EMQ X & HiveMQ comparison report

Overview 2

Highlights 3

Features 4

EMQ X Broker vs HiveMQ Community 4

EMQ X Enterprise vs HiveMQ Professional/Enterprise 5

Performance 7

EMQ X Broker 10M connections & message throughput test 7

HiveMQ 10M connections & message throughput test 9

Performance result summary 11

June, 2019

Overview

物联网的概念早在1990年代就被提出，但是大规模的物联网实际应用则是在互联网全面普及之后，伴随着人工智能 、无线数据网络、云计算、大数据、虚拟化等多项技术的成熟一起到来的。自2010年起，物联网越来越多地从理论走向应用，作为物联网灵魂的消息中间件也得到了蓬勃的发展。经过数年发展，从一开始的多家竞争，到现在的少数几家占据主要市场，消息中间件也经历了多次迭代。目前商业消息中间件主要有两家，分别为 EMQ 和 HiveMQ。EMQ 主要消息中间件产品为，

* EMQ X Broker （开源）
* EMQ X Edge （开源）
* EMQ Enterprise （商业）

而 HiveMQ 的主要产品为，

* HiveMQ Community（开源）
* HiveMQ Professional（商业）

本对比将在产品的定义、功能、性能和架构等多个维度上进行，为您提供两个产品在多方位的比较。

# Highlights

|  |  |  |
| --- | --- | --- |
| EMQ X 拥有压倒性的社区成熟度 | EMQ X 在物联网协议支持度上遥遥领先 | EMQ X 在支持同级别高并发、低延时所需的资源只需 HiveMQ 的 50% |
| EMQ 项目从2013年开始开源，累计下载量达到了200万次，版本迭代超过了上百次，是开源社区中最流行的物联网消息中间件。HiveMQ在2013年成立之后，一直以闭源方式向客户提供软件，直到2019年才开源。 | EMQ supports the most popular IoT protocols, such as MQTT, CoAP, LwM2M, MQTT-SN, and all kinds of industrial protocols, such as ModBUS. While HiveMQ only for MQTT protocol. | EMQ X requires only half number of servers comparing to HiveMQ in 10M connection benchmark testing. |

|  |  |  |
| --- | --- | --- |
| EMQ X 提供边缘到云的全栈物联网接入方案 | EMQ X 内置的丰富数据持久化支持，快速集成至后台大数据处理系统 | EMQ X SQL based rule engine 将应用开发的效率大幅提升 |
| EMQ X can be deployed at any resource constrained devices. It includes all kinds of ARM/x86 based servers, such as industrial server & Raspberry Pi. The offline data storage can reliably caching of data locally when the network is unavailable and synchronizing data to cloud services once back online. Comparing to Hive, EMQ provides capability of building a more flexible and economic IoT solution. | Most of popualr database & streaming system integrations can be used out-of-box, such as MySQL, PostGre, MongoDB, Redis, OpenTSDB, Apache Kafka and Apache Pulsar. It's not necessary to make extra development effort to support. | A powerful built-in rule engine with one-stop data extraction, filtering, and transformation supports the flexible integration of various SQL, NoSQL, TSDB databases and Kafka, Pulsar, RabbitMQ or other existing external systems, and enables rapid enterprise integration and sustainable innovation. It can greatly improve development efficiency. |

# Features

由于 HiveMQ 目前暂无针对边缘计算的产品，因此 HiveMQ 无法提供在边缘端的解决方案，本报告中将 EMQ X Edge 排除在比较之外。将 EMQ X Broker 对应 HiveMQ Community，并将 EMQ X Enterprise 对应 HiveMQ Professional / Enterprise 进行比较。

EMQ 和 HiveMQ 的全系列产品都支持 MQTT 3.1，3.1.1 和 MQTT 5 协议。协议特性（如 QoS 支持，共享订阅等）不在比较中单独列出。

## EMQ X Broker vs HiveMQ Community

共同点：

* EMQ X Broker 和 HiveMQ Community 都是开源软件，两者都遵从 Apache 2.0版权协议，都可以从 Github 上获得并自由使用。
* 两者都支持 MQTT 3.1， 3.1.1 和 MQTT 5。
* 都支持用户扩展。

主要不同点：

* EMQ X Broker 支持集群模式，而 HiveMQ Community 不支持

集群模式将多个对等的消息中间件节点组织在一起集中对外服务，除了能快速简单的扩展服务能力外，还提供了设备故障时的能力冗余。单个节点故障并不构成整体服务的中断。在单个节点故障后，只要集群有足够等容量冗余，剩余节点可以接受故障节点上的客户端的连接请求并继续提供服务。

HiveMQ Community 不支持集群模式使得用户在生产环境使用它时必须非常谨慎的设计高可用性方案，相当大程度的增加了用户的应用设计和实际部署难度。

* EMQ X Broker 支持 Epoll，而 HiveMQ Community 不支持

Epoll 是 Linux 下一种高效的多路复用的网络 IO 模型。支持 Epoll 将使吞吐性能得到较大的提升。

* 随软件安装，EMQ X Broker 同时提供文件和数据库等方式的 authentication 和ACL，HiveMQ Community 只支持文件方式。

EMQ X Broker 支持数据库方式的 Authentication 和 ACL，管理方便。对于客户端的权限控制，只需要在数据库表或键值对做简单的crud操作即可。同时，在数据表项中支持变量和通配符，规则设计简单。外部数据库形式的鉴权管理也方便用户在应用系统中统一管理，降低各应用部件之间的侵入性。目前 EMQ X Broker 支持的外部鉴权接口有：

* PostgreSQL
* MySQL
* Redis
* MongoDB
* JWT
* HTTP
* LDAP

HiveMQ 只支持文件方式的鉴权管理配置。如果需要支持更高级的鉴权管理则需要实现 OnAuthenticationCallback 接口。需要客户有 Java 开发能力，增加了应用开发的复杂度。鉴权接口的安全性和效率完全取决于客户的开发能力。

* 连接支持

两者都在TCP，WS 和 TLS 之上支持 MQTT。除此之外 EMQ X Broker还支持MQTT-SN，CoAP，Stomp，LwM2M等在物联网中常见的协议。

## EMQ X Enterprise vs HiveMQ Professional/Enterprise

HiveMQ在两个商业版中支持上述开源版中一些没有的特性包括：

* 集群
* Epoll

依然不支持：

* 数据库方式的鉴权
* 除MQTT over TCP/SSL/WS之外的连接方式

共同点：

* 都支持统一的集中管理

EMQ 提供了 EMQ Control Center, HiveMQ 提供了 HiveMQ Control Center。

* 都支持第三方监控工具

EMQ X Enterprise 和 HiveMQ Professional/Enterprise 都提供了集成到 Prometheus的方法并提供可 Grafana 界面。

不同点：

* 支持基于 SQL 的规则引擎（Rule Engine）

基于 SQL 的规则引擎给予 EMQ X Broker 超越一般消息中间件的能力。除了在接受转发消息之外，规则引擎还可以解析消息的负载，由消息的订阅，发布，确认的事件触发，根据消息的负载来执行相应的动作，极大的降低了应用开发的复杂度。EMQ Enterprise 的高级规则引擎还提供了基于历史消息统计触发动作的功能。可以依据过去一段时间内的消息统计触发动作。而 HiveMQ 不支持该功能。

* EMQ X Enterprise 支持数据持久化至数据库

EMQ X Enterprise 可以将消息数据持久化至外部数据库，提供能更可靠的持久化，并方便应用程序对数据的处理。

HiveMQ professional/Enterprise 只支持本地持久化。

* 对外数据接口、桥接

在互联网应用中，数据被生成，采集，交换，存储和处理。传统意义上的消息中间件负责消息的交换。消息中间件常常需要和其他应用交互数据，在用户的实际应用中，消息中间件经常会和其他应用组件一起部署，丰富的对外数据接口可以简化应用开发，提高应用性能。

EMQ X Enterprise支持以下接口和桥接：

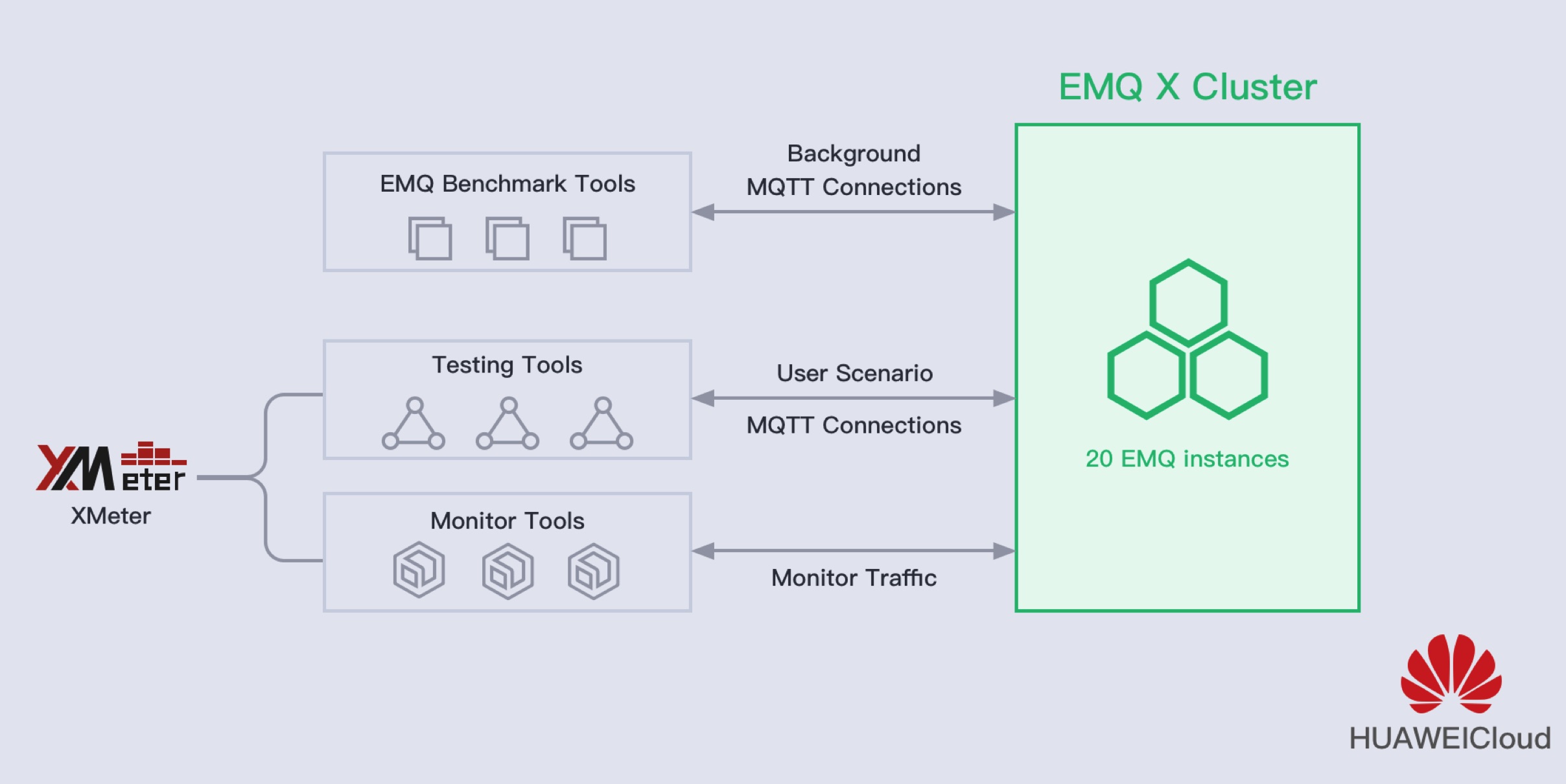
* RabbitMQ
* RocketMQ
* Kafka
* OpenTSDB
* InfluxDB
* TimescaleDB （开发中）
* AWS IoT
* 其他支持MQTT的中间件的桥接
* HiveMQ支持：
* Kafka
* InfluxDB

# Performance

在性能上，EMQ X 和 HiveMQ Professional/Enterprise 都支持千万级别的连接。我们以双方的测试报告为基准做一个比较。

## EMQ X Broker 10M connections & message throughput test

该测试由EMQ，XMeter 和华为三方共同完成，部署环境为华为云。



环境：

* EMQ X 集群：EMQ X 3.1.1， 20台 8核16G （16核32G）（暂缺配置和场景对应）
* 压力机：70台8核16G
* 测试软件：背景连接使用EMQ Benchmark Tool， 消息pub/sub使用XMeter 2.0.1
* 消息大小： 88字节

场景与结果：

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%
* 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 | 5%-10% | 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
* 平均内存试用在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。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 | 0.0353s | 25%-30% | 9GB |
| 100 | 1 | 5.477s | 0.0183s | 0.0063s | 20%-25% | 9GB |

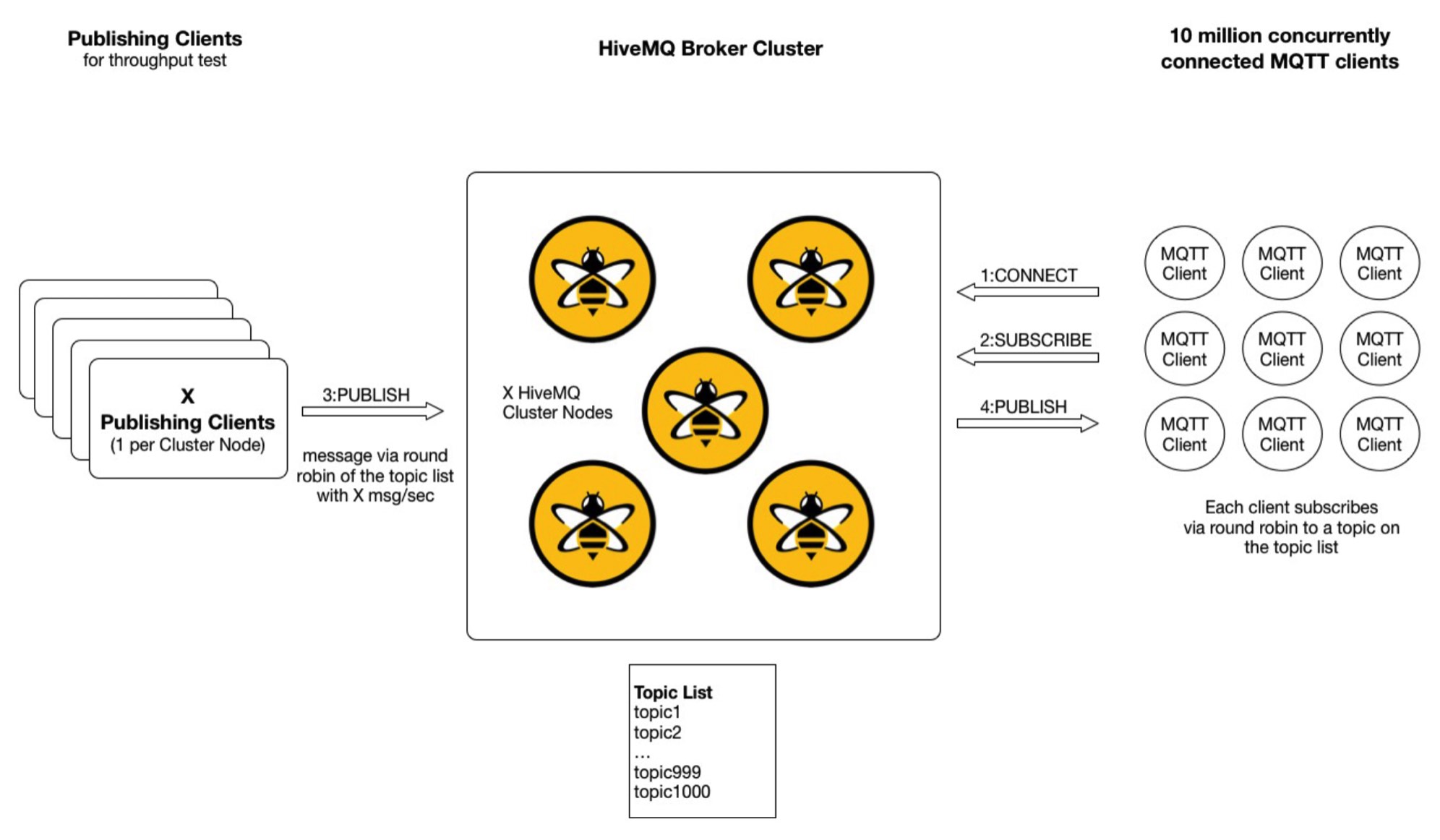
消息吞吐测试类型3： 消息广播测试

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 端在监听相关的主题，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 |

## HiveMQ 10M connections & message throughput test

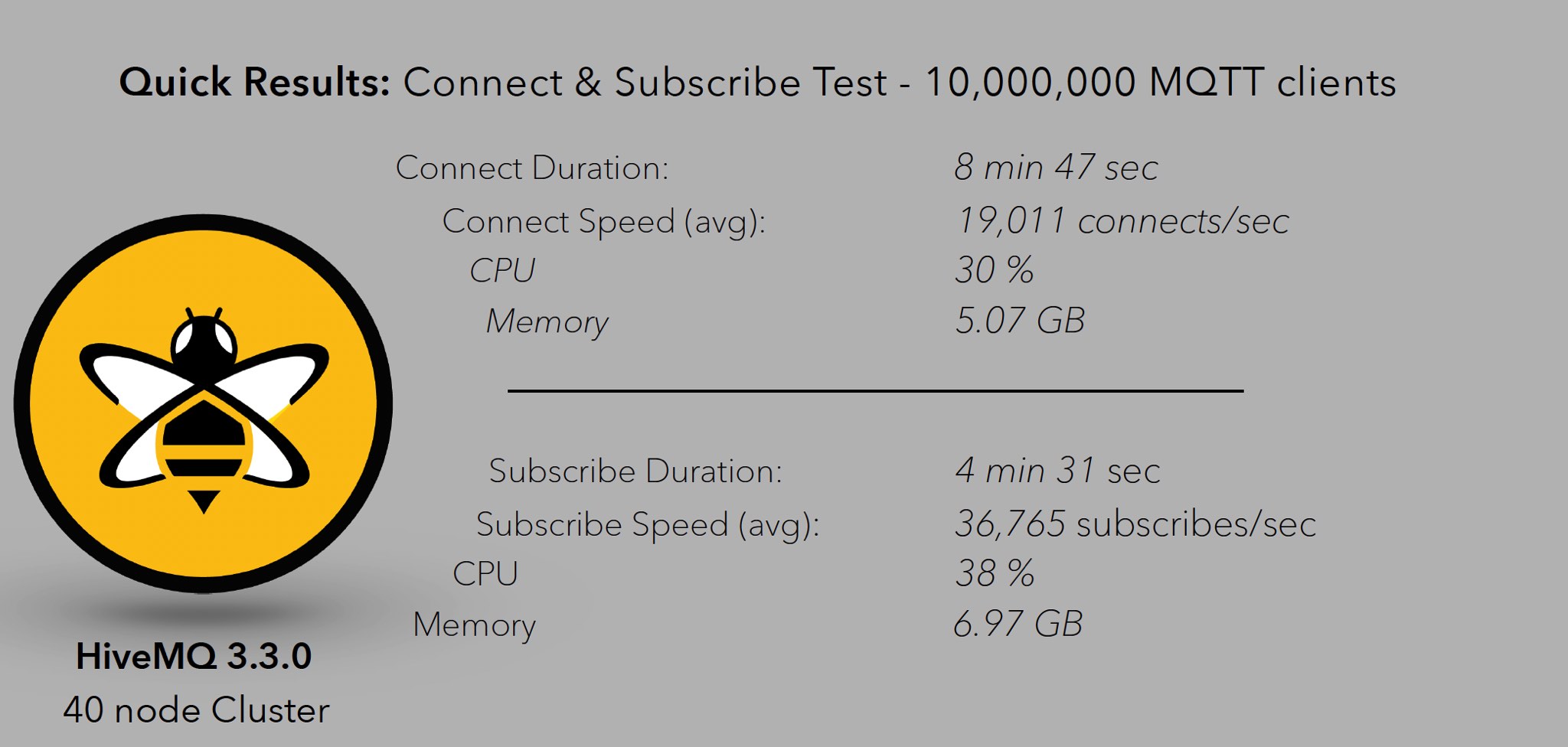
测试由 HiveMQ 完成，部署环境为 AWS。

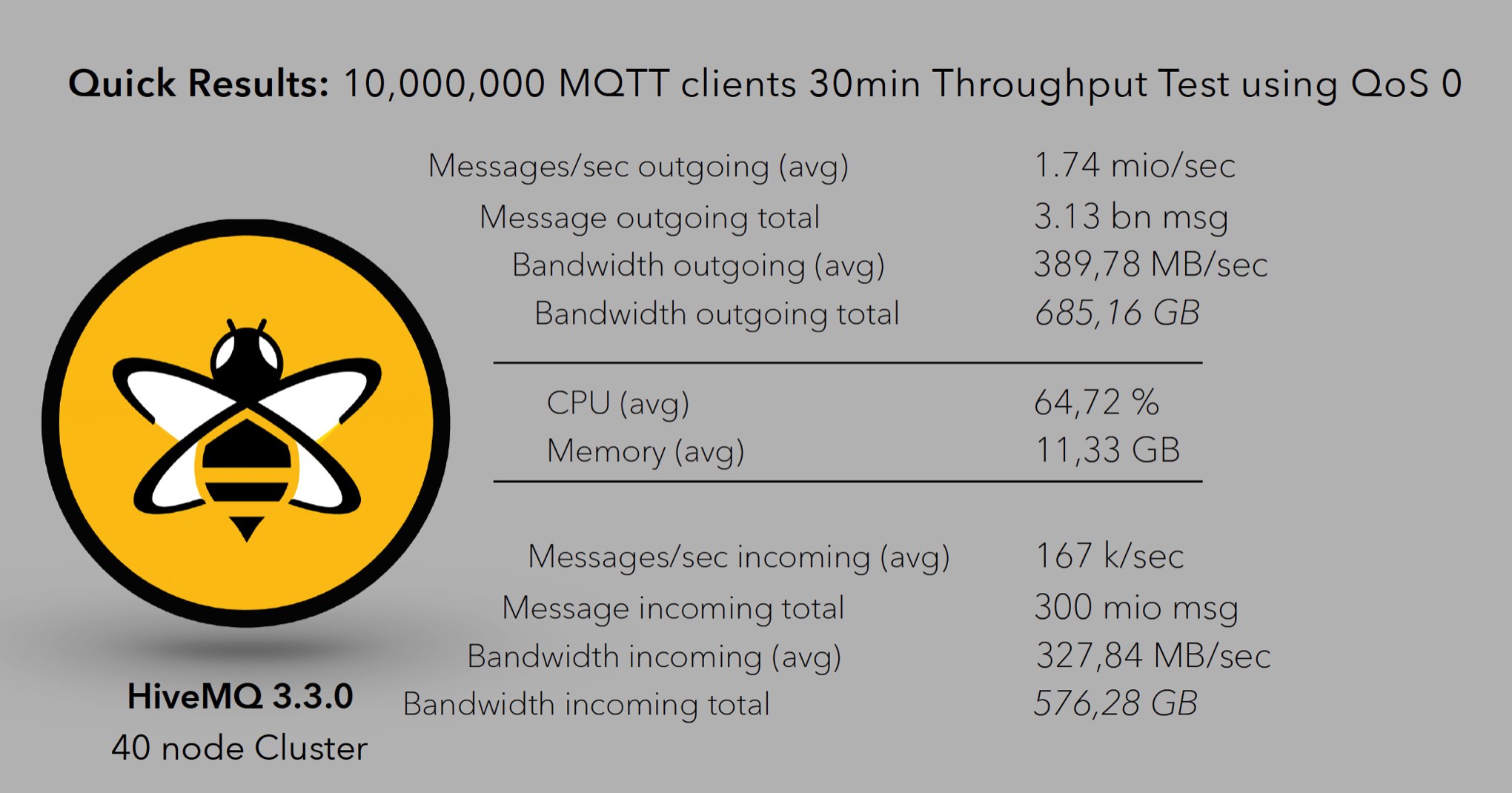


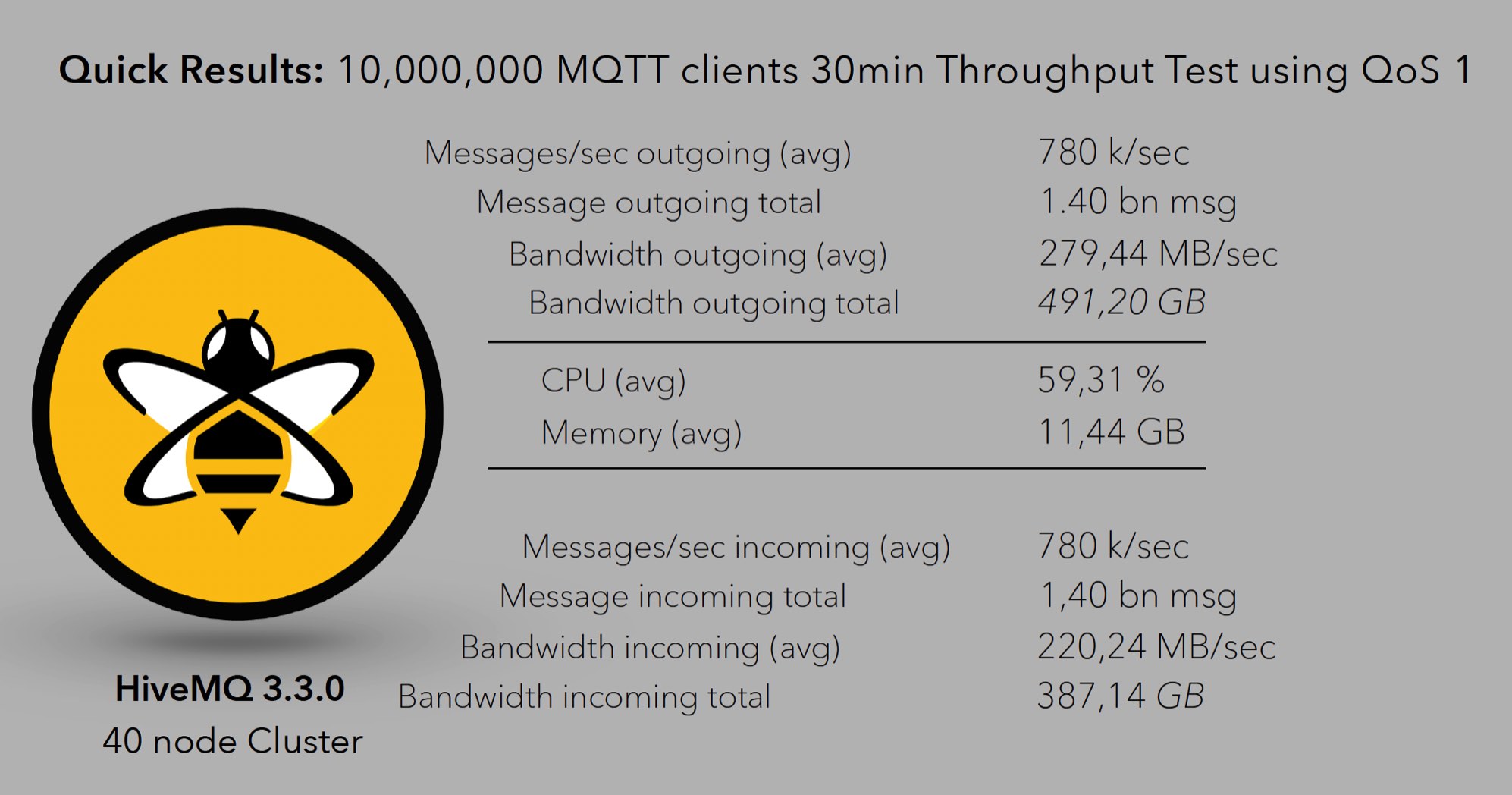
环境：

* HiveMQ 集群：HiveMQ 3.3.0， 40台 8核16G
* 压力机：40台8核16G
* 测试软件：HiveMQ Benchmark Tool
* 消息负载大小： 32字节

场景与结果：







## Performance result summary

由于测试并非由同一家组织进行，测试设置和测试项不能做到完全的1:1。但通过测试，我们依然可以得出一些结论：

由以上两个测试的对比，客观的说，EMQ X 集群在性能上优于 HiveMQ。EMQ X 使用了比 HiveMQ 少一半的集群节点（20 vs 40），使用更大的消息负载（88字节 vs 32字节），使用更多的消息主题（50万 vs 1000）；支持更大的消息吞吐量（QoS0：180万/秒 vs 160万/秒， QoS1: 90万/秒 vs 80万/秒）

在HiveMQ的测试设置中只使用了1000主题的fan-out场景，不能覆盖实际应用中的所有情况，而根据HiveMQ 提供的测试脚本，在这个fan-out场景中也缺乏同主题的消息在节点间路由的情况，这在实际应用中可能也不经常见到。