulation completed in 4 seconds
Parsing log file(s)...
Parsing log file(s) done
Generating reports...
```

```
=====
--- Global Information ---
> request count                400 (OK=400   KO=0   )
> min response time           1948 (OK=1948  KO=-   )
> max response time           4529 (OK=4529  KO=-   )
> mean response time          3373 (OK=3373  KO=-   )
> std deviation                909 (OK=909   KO=-   )
> response time 50th percentile 3347 (OK=3347  KO=-   )
> response time 75th percentile 4303 (OK=4303  KO=-   )
> response time 95th percentile 4475 (OK=4475  KO=-   )
> response time 99th percentile 4520 (OK=4520  KO=-   )
> mean requests/sec           80 (OK=80    KO=-   )
--- Response Time Distribution ---
> t < 800 ms                   0 ( 0%)
> 800 ms < t < 1200 ms        0 ( 0%)
> t > 1200 ms                  400 (100%)
> failed                       0 ( 0%)
=====
```

server0:

GET 192.168.99.100:18080/actuator/prometheus Send Save

Pretty Raw Preview Visualize Text

```
16 # TYPE process_start_time_seconds gauge
17 process_start_time_seconds 1.594474647449E9
18 # HELP http_server_requests_seconds
19 # TYPE http_server_requests_seconds summary
20 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 866.0
21 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 1879.868880494
22 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 1.0
23 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.24514087
24 # HELP http_server_requests_seconds_max
25 # TYPE http_server_requests_seconds_max gauge
26 http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 2.067787485
27 http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.0
```

2.2 3 个服务器

```
Simulation gatlingtest.GatlingTestSimulation completed in 2 seconds
Parsing log file(s)...
Parsing log file(s) done
Generating reports...
```

```
=====
--- Global Information ---
> request count                400 (OK=400   KO=0   )
> min response time           1214 (OK=1214  KO=-   )
> max response time           2534 (OK=2534  KO=-   )
> mean response time          1939 (OK=1939  KO=-   )
> std deviation                348 (OK=348   KO=-   )
> response time 50th percentile 1947 (OK=1947  KO=-   )
> response time 75th percentile 2263 (OK=2263  KO=-   )
> response time 95th percentile 2447 (OK=2447  KO=-   )
> response time 99th percentile 2479 (OK=2479  KO=-   )
> mean requests/sec          133.333 (OK=133.333 KO=-   )
--- Response Time Distribution ---
> t < 800 ms                   0 ( 0%)
> 800 ms < t < 1200 ms        0 ( 0%)
> t > 1200 ms                  400 (100%)
> failed                       0 ( 0%)
=====
```

server0:

GET

192.168.99.100:18080/actuator/prometheus

Send

Save

PrettyRawPreviewVisualizeText

13jvm_memory_used_bytes{area="heap",id="Survivor Space",} 1004440.0
14jvm_memory_used_bytes{area="nonheap",id="Compressed Class Space",} 5487072.0
15# HELP process_start_time_seconds Start time of the process since unix epoch.
16# TYPE process_start_time_seconds gauge
17process_start_time_seconds 1.594474647449E9
18# HELP http_server_requests_seconds
19# TYPE http_server_requests_seconds summary
20http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 1266.0
21http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 2449.81435761
22http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 4.0
23http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.394384653
24# HELP http_server_requests_seconds_max
25# TYPE http_server_requests_seconds_max gauge
26http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 1.932426302

server1:

GET

192.168.99.100:18081/actuator/prometheus

Send

Save

PrettyRawPreviewVisualizeText

143rabbitmq_unrouted_published_total{name="rabbit",} 0.0
144# HELP jvm_buffer_memory_used_bytes An estimate of the memory that the Java virtual machine has used for off-heap memory.
145# TYPE jvm_buffer_memory_used_bytes gauge
146jvm_buffer_memory_used_bytes{id="direct",} 2170880.0
147jvm_buffer_memory_used_bytes{id="mapped",} 0.0
148# HELP jvm_memory_used_bytes The amount of used memory
149# TYPE jvm_memory_used_bytes gauge
150jvm_memory_used_bytes{area="heap",id="Tenured Gen",} 3.1330512E7
151jvm_memory_used_bytes{area="heap",id="Eden Space",} 4237736.0
152jvm_memory_used_bytes{area="nonheap",id="Metaspace",} 4.4556416E7
153jvm_memory_used_bytes{area="nonheap",id="Code Cache",} 1.35504E7
154jvm_memory_used_bytes{area="heap",id="Survivor Space",} 1418768.0
155jvm_memory_used_bytes{area="nonheap",id="Compressed Class Space",} 5517384.0
156# HELP jvm_threads_states_threads The current number of threads having NEW state
157# TYPE jvm_threads_states_threads gauge
158jvm_threads_states_threads(state="runnable",} 7.0
159jvm_threads_states_threads(state="blocked",} 0.0
160jvm_threads_states_threads(state="waiting",} 3.0
161jvm_threads_states_threads(state="timed-waiting",} 61.0
162jvm_threads_states_threads(state="new",} 0.0
163jvm_threads_states_threads(state="terminated",} 0.0
164# HELP http_server_requests_seconds
165# TYPE http_server_requests_seconds summary
166http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 466.0
167http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 1321.419254884
168http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 2.0
169http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.214320069
170# HELP http_server_requests_seconds_max
171# TYPE http_server_requests_seconds_max gauge
172http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 2.150793756

server2:

GET

192.168.99.100:18082/actuator/prometheus

Send

Save

PrettyRawPreviewVisualizeText

140jvm_threads_states_threads(state="waiting",} 3.0
141jvm_threads_states_threads(state="timed-waiting",} 61.0
142jvm_threads_states_threads(state="new",} 0.0
143jvm_threads_states_threads(state="terminated",} 0.0
144# HELP jvm_threads_daemon_threads The current number of live daemon threads
145# TYPE jvm_threads_daemon_threads gauge
146jvm_threads_daemon_threads 67.0
147# HELP jvm_classes_unloaded_classes_total The total number of classes unloaded since the Java virtual machine has started execution
148# TYPE jvm_classes_unloaded_classes_total counter
149jvm_classes_unloaded_classes_total 0.0
150# HELP jvm_memory_committed_bytes The amount of memory in bytes that is committed for the Java virtual machine to use
151# TYPE jvm_memory_committed_bytes gauge
152jvm_memory_committed_bytes{area="heap",id="Tenured Gen",} 3.9407616E7
153jvm_memory_committed_bytes{area="heap",id="Eden Space",} 1.5794176E7
154jvm_memory_committed_bytes{area="nonheap",id="Metaspace",} 4.9152E7
155jvm_memory_committed_bytes{area="nonheap",id="Code Cache",} 1.376256E7
156jvm_memory_committed_bytes{area="heap",id="Survivor Space",} 1966080.0
157jvm_memory_committed_bytes{area="nonheap",id="Compressed Class Space",} 6422528.0
158# HELP http_server_requests_seconds
159# TYPE http_server_requests_seconds summary
160http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 467.0
161http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 1294.500719872
162http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 2.0
163http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.26910336
164# HELP http_server_requests_seconds_max
165# TYPE http_server_requests_seconds_max gauge
166http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 2.250831601

2.3 6 个服务器

```
Simulation gatlingtest.GatlingTestSimulation completed in 3 seconds
Parsing log file(s)...
Parsing log file(s) done
Generating reports...
```

```
=====
--- Global Information ---
> request count                400 (OK=400 KO=0 )
> min response time           1363 (OK=1363 KO=- )
> max response time           3132 (OK=3132 KO=- )
> mean response time          2364 (OK=2364 KO=- )
> std deviation                502 (OK=502 KO=- )
> response time 50th percentile 2437 (OK=2437 KO=- )
> response time 75th percentile 2811 (OK=2811 KO=- )
> response time 95th percentile 3054 (OK=3054 KO=- )
> response time 99th percentile 3097 (OK=3097 KO=- )
> mean requests/sec           100 (OK=100 KO=- )
--- Response Time Distribution ---
> t < 800 ms                   0 ( 0%)
> 800 ms < t < 1200 ms        0 ( 0%)
> t > 1200 ms                  400 (100%)
> failed                       0 ( 0%)
=====
```

server0: 端口号 18080

```
18 # HELP http_server_requests_seconds
19 # TYPE http_server_requests_seconds summary
20 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 1400.0
21 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 2670.6871294
22 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 6.0
23 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.438210861
24 # HELP http_server_requests_seconds_max
25 # TYPE http_server_requests_seconds_max gauge
26 http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 1.946132594
```

server1: 端口号 18081

```
164 # HELP http_server_requests_seconds
165 # TYPE http_server_requests_seconds summary
166 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 599.0
167 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 1506.876000444
168 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 4.0
169 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.355336322
170 # HELP http_server_requests_seconds_max
171 # TYPE http_server_requests_seconds_max gauge
172 http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 1.743644968
```

server2: 端口号 18082

```
158 # HELP http_server_requests_seconds
159 # TYPE http_server_requests_seconds summary
160 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 601.0
161 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 1482.059744733
162 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 4.0
163 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.370624947
164 # HELP http_server_requests_seconds_max
165 # TYPE http_server_requests_seconds_max gauge
166 http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 1.880436662
```

server3: 端口号 18083

```
19 # HELP http_server_requests_seconds
20 # TYPE http_server_requests_seconds summary
21 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 200.0
22 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 979.304978893
23 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 1.0
24 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.240837742
25 # HELP http_server_requests_seconds_max
26 # TYPE http_server_requests_seconds_max gauge
27 http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/infoList",} 2.822611694
```

server4: 端口号 18084

```
143 # HELP http_server_requests_seconds
144 # TYPE http_server_requests_seconds summary
145 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 200.0
146 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 952.59993955
147 # HELP http_server_requests_seconds_max
148 # TYPE http_server_requests_seconds_max gauge
149 http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 2.764066864
```

server5: 端口号 18085

```
11 # HELP http_server_requests_seconds
12 # TYPE http_server_requests_seconds summary
13 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 200.0
14 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 863.863954592
15 http_server_requests_seconds_count{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 1.0
16 http_server_requests_seconds_sum{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/actuator/prometheus",} 0.18140711
17 # HELP http_server_requests_seconds_max
18 # TYPE http_server_requests_seconds_max gauge
19 http_server_requests_seconds_max{exception="None",method="GET",outcome="SUCCESS",status="200",uri="/info",} 2.893009913
```

三、测试结论

同样是 400 个并发的 http 请求，当只有一个服务器时，最小响应时间是 1948ms，最大响应时间是 4529ms；当有三个服务器时，最小响应时间 1214ms，最大响应时间 2534ms，比只有一个服务器时提速了不少；当有六个服务器时，最小响应时间 1363ms，最大响应时间 3132ms，与有三个服务器时相比，性能变化不大，说明服务器较多时，haproxy 调度消耗的时间也变多了。