

The background features a dark navy blue field with several thin, teal-colored lines that curve upwards from the bottom, creating a sense of motion or data flow. There are also some semi-transparent teal squares overlaid on the background.

EMQ X

千万连接性能测试报告

2019/06

关于 EMQ.....	3
特别致谢.....	3
测试结果概述.....	4
测试场景.....	6
千万连接测试.....	6
Pub 消息测试.....	6
50 万主题使用测试.....	7
消息广播测试.....	8
测试环境.....	9
测试工具.....	11
测试结果.....	12
千万级连接测试.....	12
Pub 消息测试.....	15
50 万主题使用测试.....	19
消息广播测试.....	22
附录.....	26
系统优化.....	26
测试脚本.....	27



关于 EMQ

EMQ (<https://www.emqx.io>) 是家面向 5G 和物联网市场的消息与流处理开源软件公司。

2013 年，EMQ 作为一个开源项目在 Github 上发布，目前已成为全球市场广泛应用的开源 MQTT 消息服务器。2017 年，在客户需求快速增长的驱动下，我们创立公司并推出 EMQ X 企业版，以加快开源项目的产品化部署应用，并为大型企业客户提供专业技术支持与服务。

EMQ 创始团队主要来自华为、IBM、亚马逊和苹果，拥有多年的工作经验和对行业的深刻理解，并对开源软件产业的未来抱有坚定的信念。

特别致谢

本次测试得到了华为云和 XMeter 的大力帮助。华为云为本次测试提供了所有测试所需的机器和网络资源；XMeter 为本次测试提供了支撑千万 MQTT 连接所需的专业性能测试工具平台。

华为云 (<https://www.huaweicloud.com>)：华为云是华为公司倾力打造的云战略品牌，致力于为全球客户提供领先的公有云服务，包含弹性云服务器、云数据库、云安全等云计算服务，软件开发服务，面向企业的大数据和人工智能服务，以及场景化的解决方案。

XMeter (<https://www.xmeter.net>)：云端 SaaS 性能测试产品 XMeter，让性能测试变得简单易行。XMeter 致力于在软件性能测试领域为客户提供高效便捷的性能测试生产工具及技术领先的低成本性能测试解决方案。XMeter 在开源社



区提供的 MQTT JMeter 插件，在全球有众多用户。

测试结果概述

本次测试在华为云上进行，主要进行了千万级的 MQTT 连接测试，以及多种组合的消息吞吐量测试，相关的测试结果如下所示。

注：如果不做特别说明，所有的连接默认都设置了 300 秒的 Ping 消息包。

1000 万连接测试：该测试只测试 MQTT 的连接，除了固定时间的 Ping 包，每个客户端不会 pub/sub 任何 MQTT 消息：

- 90%平均响应时间< 2ms；最大响应时间为 2.3 秒
- 每个节点 CPU 平均使用率大约为 8%左右，最高不超过 20%
- 每个节点平均内存使用约 9GB

最大 响应时间	最小 响应时间	前 90% 平均 响应时间	EMQ X CPU 使用	EMQ X 内存使用
2.308s	0s	0.0015s	8%-20%	9GB

消息吞吐测试：背景连接 900 万，用于模拟具有连接和发送 Ping 包的客户端，而剩下的 100 万则模拟用于发送实际业务数据的客户端

- 消息体为 88 字节
- 90% 平均响应时间 <70ms
- CPU 使用随着消息吞吐量增加变大
- 每个节点平均内存使用在 9 - 10GB

消息吞吐测试类型 1：Pub 消息测试

该测试类型下除了 900 万的背景连接，剩下的 100 万业务连接只做 pub 消息的操作。测试结果表明响应时间都非常小，CPU 的使用率小于 25%，内存为 9GB。



Pub (k/s)	Qos	最大 响应时间	最小 响应时间	前 90% 平均 响应时间	EMQ X CPU 使用	EMQ X 内 存使用
200	0	6.173s	0s	0.0001s	15%-20%	9GB
200	1	7.484s	0s	0.0087s	15%-25%	9GB

消息吞吐测试类型 2：50 万主题使用

该测试类型下除了 900 万的背景连接，剩下的 100 万连接主要将大量的主题用于消息转发的测试。该测试结果表明在使用 50 万主题的情况下，QoS 在 0 的时候，Pub 和 Sub 的吞吐量分别都可以稳定达到 20 万/秒；而 QoS 在 1 的时候，Pub 和 Sub 的吞吐量分别都可以稳定达到 10 万/秒。在大量主题使用的场景下，EMQ X 表现良好。

Pub /Sub (k/s)	Qos	最大 响应时间	最小 响应时间	前 90% 平 均响应时间	EMQ X CPU 使用	EMQ X 内 存使用
200	0	12.014s	0s	0.0355s	25%-30%	9GB
100	1	5.477s	0s	0.0063s	20%-25%	9GB

消息吞吐测试类型 3：消息广播测试

该测试类型下除了 900 万的背景连接，剩下的 100 万连接主要做消息广播的测试。该测试场景下，主要是用少量的 Pub 端往主题上发送消息，大量的 Sub 端在订阅相关的主题，实现消息在大量 Sub 端的广播。测试结果显示，在 QoS 为 0 的情况下，广播可以达到 180 万/秒的消息吞吐量；而 QoS 为 1 的情况下，广播可以达到 90 万/秒的消息吞吐量。

Pub (k/s)	Sub (k/s)	Qos	最大 响应时间	最小 响应时 间	前 90% 平 均响应时间	EMQ X CPU 使用	EMQ X 内 存使用
4	1800	0	159.561s	0s	0.0654s	30%-45%	9.5G
2	900	1	140.271s	0s	0.0500s	35%-55%	9.8G



测试场景

千万连接测试

该测试场景只测试客户端到服务器端的 MQTT 协议连接，验证千万级并发连接下 EMQ X 的表现。关注的主要参数有连接响应时间、EMQ X 服务器的 CPU、内存和网络等使用情况。

消息吞吐类测试

Pub 消息测试

该测试类型下除了 900 万的背景连接，剩下的 100 万业务连接只做 pub 消息的操作。该测试主要关注的目标是在 Pub 大量消息的情况下，EMQ 系统的表现。关注的主要参数有：消息 Pub 响应时间、EMQ X 服务器的 CPU、内存和网络等使用情况。

QoS 0 * 20 万/秒

- 连接
 - 900 万背景连接
 - 100 万 Pub 端连接
- Pub: 20 万/秒，100 万 Pub 客户端，每客户端间隔 5 秒 Pub 一次消息
- QoS: 0
- 主题数: 1000
- 消息体: 88 字节



QoS 1 * 20 万/秒

- 连接
 - 900 万背景连接
 - 100 万 Pub 端连接
- Pub: 20 万/秒, 100 万 Pub 客户端, 每客户端间隔 5 秒 Pub 一次消息
- QoS: 1
- 主题数: 1000
- 消息体: 88 字节

50 万主题使用测试

该测试类型下除了 900 万的背景连接, 剩下的 100 万连接主要将大量的主题用于消息转发的测试。该测试主要关注的目标是在使用大量主题的情况下, EMQ 系统的表现。关注的主要参数有: 消息 Pub、Sub 响应时间、EMQ X 服务器的 CPU、内存和网络等使用情况。

Pub: QoS 0 * 20 万/秒 + Sub: QoS 0 * 20 万/秒

- 连接
 - 900 万背景连接
 - 50 万 Pub 端连接
 - 50 万 Sub 端连接
- Pub: 20 万/秒, 每客户端间隔 2.5 秒 Pub 一次消息
- Sub: 20 万/秒, 每客户端间隔 2.5 秒收到一次消息
- QoS: 0
- 主题数: 50 万
- 消息体: 88 字节



Pub: QoS 1 * 10 万/秒 + Sub: QoS 1 * 10 万/秒

- 连接
 - 900 万背景连接
 - 50 万 Pub 端连接
 - 50 万 Sub 端连接
- Pub: 10 万/秒，每客户端间隔 5 秒 Pub 一次消息
- Sub: 10 万/秒，每客户端间隔 5 秒收到一次消息
- QoS: 1
- 主题数: 50 万
- 消息体: 88 字节

消息广播测试

该测试类型下除了 900 万的背景连接，剩下的 100 万连接主要做消息广播的测试。该测试场景下，主要是用少量的 Pub 端往主题上发送消息，大量的 Sub 端在监听相关的主题，实现消息在大量 Sub 端的广播。该测试主要关注的目标是在使用消息广播场景下，EMQ 系统的表现。关注的主要参数有：消息 Pub、Sub 响应时间、EMQ X 服务器的 CPU、内存和网络等使用情况。

Pub: QoS 0 * 4 k/秒 + Sub: QoS 0 * 180 万/秒

- 连接
 - 900 万背景连接
 - 10 万 Pub 端连接
 - 90 万 Sub 端连接
- Pub: 4 千/秒，每客户端间隔 25 秒 Pub 一次消息
- Sub: 180 万/秒
- QoS: 0



- 主题数: 2k
- 消息体: 88 字节

Pub: QoS 1 * 2 k/秒 + Sub: QoS 1 * 90 万/秒

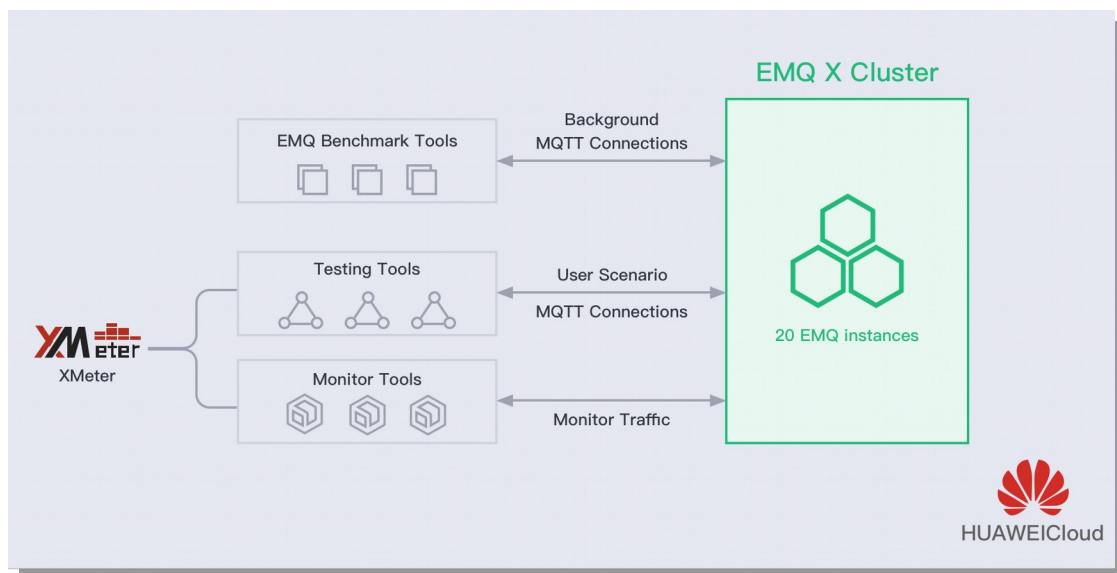
- 连接
 - 900 万背景连接
 - 10 万 Pub 端连接
 - 90 万 Sub 端连接
- Pub: 2 千/秒, 每客户端间隔 50 秒 Pub 一次消息
- Sub: 90 万/秒
- QoS: 1
- 主题数: 2k
- 消息体: 88 字节

测试环境

本次测试过程中由于资源协调的问题, 中间调整过一次, 因此有两套测试环境, 本章后面将列出测试环境与测试场景的对应表。

华为云中的测试部署图如下所示, 本次测试中使用了 EMQ Benchmark Tools 来模拟大量的 MQTT 连接; 而 XMeter 提供的基于 JMeter MQTT 插件的测试工具来模拟业务测试场景; XMeter 内置支持的监控工具用于监控 20 台被测 EMQ X 集群的资源使用情况。





服务器环境 1

- EMQ X 版本: 3.1.1
- 被测试机 (20 台)
 - 操作系统: CentOS Linux release 7.4.1708
 - CPU: 16 核
 - 内存: 32 G
 - 磁盘: 120GB
- 压力机
 - 20 台: 用于执行 Benchmark Tools 模拟背景连接, 每台主机 10 个网卡, 每个网卡模拟 5 万客户端连接 EMQ X
 - 操作系统: CentOS Linux release 7.4.1708
 - CPU: 16 核
 - 内存: 32 G
 - 磁盘: 120GB
 - 25 台用于 XMeter 连接 EMQ X, 模拟业务数据收发
 - 操作系统: CentOS Linux release 7.4.1708
 - CPU: 8 核
 - 内存: 16 G
 - 磁盘: 120GB
- 网络连接: 华为云内网连接



该测试环境执行了以下的测试场景：

- 千万连接测试
- Pub 消息测试
- 50 万主题测试使用测试

服务器环境 2

- EMQ X 版本: 3.1.1
- 被测试机: 20 台
 - 操作系统: CentOS Linux release 7.4.1708
 - CPU: 8 核
 - 内存: 16 G
 - 磁盘: 120GB
- 压力机: 70 台
 - 操作系统: CentOS Linux release 7.4.1708
 - CPU: 8 核
 - 内存: 16 G
 - 磁盘: 120GB
- 网络连接: 华为云内网连接

该测试环境执行了以下的测试场景：

- 消息广播测试

EMQ 集群安装

EMQ 安装成功后，使用命令行显示的结果，总共安装了 20 台 EMQ X 的一个集群。

```
Cluster    status:    [{running_nodes,    ['emqx@172.16.156.129',  
'emqx@172.16.155.227', 'emqx@172.16.154.60', 'emqx@172.16.153.8',  
'emqx@172.16.153.117', 'emqx@172.16.149.98', 'emqx@172.16.148.111',  
'emqx@172.16.143.85', 'emqx@172.16.141.174', 'emqx@172.16.137.124',  
'emqx@172.16.136.184', 'emqx@172.16.135.58', 'emqx@172.16.135.110',  
'emqx@172.16.134.203', 'emqx@172.16.132.229', 'emqx@172.16.131.175',  
'emqx@172.16.128.109', 'emqx@172.16.144.37', 'emqx@172.16.143.140',
```



```
'emqx@172.16.130.177']}]
```

测试工具

- XMeter 企业版 2.0.1: <https://www.xmeter.net>

XMeter 是一个性能测试管理平台，基于开源的 JMeter 性能测试工具。XMeter 可以支持大规模、高并发的性能测试，比如实现千万级别的 MQTT 并发连接测试。除了测试 MQTT 协议之外，还可以支持 HTTP/HTTPS 等主流的应用的测试。

- JMeter-MQTT 插件: mqtt-xmeter-1.13 –

<https://github.com/emqx/mqtt-jmeter>

由 XMeter 实现的开源 MQTT 性能测试插件，在众多的项目中得到了使用，目前是 JMeter 社区中流行度最高的 MQTT 插件。

- JMeter 版本: JMeter5.0 – <https://jmeter.apache.org>
- emqtt-benchmark-tools:

https://github.com/emqtt/emqtt_benchmark

由 EMQ 实现的用于模拟大量 MQTT 连接的测试工具。

测试结果

千万级连接测试

测试方法描述：使用了 emqtt-benchmark-tools 来进行连接测试。以下说明了该命令行工具的使用方式。

```
emqtt_bench_conn -h ${emqx_node_ip} -c 50000 -n 1 -i 20 --ifaddr ${ips}
```



-h \${emqx_node_ip} 为 emqx 节点 IP, -c 50000 最大连接用户数, -n 1 开始编号, -i 20 创建连接等待时间, --ifaddr \${ips} 指定使用 IP

下面列出了命令行所支持的参数列表及其对应含义。

```
mqtt server port number [default: 1883]
-c, --count      max count of clients [default: 200]
-n, --startnumber start number [default: 0]
-i, --interval   interval of connecting to the broker [default: 10]
-u, --username   username for connecting to server
-P, --password   password for connecting to server
-k, --keepalive  keep alive in seconds [default: 300]
-C, --clean      clean session [default: true]
-S, --ssl        ssl socoket for connecting to server [default: false]
--certfile       client certificate for authentication, if required by
                  server
--keyfile        client private key for authentication, if required by
                  server
--ifaddr         local ipaddress or interface address
```



Stats(20)			
Name	Connections/Count	Connections/Max	
emqx@172.16.152.2 35	500000	500000	
emqx@172.16.140.1 76	500000	500000	
emqx@172.16.129.1 82	500000	500000	
emqx@172.16.164.1 58	500000	500000	
emqx@172.16.162.1 46	500000	500000	
emqx@172.16.179.2 45	500000	500000	
emqx@172.16.165.1 73	500000	500000	
emqx@172.16.147.9 5	500000	500000	
emqx@172.16.173.1 99	500000	500000	

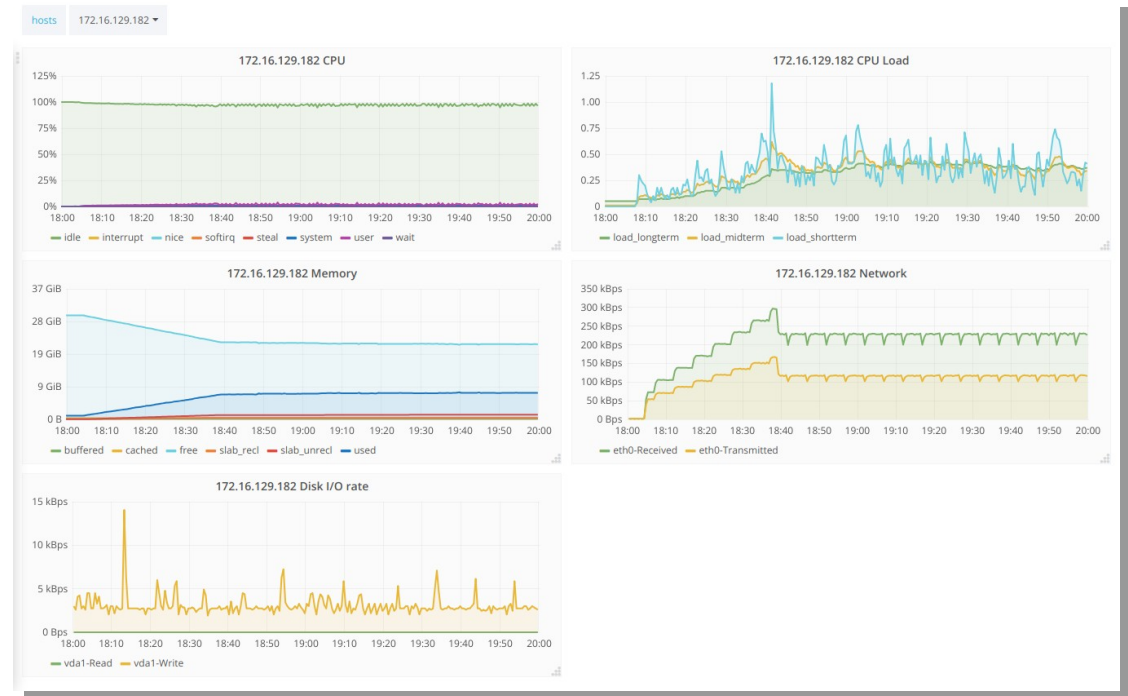
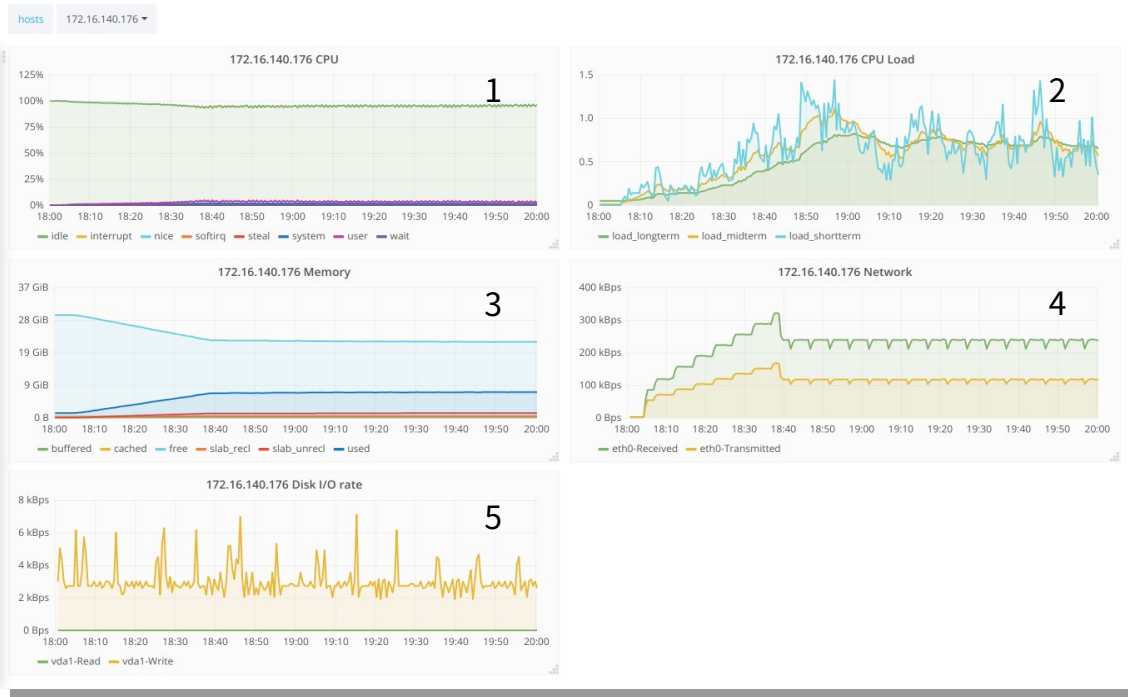
emqx@172.16.181.2 43	500000	500000	
emqx@172.16.154.1	500000	500000	
emqx@172.16.151.2 51	500000	500000	
emqx@172.16.161.1 81	500000	500000	
emqx@172.16.166.3 8	500000	500000	
emqx@172.16.129.1 30	500000	500000	
emqx@172.16.154.1 7	500000	500000	
emqx@172.16.180.1 76	500000	500000	
emqx@172.16.158.2 05	500000	500000	
emqx@172.16.185.2 29	500000	500000	
emqx@172.16.137.1 57	500000	500000	

如上截图为成功达到了 1000 万连接后，EMQ X Dashboard 的连接截图。每台机器连接 50 万*20 台 = 1000 万连接。

EMQ X 各个实例资源使用均衡，如下图所示是选择了其中两台机器的资源监控。CPU、内存和网络的使用情况都非常稳健。被测系统报告的解释（本文剩下的测试报告中对机器监控的图表都类似，如果没有特别之处，我们后面将不再赘述）：

1. CPU 使用：监控项为 idle, interrupt, nice, softirq, steal, system, user, wait;报告中数值均采用 user 项；
2. CPU 负载：load_longterm, load_midterm, load_shortterm 分别对应 CPU 负载 15 分钟，5 分钟，1 分钟；本报告中均采用 1 分钟值
3. 内存使用：监控项为 buffered, cached, free, slab_recl, slab_ubrecl, used; 本报告值均采用 used
4. 网络使用：网卡收到与发出流量
5. 磁盘：主要为操作系统磁盘操作及日志写入磁盘，本报告不关注此项





Pub 消息测试

该测试类型下除了 900 万的背景连接，剩下的 100 万业务连接只做 pub 消息的操作。

QoS 0 * 20 万/秒

报告解析

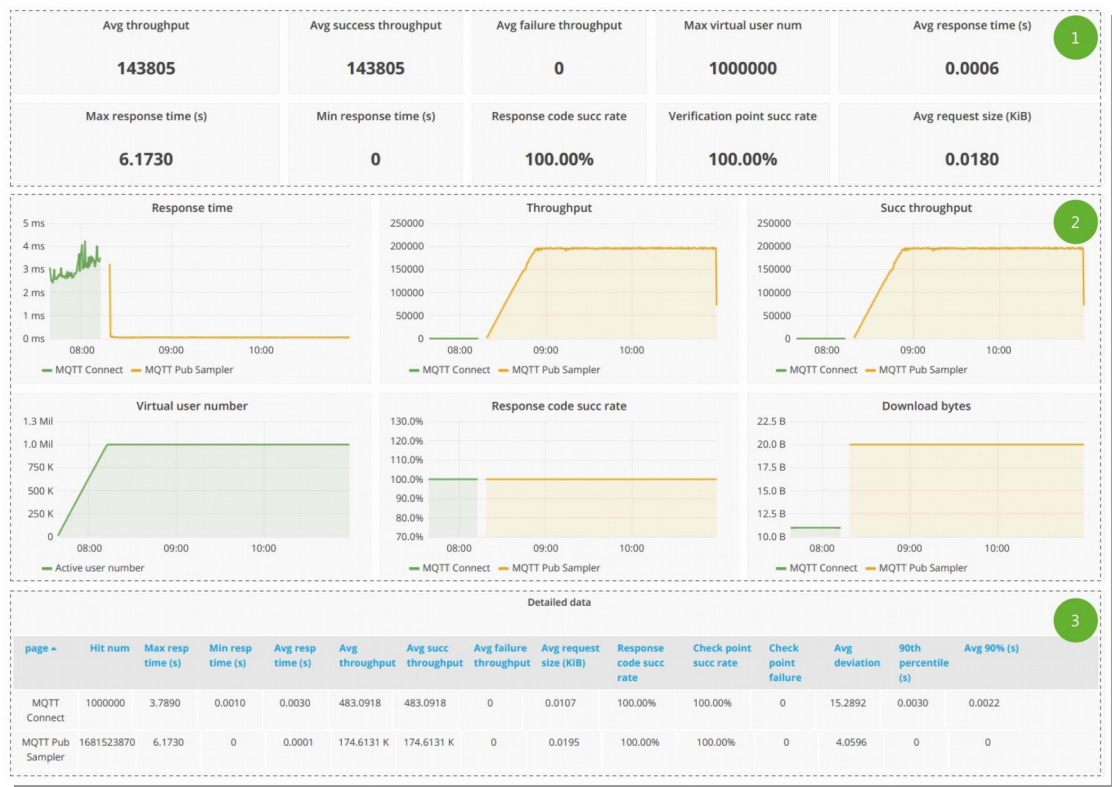
如下图所示，XMeter 的性能测试报告分成三大部分。后面报告的结果解读方式都类似，如果没有特别之处，我们不再赘述。

1. 概览：此区域的测试报告将测试步骤中所有的返回结果进行计算。如果测试脚本中有多个步骤（比如包含了连接和消息发布），那么这里的结果是基于这两部的测试结果计算得到。
 - i. Avg throughput 所有请求平均吞吐量
 - ii. Avg success throughput 所有请求平均成功吞吐量
 - iii. Avg failed throughput 所有请求平均失败吞吐量
 - iv. Max virtual user num 最大虚拟用户数
 - v. Avg response time(s) 所有请求平均响应时间
 - vi. Max response time(s) 所有请求最大响应时间
 - vii. Min response time(s) 所有请求最小请求时间
 - viii. Response code succ rate 所有请求成功率
 - ix. Verification point succ rate 所有请求验证点成功率
 - x. Avg request size(KiB) 所有请求平均大小
2. 图表区域
 - i. Response time 每个请求响应时间
 - ii. Throughput 每个请求所有吞吐量
 - iii. Succ Throughput 每个请求成功吞吐量
 - iv. Virtual User 虚拟用户数
 - v. Response code succ rate 响应码成功率
 - vi. Download bytes 响应数据大小



3. 详细数据 (Detailed data 表格)

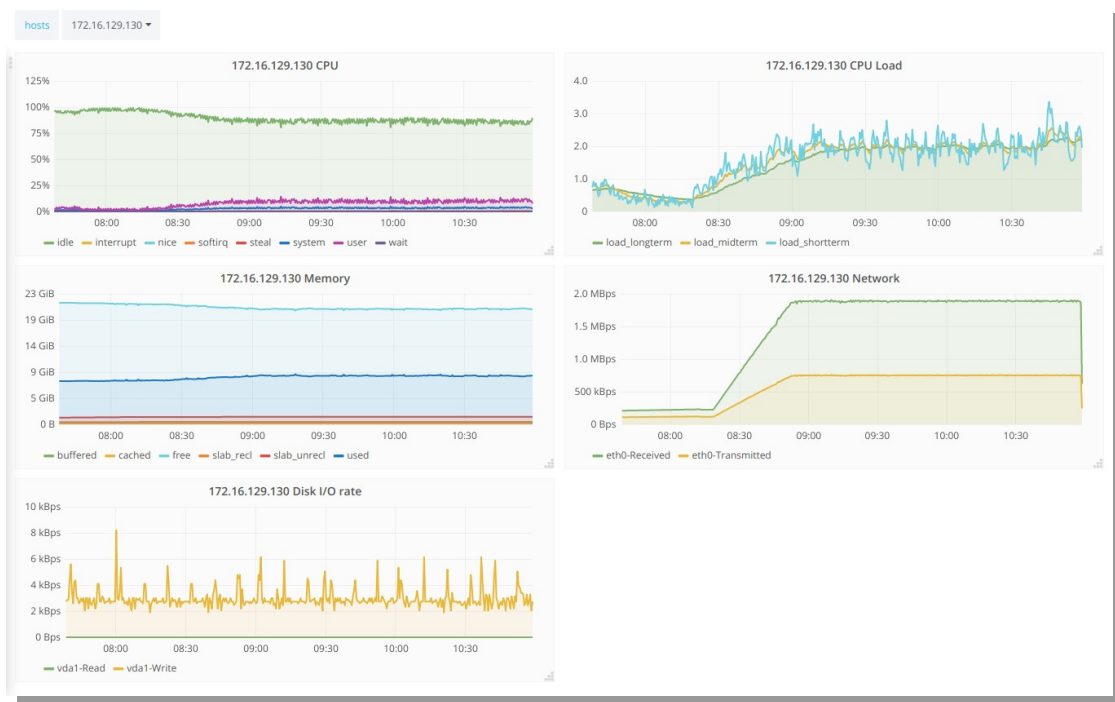
- i. Page 请求名称
- ii. Hit num 运行次数
- iii. Max resp time (s) 最大响应时间
- iv. Min resp time (s) 最小响应时间
- v. Avg resp time(s) 平均响应时间
- vi. Avg throughput 所有平均吞吐量
- vii. Avg succ throughput 成功平均吞吐量
- viii. Avg failure throughput 失败平均吞吐量
- ix. Avg request size(KiB) 平均大小
- x. Response code succ rate 响应码成功率
- xi. Check point succ rate 验证点成功率
- xii. Check point failure 验证点错误数
- xiii. Avg deviation 平均标准差
- xiv. 90th percentile(s) 90 分位响应时间
- xv. Avg 90%(s) 90%平均响应时间



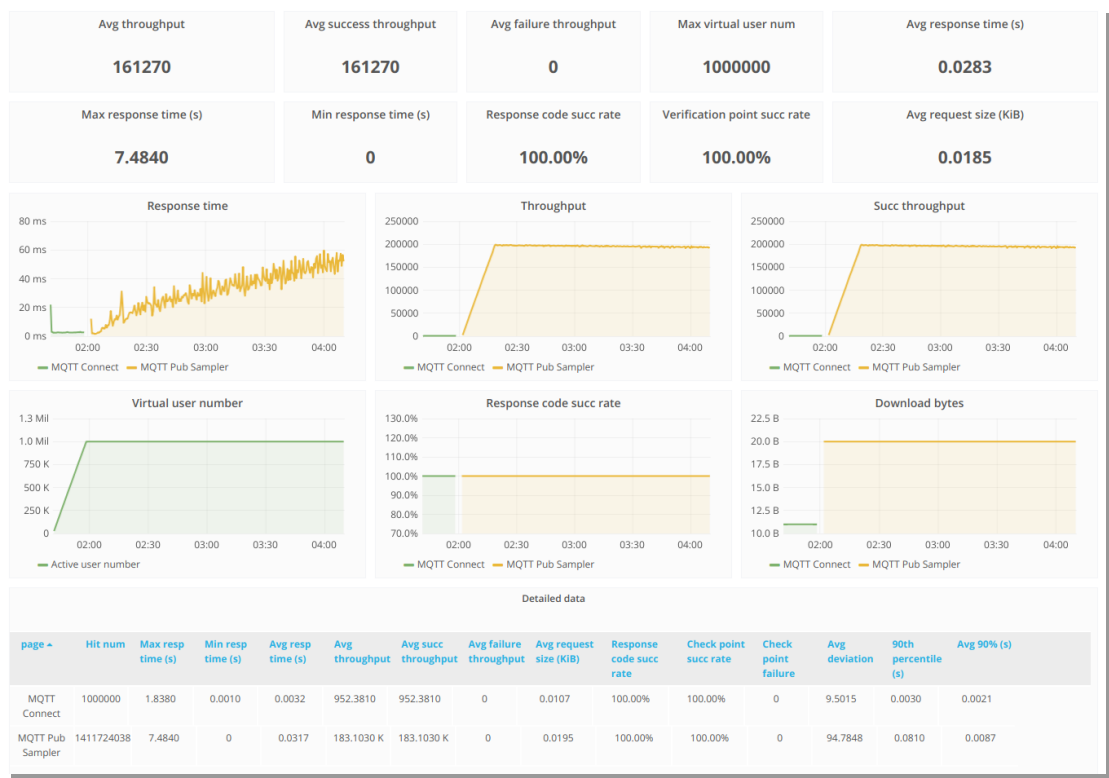
如上图所示，脚本的运行过程是在一定时间内完成连接建立（如上图绿色折线“MQTT Connect”所示），然后开始进行消息发送。测试在进入 20 万/秒吞



吐量之后，持续运行了 2 个小时。吞吐量非常稳定，所有响应都成功。



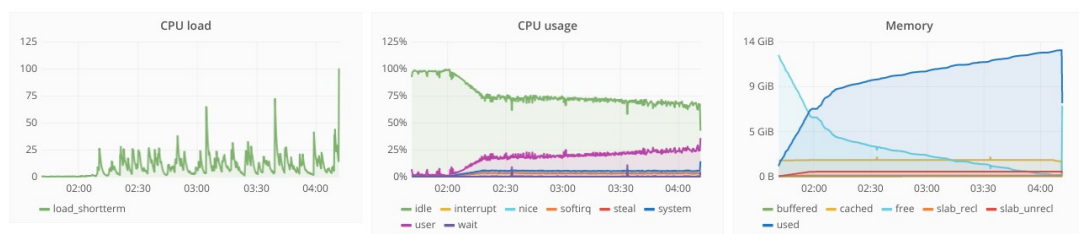
QoS 1 * 20 万/秒



如上图所示，脚本的运行过程是在一定时间内完成连接建立（如上图绿色折线“MQTT Connect”所示），然后开始进行消息发送。测试在进入 20 万/秒吞吐量之后，持续运行了 2 个小时。吞吐量非常稳定，所有响应都成功。QoS 1 情况压力机处理 ack 等待线程较多压力机 CPU 负载较高，造成响应时间统计不够稳定。

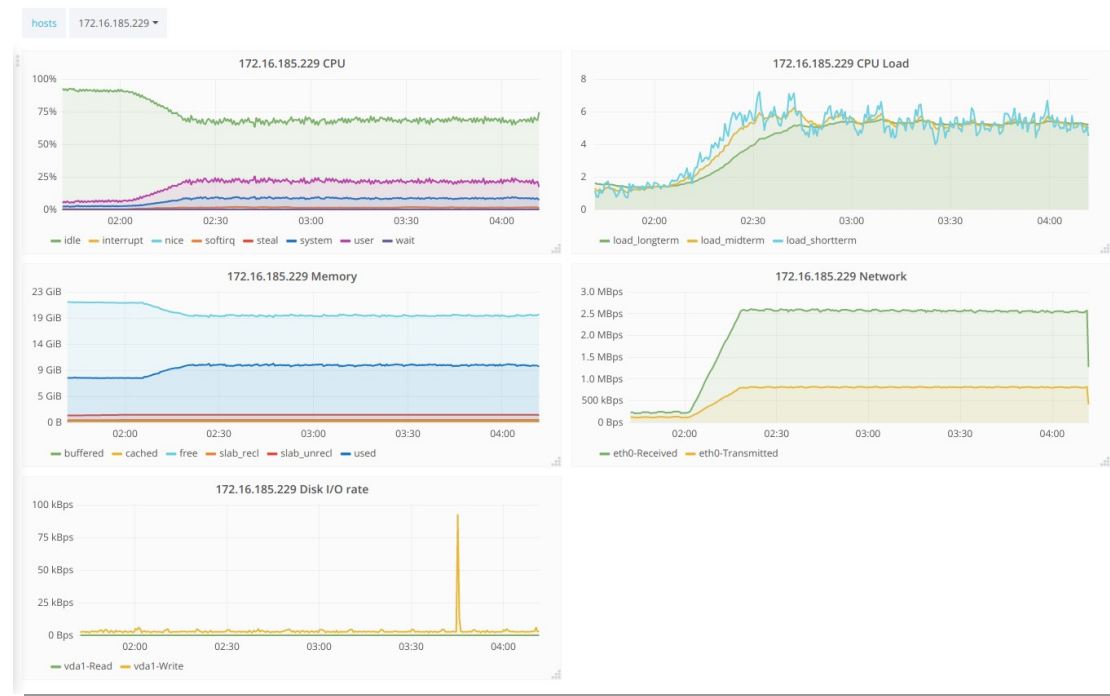
压力机监控如下：

ecs-4908



EMQX 监控：

CPU、内存和网络等表现非常稳定。



50 万主题使用测试

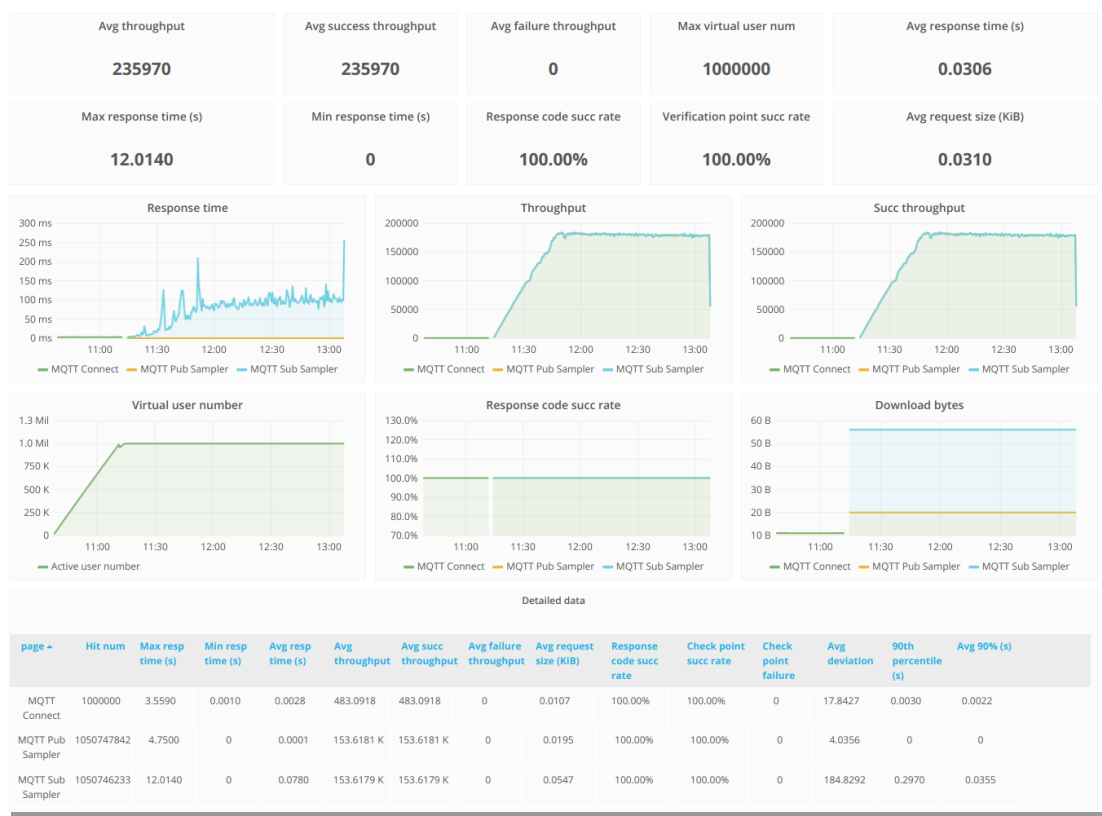
背景连接数：45 万*20 台 = 900 万

模拟 Sub 客户端数：25000*20 台=50 万

模拟 Pub 客户端数：25000*20 台=50 万

每个 Sub 客户端订阅不同 topic，同时有对应 Pub 客户端，Topics=50 万

Pub: QoS 0 * 20 万/秒 + Sub: QoS 0 * 20 万/秒

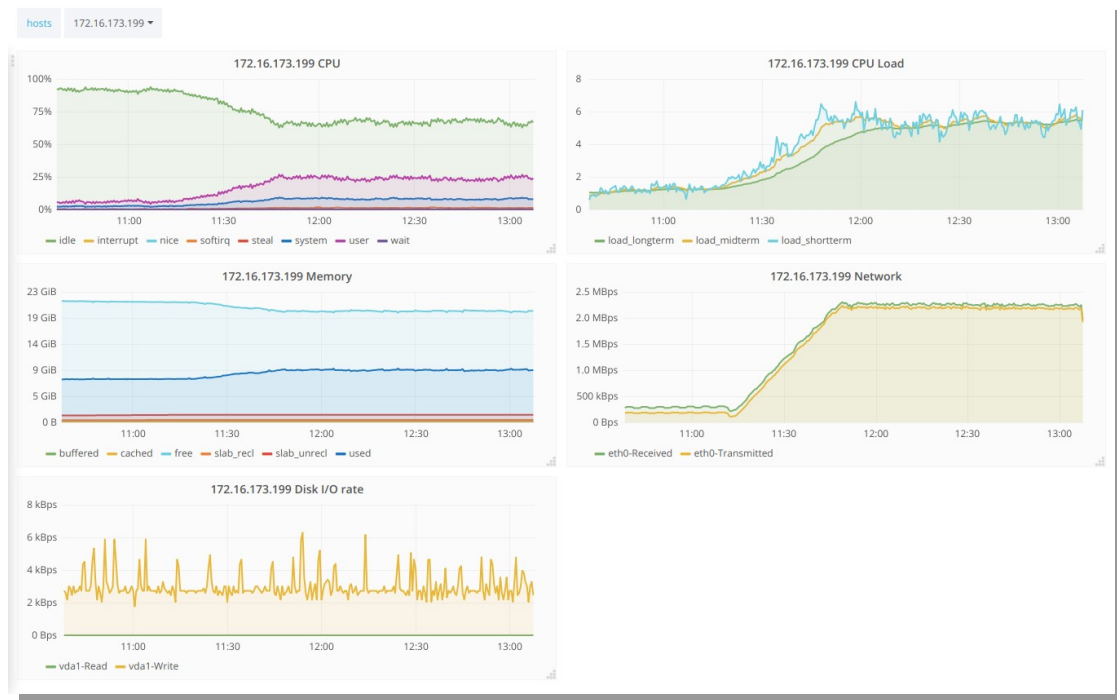


如上图所示，脚本的运行过程是在一定时间内先完成连接建立，同时完成订阅，然后开始进行消息发送。测试在进入 18 万/秒 Pub + 18 万/秒 Sub 之后，持续运行了 1 个小时。吞吐量非常稳定，所有响应都成功。

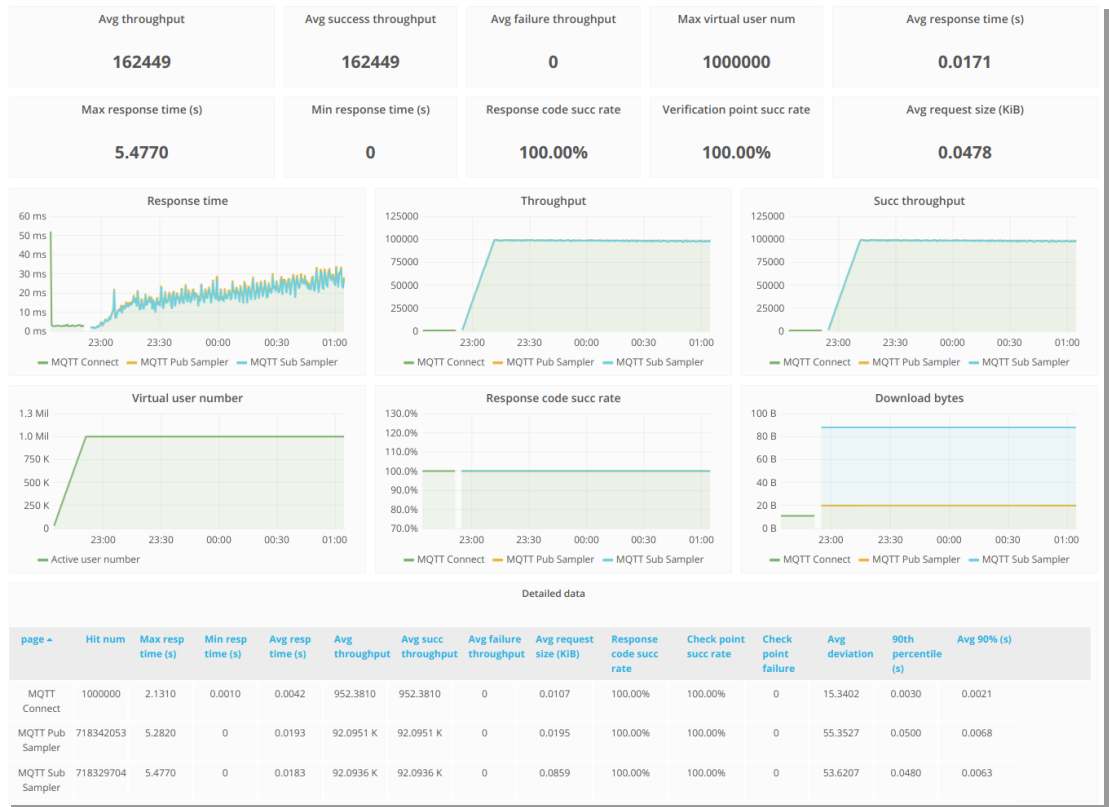
EMQX 资源监控：

CPU、内存和网络等表现非常稳定。





Pub: QoS 1 * 10 万/秒 + Sub: QoS 1 * 10 万/秒



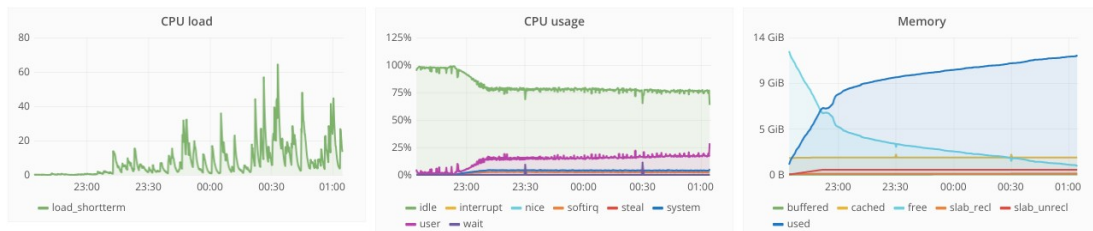
如上图所示，脚本的运行过程是在一定时间内先完成连接建立，同时完成订阅，



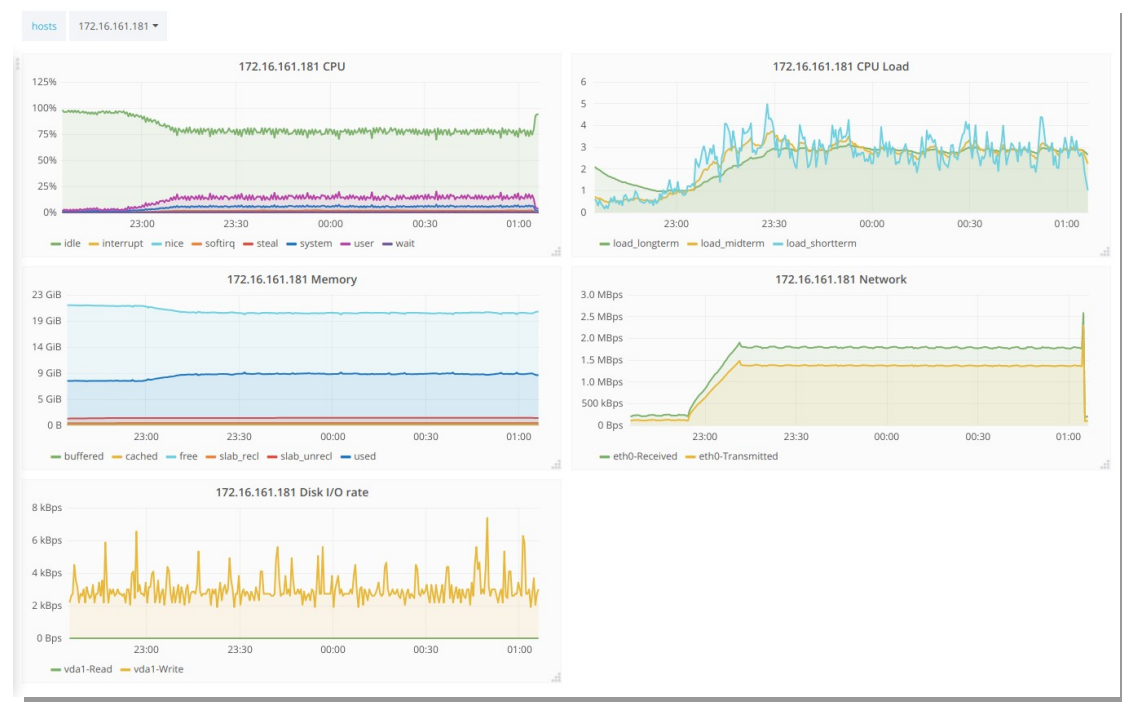
然后开始进行消息发送。测试在进入 10 万/秒 Pub+10 万/秒 Sub 之后，持续运行了 1 个小时。吞吐量非常稳定，所有响应都成功。QoS 1 情况压力机处理 ack 等待线程较多压力机 CPU 负载较高，造成响应时间统计不够稳定。

压力机资源监控：

ecs-19df



EMQX 资源监控：



消息广播测试

背景连接：45 万*20 台=900 万

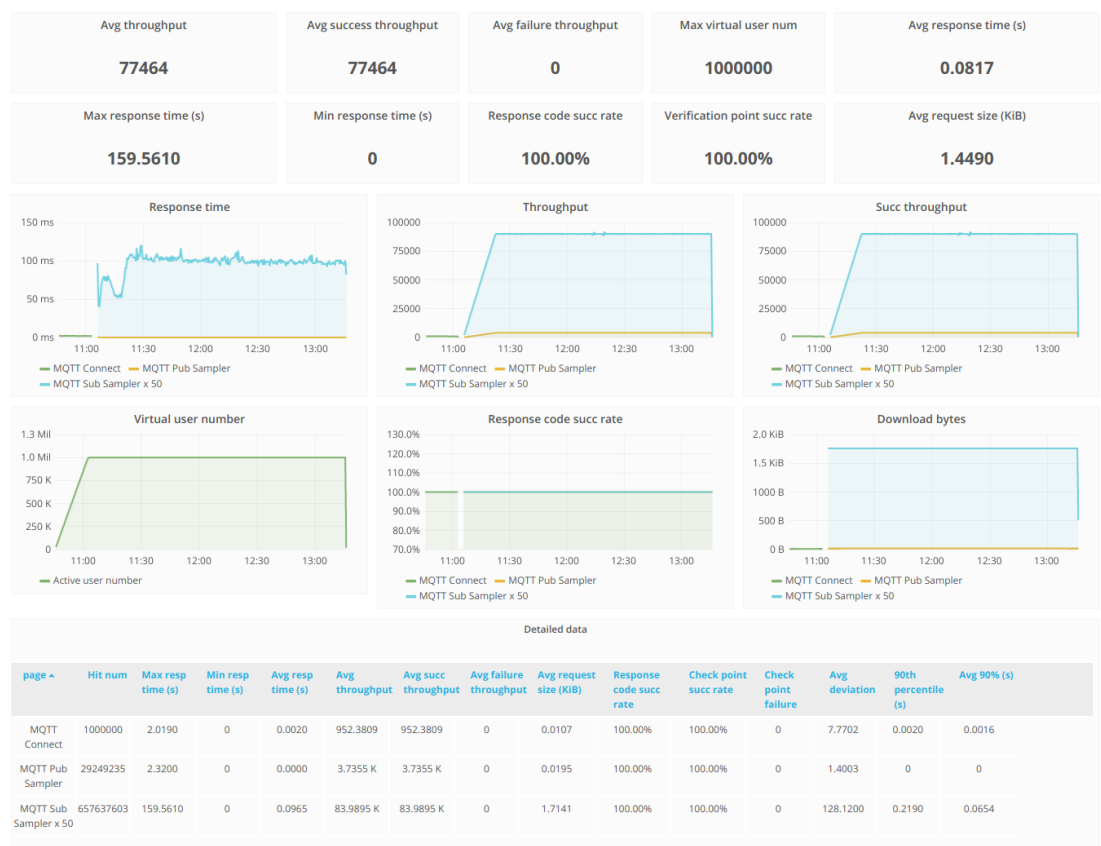
Sub 客户端：45000*20 台=90 万



Pub 客户端: 5000*20 台=10 万

Sub 客户端分别订阅 2000 个 topic, Pub 客户端分别 pub 消息到 2000 个主题,
设置不同 Pub 间隔时间控制吞吐量

Pub: QoS 0 * 4 k/秒 + Sub: QoS 0 * 180 万/秒



如上图所示，脚本的运行过程是在一定时间内先完成连接建立，同时完成订阅，然后开始进行消息发送。测试在进入 4000/秒 Pub+180 万/秒 Sub 之后，持续运行了 2 个小时。吞吐量非常稳定，所有响应都成功。**本测试 Sub Sampler 收到 20 次响应统计一次，图表显示吞吐量×20 为实际吞吐量。**

响应消息如下图：



Trace Log
Error Log
Other Logs

Trace Log

1. MQTT_Sub_Client_Fix2.2.2(total: 142)
MQTT Connect_10:45:30.855
MQTT Sub Sampler x 50_11:05:31.418
MQTT Sub Sampler x 50_11:05:31.483
MQTT Sub Sampler x 50_11:05:56.305
MQTT Sub Sampler x 50_11:05:56.338
MQTT Sub Sampler x 50_11:05:58.217
MQTT Sub Sampler x 50_11:06:21.323
MQTT Sub Sampler x 50_11:06:21.542
MQTT Sub Sampler x 50_11:06:46.332
MQTT Sub Sampler x 50_11:06:46.390
MQTT Sub Sampler x 50_11:06:46.592
MQTT Sub Sampler x 50_11:07:11.358
MQTT Sub Sampler x 50_11:07:11.565
MQTT Sub Sampler x 50_11:07:36.365
MQTT Sub Sampler x 50_11:07:36.459
MQTT Sub Sampler x 50_11:07:36.995
MQTT Sub Sampler x 50_11:08:01.316
MQTT Sub Sampler x 50_11:08:01.547

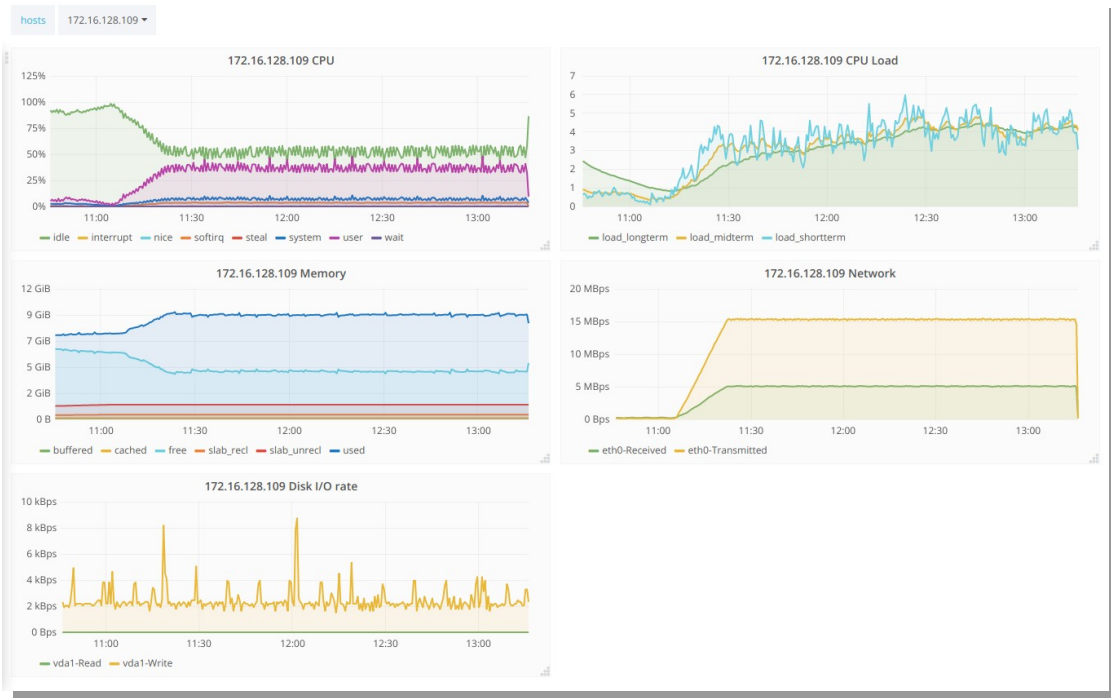
Request Body

Response Body

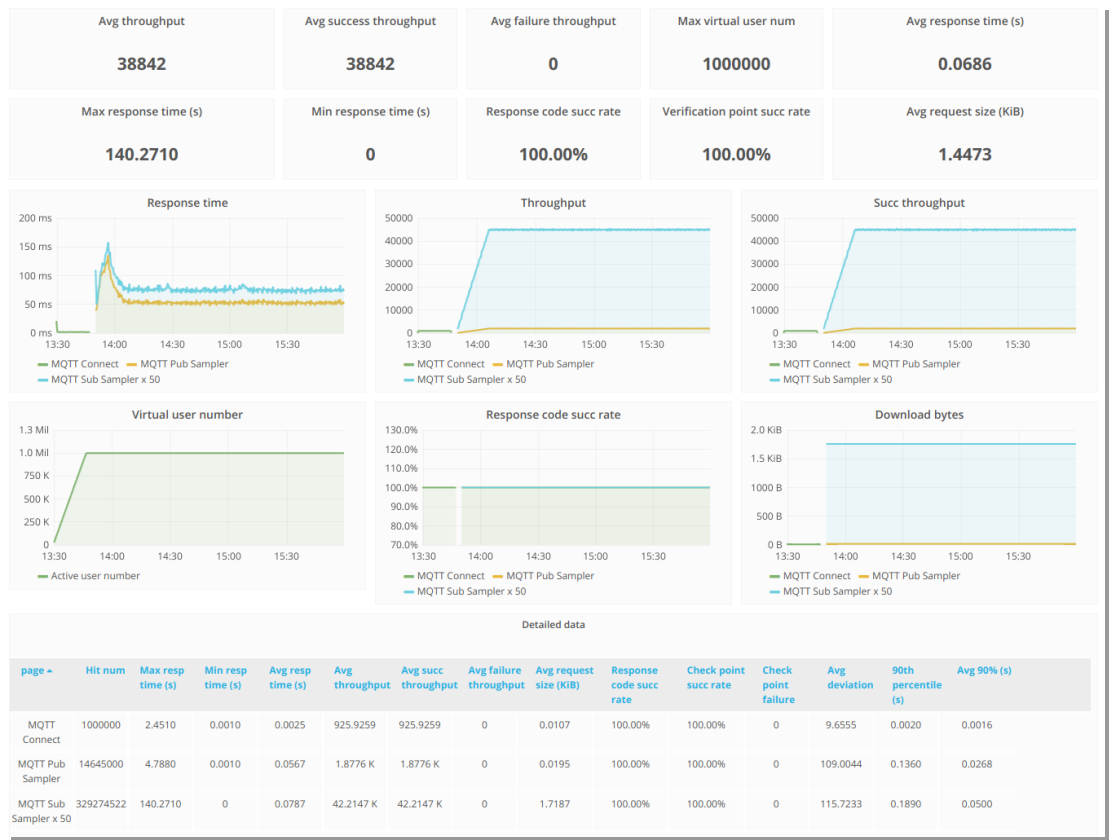
Preview

1560481531325ts_sep_flagt004k11me3m5u3848bey3m7sxsxvyju6dukurqqikema3miq34bjay8mff0hfl1g6
1560481531354ts_sep_flagleibucocv4k5dghem5ch2hu3glv646vdss6m11mqyibhjiwdvhhhgime38ff7dck
1560481531356ts_sep_flagtjivb5wd2sxbgttig85e6nt2caensay5813he5x1f34nr5nq7jn2jv3ovijpy6u
1560481531357ts_sep_flagh2flpmg7qmfjq5vidbl1ymv7aekdptbbva5i7s3dss32x30htvp05gr2es28n1nj
1560481531362ts_sep_flag03ymif82pulsbvpj1boac23eo8k7arfxq3ho577hofsej42rrkb2e51kftmaqa3py
1560481531365ts_sep_flagyg41dnya1u5egfxcclh1kebfxu137o6l8mawxn2pqvnhvucgqwovtghyrs1v6cy6c
1560481531358ts_sep_flag5007asg0e38go20u55t18dojq1xvt8ab6nlbpjpkbms2mu33b14vyr73g7fd3geq
1560481531362ts_sep_flag6iulebf6oxrpud12o11t07fip2gwipec1dk6eu2hic24k0unwmmw37d3iqx02qn
1560481531363ts_sep_flagseco6yo17oun5hrw0umseg71lylqgg10e1erxtm8a2tpuvdpbtaohu83qqcx8oo
1560481531368ts_sep_flagcmh12vs1xt0fqnukef1fgtl5mymqmm4img2do8d1ococ4j7pavxsqenl1jkn8
1560481531375ts_sep_flagx7u04inhob1hepuxu0trxsijaqel1fb2c5oxwkv0xny8kr6rxn6erj046vcd
1560481531366ts_sep_flagxs88odblnamqve1vud3k8j64pwbnamys35u3n5qh2dibvqa5liyhxy7yn2q7ghiv
1560481531393ts_sep_flagbaaf3mmvrlus2gptdeimuahm3kmsjirui0klrgynmo0jafckjbubp6ggy8fbtvev
1560481531400ts_sep_flagoewipycexovf6ihxovw2i7emarg5km6csyrvf6mhoo20p4ug3wsc8m480ahannah
1560481531402ts_sep_flagljcmo7qvh7xt8x0lrt4wuuvhwvi6xkjs0logi08u5g8016bvmv7ygmnu8uxiwo
1560481531406ts_sep_flag7x730aqc71brslgfofv4auemy4wmlfg4ppoljgm07skq6wcy302lc3600s2do2
1560481531383ts_sep_flagau7r020ao62aa3mve1nrd80b400tpm8c8aby2djxy0rk43cwyubxuld7pjowspv1
1560481531435ts_sep_flagxavdsdifecl2vdx8cqwqgximmkbnbmppwpcmetvo3x7e0e4tuagghmidj725r6sh
1560481531443ts_sep_flagtm8o4viwd3iq8egmjcmymwixcsts2sbng1qi57qds7il6lsx08idpunkwvgms
1560481531448ts_sep_flagytqk3geaix5iecpqhco4fcql8uynjeua2w1ot4oyg7usiy61gimkno284gm8r51x

EMQX 资源监控：



Pub: QoS 1 * 2 k/秒 + Sub: QoS 1 * 90 万/秒



如上图所示，脚本的运行过程是在一定时间内先完成连接建立，同时完成订阅，然后开始进行消息发送。测试在进入 2000/秒 Pub+90 万/秒 Sub 之后，持续运行了 2 个小时。吞吐量非常稳定，所有响应都成功。本测试 Sub Sampler 收到 20 次响应统计一次，图表显示吞吐量×20 为实际吞吐量。

响应消息如下图：



Trace Log
Error Log
Other Logs

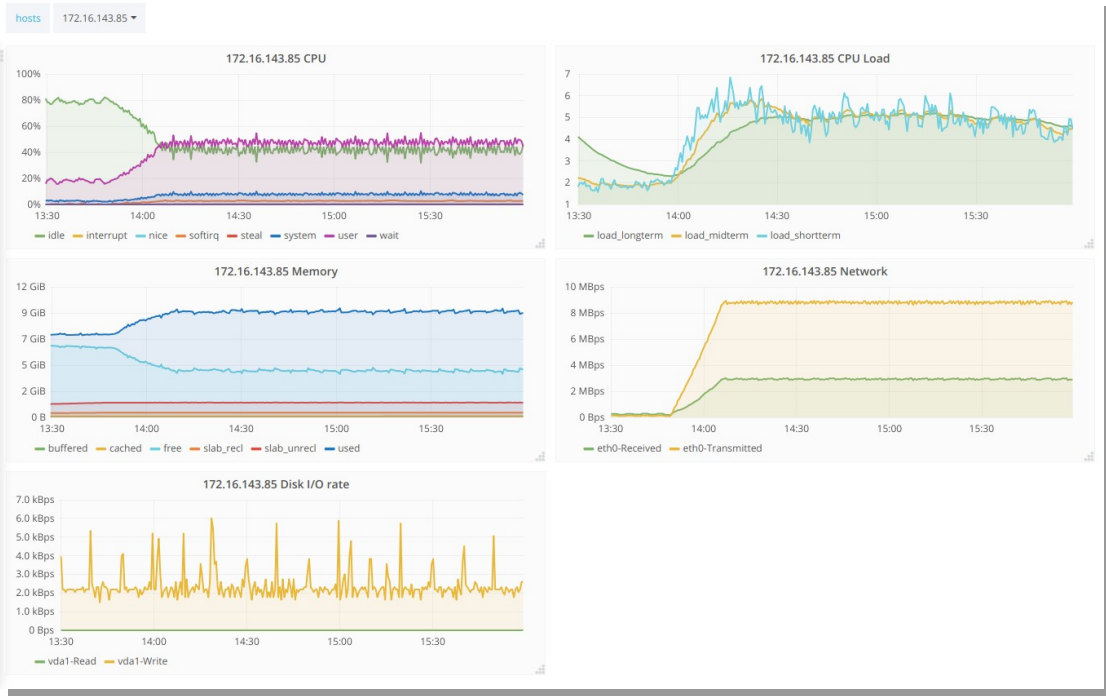
Trace Log

- 1.MQTT_Pub_Client_Fix1 1-1(total: 58)
- 2.MQTT_Sub_Client_Fix2 2-4(total: 142)
 - MQTT Connect_13:29:28.681
 - MQTT Sub Sampler x 50_13:49:30.025
 - MQTT Sub Sampler x 50_13:49:30.138
 - MQTT Sub Sampler x 50_13:50:19.964
 - MQTT Sub Sampler x 50_13:50:20.091
 - MQTT Sub Sampler x 50_13:50:20.482
 - MQTT Sub Sampler x 50_13:51:09.947
 - MQTT Sub Sampler x 50_13:51:10.344
 - MQTT Sub Sampler x 50_13:52:00.028
 - MQTT Sub Sampler x 50_13:52:00.025
 - MQTT Sub Sampler x 50_13:52:00.426
 - MQTT Sub Sampler x 50_13:52:49.954
 - MQTT Sub Sampler x 50_13:52:50.099
 - MQTT Sub Sampler x 50_13:53:40.041
 - MQTT Sub Sampler x 50_13:53:40.085
 - MQTT Sub Sampler x 50_13:53:40.418
 - MQTT Sub Sampler x 50_13:54:30.062

Request Body
Response Body
Preview

1560491469886ts_sep_flagenmrlxyv7paasjp5o028rcwuw2hjyyxip140qa3bj8p2cxvegk57v36imt0mqpl
1560491469894ts_sep_flaglvub2nq154wksshynoi556comogru7rbwkmodv18mytg72ogeivd5c5wvy4ku2dh
1560491469894ts_sep_flag37f5272unw5mub4f7i0h248dkkvblj0tsus87oub6qy5m06bsbsuhf1lmf0mtak1
1560491469894ts_sep_flagstqeikdlq1q1htvniuktyft1n1m3mlmccn4p6fycmojt51eykd7kdpbrb4xq8m5fi
1560491469902ts_sep_flagxpssepwlmg5vuj05odi65yubf0f0ivu8njkwbcnxfipobyeatbdx4krpfkk8weiv4
1560491469904ts_sep_flagtqxdcl5p8j4a34802v0rqsn8ypdce8b6s5ttmpu1c5n01usfirho7fgrhunyej4d
1560491469904ts_sep_flagn8hhullm4qt0lnefthfmacjkmppa1mfukfbama1yxv85gku3ertfmbcl1ci7msu4
1560491469906ts_sep_flagou5hpxwefput1u2xlr4ku6rpsjheiujsp01q025msghdq4jj7g4k7mrnb8wxwro
1560491469906ts_sep_flag6sl820rb78nege1y667m1k52o05wxpe2f6p2u68plmbbphv17018f0uatmai4x5
1560491469907ts_sep_flagrudwp4b2v4f5yeelt13vgo7xetomueplu7catw1k6wb0gamqm210o32d3l85m7mb
1560491469909ts_sep_flagcqi4s2igcm6qt82fsd2ymr4rbjmkv1xv6oduxg7fegoyi5c3ue5ytcym8eipeh6
1560491469910ts_sep_flagqf2wcj3ncg643qcvwb22ftklaorarmju6ncbddy2dl6sxmso173y0f7blgvpm
1560491469910ts_sep_flagiybnxmgm5drmt0i8ikab0eup2j3v3n378n5bvkdkpyirkx8nneci563dujdmn5n
1560491469916ts_sep_flagwpryorqm65tuah7m4nv31fgl5qv40xaumi1a0h47vllrk1nc1ug2ipv5pc1t4e
1560491469910ts_sep_flaghnrn6t0og8qpsmsf70kqmx3uqhsijifbeqiwbqceu1yu6s7m6plh14njpcydyb7
1560491469917ts_sep_flag7mbm1scr1aeakqm38ug0r5vlnvluoi2h2ymjueim4qxkivxlldkmoisc4wobck
1560491469921ts_sep_flag4vhi4aebnj3hnb8cyjmq7pq1vy7p2vnuwislg23klxm53nkhe5kfm2mupr8ybyg
1560491469918ts_sep_flag5spt60epwpyamvxvhtor5mmvwp15u00vc8iuthmvehjwdkpgm6mhyd0h3lqma3
1560491469922ts_sep_flag6goth1bgpmkhn2yh6dulx4yjbhxm0sf0l7ch60ep4d6u2pmqa3miv5yn45ubycyb
1560491469922ts_sep_flag0avu4vcm8gwa2migx6lm3v30wfgun7hh86m7h3wysqisxagjtmj062rfy2sl6uy

EMQX 资源监控：





附录

系统优化

参考：<https://developer.emqx.io/docs/broker/v3/en/tune.html>

测试脚本

测试脚本：script.zip





网站: <https://www.emqx.io>

电话: (+86) 18058747908

邮箱: contact@emqx.io

总部: 杭州市余杭区龙园路 88 号 3 幢 A303

分部: 硅谷, 北京, 法兰克福, 昆明, 重庆, 深圳