PUSH TECHNOLOGY

Diffusion™ 5.0 Performance Benchmarks

Contents

1 Introduction

Customers today demand that mobile applications are as rich and responsive as desktop-based applications. As a result, businesses must meet that demand for tens, even hundreds of thousands of users, at the same time.

Historically, Web applications were designed to fetch data from backend systems when requested by the user, or at some pre-defined point in time. This approach often stretches networks and enterprise applications to their limits, resulting in a poor user experience.

Diffusion 5.0 (Diffusion) from Push Technology employs the opposite approach. Data is pushed out to mobile applications, as it changes, in a highly scalable and efficient manner. Only the data that actually changes is sent and attention is paid to the capabilities of the mobile devices running the Web application and the networks over which the data is sent, to ensure the best possible performance to every end user.

Diffusion is used by some of the world's leading eGaming, financial, broadcast and media, utilities, and transport and logistics organizations to ensure competitive advantage, drive real-time revenues and enhance customer intelligence and engagement.

This report details the findings from a benchmarking initiative to demonstrate the significant performance and scalability of Diffusion, and explains why it is the most effective data distribution technology on the market today. Conducted by Push Technology in May 2014, the benchmarking study used commodity, off-the-shelf hardware.

2 Benchmark Overview

Two technical benchmarks were undertaken to determine the performance profile of a single running instance of Diffusion. The first, a throughput benchmark, is designed to stress the server and demonstrate how many messages per second Diffusion can publish on a single machine and to how many clients. The second, a latency benchmark, is designed to demonstrate low latency between servers and clients. Both benchmarks were performed with message sizes of 10 and 125 bytes reflecting the message sizes seen in production deployments.

The throughput benchmark demonstrated Diffusion scaling linearly up to 87,000 clients at a sustained rate of up to 15 million messages per second (i.e. 172 messages per second per client).

The latency benchmark demonstrated average latency down to sub 100 microseconds.

The following sections detail the benchmark methodology and results, and present an overall conclusion.

3 Methodology

Two technical benchmarks were undertaken to determine the performance profile of a single running instance of Diffusion. The first, a throughput benchmark, is designed to stress the server and demonstrate how many messages per second Diffusion can publish on a single machine and to how many clients. The second, a latency benchmark, is designed to demonstrate low latency between servers and clients.

The server was constrained to two network interfaces used for the test runs. The specification of the test servers are shown in Appendix A.

The following constraints were applied during the benchmarks:

1. Only the IETF Websocket protocol was used to transport data. Diffusion provides support for further transports, but the Websocket protocol is most commonly used by competing offerings, providing for a more valid comparison.
2. The same configuration of Diffusion was used for both benchmarks. This configuration is the out-of-the-box default configuration and is neither optimized for throughput, nor latency. The only change made was to match the number of multiplexer threads to the number of hardware threads available. Finer grained configuration for the respective benchmark improves the results, but has no bearing on the overall characteristics.

3.1 Throughput

Overview

The throughput benchmark model is designed around a number of configurable dimensions.

The test is run repeatedly from a cold start of a Diffusion server instance and for a fixed duration of five minutes. Each benchmark run uses a different message payload size.

Client connections are added continuously for the duration of each run. Clients are ramped at a fixed interval of five seconds. All new clients being connected in an interval are started simultaneously. This exaggerates disconnections and adverse conditions in a controlled way. Each client subscribes to a fixed set of 50 topics.

Benchmark Duration	- 5 minute runs
Message Payload Sizes	- 10 and 125 bytes
Ramping Interval	- 5 seconds
New Client Connections Per Interval	- 750 per interval
Messages Per Second Per Client	- 200 messages

3.2 Latency

Overview

The latency benchmark is a simple round-trip benchmark. A single client is used. The client sends a request message to the server. The request is received by and responded to by the Diffusion server with a response message. The client receives the response message and the round-trip latency is measured. This process continues for the benchmark duration.

This benchmark therefore represents a best-case latency profile.

Benchmark Duration	**- 5 minute runs**
Message Payload Sizes	**- 125 bytes**

4 Results

The benchmark results show that Diffusion scales linearly to high numbers of connected clients at high data rates while maintaining low message latency. In fact, the tests highlighted that hardware and network resource was the limiting factor on performance.

Even when resource capacity was reached, the benchmark tests were successfully completed demonstrating Diffusion's resilience.

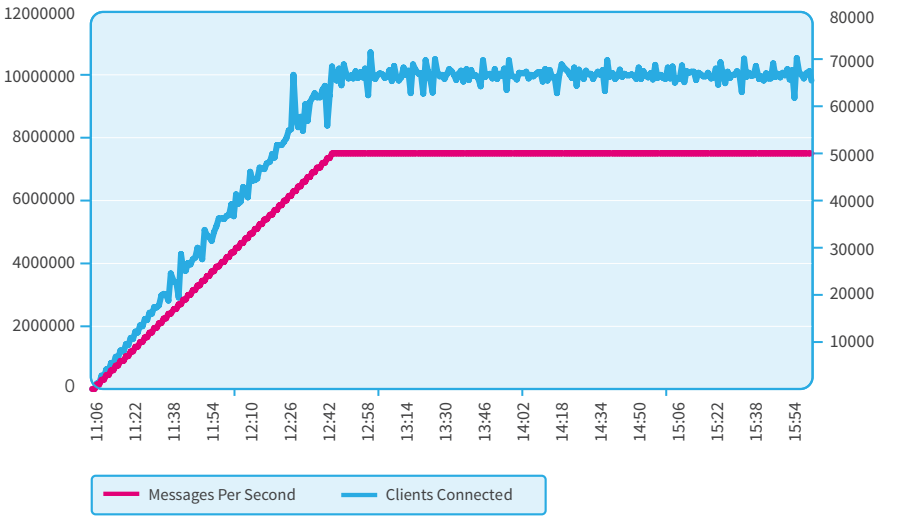
4.1 Throughput

The throughput profiles generated by the execution of the benchmarks are shown in the following charts.

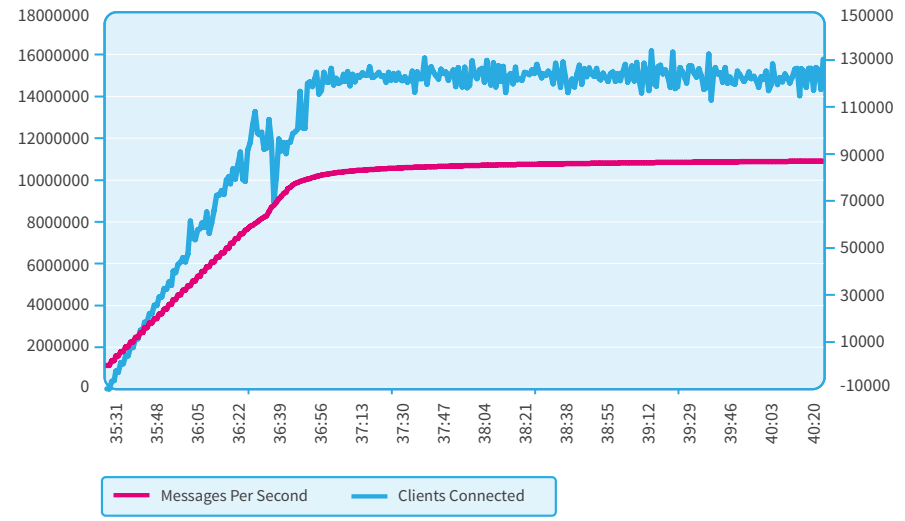
For 10 byte messages, the test ran to completion without incident. Full linear scaling was achieved as both clients and messages per second were increased. The runs ended with Diffusion servicing 50,000 clients at a sustained rate of 10 million messages per second (i.e. 200 messages per second per client).

The next test run (125 byte payload), resource saturation occurred. Full linear scaling was again achieved until resource saturation, which occurred at approximately 85K clients. The runs ended with Diffusion servicing 87,000 clients at a sustained rate of 15 million messages per second (i.e. 172 messages per second per client).

Throughput for 10 byte Messages, up to 50,000 Clients on 1 Client Machine



Throughput for 125 byte Messages, up to 87,000 Clients on 2 Client Machines



Diffusion is designed to minimize the impact on service levels to already connected clients when events, such as network saturation occur via the graceful disconnection of unresponsive clients. In the 125 byte test, it can be seen how resource exhaustion causes the number of connected clients to level off, but all clients continue to receive messages.

5.2 Latency

The following histogram shows round-trip, client-Diffusion-client latency statistics: The minimum, average, 99th and 99.99th percentiles.

Minimum	Average	99% Percentile	99.99% Percentile
31.44µs	43.28µs	41.92µs	53.12µs

The results show that Diffusion delivers low round-trip latency, even as the number of messages delivered and clients connected ramp up.

5 Conclusion

Diffusion has been built from the ground up to provide a highly scalable and reliable data distribution platform that excels in delivering data to applications on internet connected devices together with its ability to work well with proxies, firewalls and load-balancers.

This report demonstrates Diffusion's high performance characteristics in terms of its ability to reliably handle high volumes of data and connections and keep latency to the bare minimum. This means that not only is Diffusion suitable for intelligently distributing data to mobile and Internet devices and browsers, but it is also suitable for distributing data to enterprise applications written in Java, C# or Web applications deployed on the enterprise intranet.

The benchmarks documented in this report were executed using the Websocket transport. This decision was made in order that the performance numbers produced may more easily be compared to competing offerings, the majority of which only support Websocket. Diffusion, on the other hand, supports a number of other transports and will gracefully transition between them to select the best one for a client device and the mobile network it is connected to.

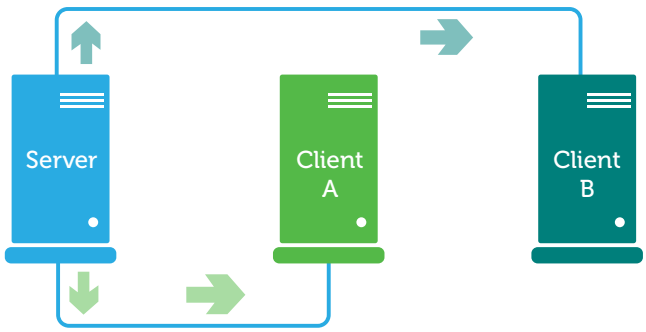
Diffusion by Push Technology is the most complete data distribution solution to provide real-time, bi-directional interactions between web and mobile applications across the Internet. Designed to deliver the right content to the right user at the right time, regardless of connectivity or location, it offers best-in-class performance in terms of capacity, being able to scale up to millions of simultaneous connections, and offering the lowest possible latencies for data delivery.

Able to deliver real-time user experiences owing to bandwidth efficiency, Diffusion only sends a snap-shot and then only the deltas of change, therefore only putting small amounts of data out over the Internet.

Diffusion reduces bandwidth and radically reduces the hardware needed to serve your mobile and Internet channels.

Appendix A Environment

The test environment consists of three machines, with two central processing unit (CPU) sockets per machine and six hyper-threaded cores per socket: 1 server and 2 clients. The 2 client machines are directly connected to the server. Only 1 client machine was employed during the 10 byte message runs and was not saturated.



Machine Configuration

Machine Configuration	
Manufacturer	Dell
Model	PowerEdge R720
Processors (x 2)	Intel Xeon ES-2630 2.00GHz, 15M Cache, 7.2GT/s QPI
Memory (x 4)	8GB RDIMM, 1333MHz, Low Volt, Dual Rank
Networking	Intel 10Ge cards
OS	Fedora Core 17 – with Linux kernel: 3.6.5-1.fc17.x86_64

Benchmark Environment

The server is deployed on one of the machines and constrained to a single processor.

All clients are deployed on the other two machines and constrained to 2 processors.

Server and client processor constraints are imposed on each machine to force saturated conditions in the throughput benchmark earlier in test runs.

The **Diffusion** server is developed in Java. The following Java runtime was used on both benchmark machines:

Java(TM) SE Runtime Environment (build 1.7.0_51-b13).

Java HotSpot(TM) 64-Bit Server VM (build 24.51-b13, mixed mode).

The benchmarks were run with the machines directly interconnected using Intel 10Ge network cards.

About Push Technology

We make the Internet work for our mobile-obsessed, everything-connected world. Leading brands like 888 Holdings, DAB Bank, IBM, and William Hill leverage our technology to power applications critical to revenue growth, customer engagement, and business operations. Learn how to deliver apps at scale and speed at **www.pushtechnology.com**

LONDON
+44(0) 2035 880900

SAN JOSE
+1 408 780 0720

www.pushtechnology.com