MINIO

Accelerating object storage with PORTrockIT

BRIDGEWORKS

EXECUTIVE SUMMARY

Many modern applications require a scalable way to store and retrieve files and other types of data that cannot be stored efficiently in a traditional database. For example, any application that allows users to upload and view images, videos or files of any type will probably store those assets in an object storage cluster.

Amazon S3, a cloud service, is the market leader for object storage – but many companies have information security or compliance requirements that prevent them from storing their data in the public cloud. MinIO provides an open source alternative: an enterprise-grade object storage platform that provides an S3-compatible API, but is not tied to any particular infrastructure. Users can set up their own clusters on-premises, or in a local data centre.

However, running your own MinIO cluster can pose challenges, especially when you lack the resources of a large company like Amazon. Transferring files across a wide area network (WAN) using a traditional TCP/IP connection can be extremely slow, especially when network latency and packet loss degrade performance.

If users have to wait minutes or hours for a file to upload or download, it reduces their productivity and creates a major barrier to adoption. And if a company is relying on MinIO to back up its file servers or provide access to large datasets for analytics, a slow network can cause long-running jobs to overrun their allotted time-window or even fail altogether.

This paper shows how PORTrockIT can transform the performance of MinIO at both storing and retrieving data across a WAN by counteracting the effects of latency and packet loss.

"In several cases, the transfer rate with PORTrockIT was ***more than 99 times faster*** *than a traditional network architecture."*

Our benchmarks show that PORTrockIT was able to accelerate data transfer significantly in every scenario we tested; in several cases, its transfer rate was more than 99 times faster than a traditional network architecture.

WHY SPEED MATTERS

In a typical MinIO configuration, users will interact with a front-end application running on their desktop or in a web browser. When they need to access the object store to upload or download a file, the application will send an HTTP request over the WAN to the MinIO cluster's API. To send a file to MinIO, you include it in the body of a PUT request, while to retrieve a file, you send a GET request with the file's unique ID, and MinIO will include the file in the body of its response.

For technical reasons (which we will explore in the next section), sending and receiving these data-heavy PUT requests and GET responses across a WAN can be a slow process. This can be a major issue for many reasons, depending on the business use case. You can see some examples in the table below.

Business use case	Examples	Impact of slow transfer rates
Internal business applications	Content management systems, document archives, medical imaging platforms	• Reduce user productivity • Reduce efficiency of business processes • Make it difficult to guarantee on-time delivery to customers • Hinder business decision-making
Data replication processes	Backing up file servers or virtual machine images	• Cause long-running tasks to overrun their allotted window, impacting other systems • Cause tasks to time-out and fail, leaving critical data unprotected
Analytics processes	Training machine learning or deep learning models on large data sets	• Increase the time taken to train a model • Reduce the number of iterations that can be performed • Impact the quality and accuracy of models • Restrict the flow of insight to the business

MinIO includes built-in data compression and parallelisation features that attempt to mitigate the risk of slow transfer rates. However, these capabilities have their limitations:

- Additional compression does not benefit modern video and image file formats, which already utilise extremely advanced compression algorithms.

- Parallel threads help MinIO take greater advantage of available bandwidth, but TCP/IP limitations still limit performance on a network where latency and packet loss are high.

Although MinIO's compression and parallelisation capabilities may mitigate the symptoms of an unreliable WAN to some extent, they do not address the root causes of slow transfer rates. To understand why, we need to take a deep dive into how WAN data transfers work.

THE PROBLEMS: LATENCY AND PACKET LOSS

The chief culprit for poor WAN replication performance is latency – the time delay between a source system sending a packet across the network, and the target system receiving that packet.

The main cause of latency is the physical distance that the packet has to travel. Even with high-speed fibre-optic cabling, latency can increase at a rate of up to 5 microseconds per kilometre travelled.

The need to receive, queue and process packets at either end of the connection and at any intermediate gateways also adds significantly to the round-trip time. In short, the further the data has to travel and the more gateways it has to pass through, the greater the latency.

For network traffic sent via the TCP/IP protocol (and almost all HTTP traffic falls into this category), high latency can cripple transfer rates.

TCP/IP works by sending a group of packets, then waiting for an acknowledgement that the packets have been received before it sends the next group. If the latency of the connection is high, then the sender spends most of its time waiting for acknowledgements, rather than actually sending data. During these periods, the network is effectively idle, with no new data being transferred.

When packet loss occurs, the situation gets even worse. If a packet is lost before it is received by the recipient, or the acknowledgement goes astray before it reaches the sender, TCP/IP automatically reduces the number of packets it sends in the next group to compensate for the unreliability of the connection. As a result, network utilisation falls even further, because the sender is sending fewer packets in the same amount of time.

Companies often try to solve TCP/IP performance issues by investing in more expensive network infrastructure that offers a larger maximum bandwidth. However, this does not fix the problem. As we have seen, latency and packet loss prevent TCP/IP connections from fully utilising the available bandwidth – so any investment in additional bandwidth will simply be wasted unless the latency and packet-loss issues can be addressed.

> *"**Investment in additional bandwidth will be wasted** unless latency and packet-loss issues can be addressed."*

THE SOLUTION: PORTROCKIT

PORTrockIT offers a solution to WAN transfer performance issues. Instead of sending a group of packets down a single physical connection and waiting for a response, the solution creates a number of parallel virtual connections that send a constant stream of data across the connection.

As soon as a virtual connection has sent its packets and starts waiting for an acknowledgement from the recipient, PORTrockIT immediately opens another virtual connection and sends the next set of packets. Further connections are opened until the first connection receives its acknowledgement; this first connection is then re-used to send another set of packets, and the whole process repeats.

This parallelisation practically eliminates the effects of latency by ensuring that the physical connection is constantly transferring new packets from the sender to the recipient: there is no longer any idle time, and the network's bandwidth can be fully utilised.

The solution also significantly reduces the impact of packet loss. If one of the PORTrockIT virtual connections loses a packet, TCP/IP will only reduce the number of packets in the next group sent by that specific connection. All the other connections continue to operate at full speed.

PORTrockIT is also capable of optimising the flow of data across the WAN in real time, even if network conditions change. The solution incorporates artificial intelligence engines that continuously manage, control and configure the connection settings – enabling the appliance to operate optimally at all times, without input from a network administrator.

PORTrockIT is installed as a pair of appliances (or virtual instances), deployed at either end of the WAN.

When a user's application sends a GET or PUT request to MinIO, it is routed to the PORTrockIT appliance on the near side of the WAN – for example, in the server room of the office where the user is working.

PORTrockIT then passes the request through a set of virtual connections to a second PORTrockIT appliance on the far side of the WAN, where the MinIO cluster lives. Once the second PORTrockIT appliance begins receiving packets, it routes them seamlessly to MinIO, and returns the response via the same route.

> *"PORTrockIT provides an* ***instant and dramatic boost to network transfer performance****, with no need to make any changes to the rest of the network."*

The result is an instant and dramatic boost to network transfer performance, simply by plugging the two appliances into the existing network. There is no need to make any changes to the rest of the network architecture, or to invest in additional servers, gateways, or network fibre.

TURNING THEORY INTO PRACTICE

To demonstrate the results that PORTrockIT can deliver for MinIO environments, Bridgeworks conducted a set of performance tests at an independent testing facility in the UK. The test infrastructures mimicked a real-world WAN architecture, using a WANulator to simulate different levels of latency and packet loss between the MinIO client application and the MinIO server.

The first set of tests were performed on an unaccelerated architecture, where the client and the server were both connected directly to the WANulator (see Figure 1).

TEST EQUIPMENT

SOFTWARE:

- MinIO server (RELEASE.2019-03-06T22-47-10Z)
- MinIO client (RELEASE.2019-03-09T00-30-31Z)

HARDWARE:

- 2 x Dell PowerEdge R710 servers
  - 2 x Intel Xeon E5645 processors
  - 25 GB RAM
  - Ubuntu 16.04.6
- 2 x PORTrockIT 200 series nodes
- 1 x WANulator

The same tests were then repeated on an architecture that added two PORTrockIT appliances on either side of the WANulator, and routed all traffic from the client and the server through these appliances.

Bridgeworks tested the performance of MinIO for both GET and PUT requests. All of the tests on the PORTrockIT accelerated architecture (see Figure 2) used a pre-compressed 5 GB file to assess the average transfer speed.

For the tests on the unaccelerated architecture, it was necessary to switch to smaller files at higher levels of latency and packet loss – otherwise the tests would have taken many hours to complete. In subsequent sections of this paper, these results have been scaled proportionally to indicate how long the architecture would have taken to transmit a 5 GB file. This makes it easier for readers to compare the performance of the two architectures.

Figure 1: Unaccelerated environment setup

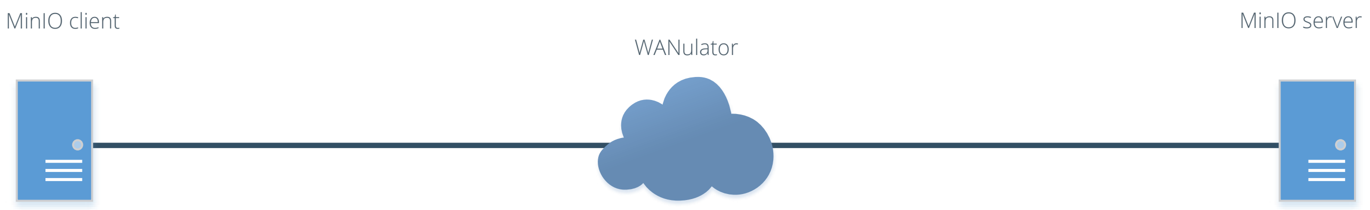
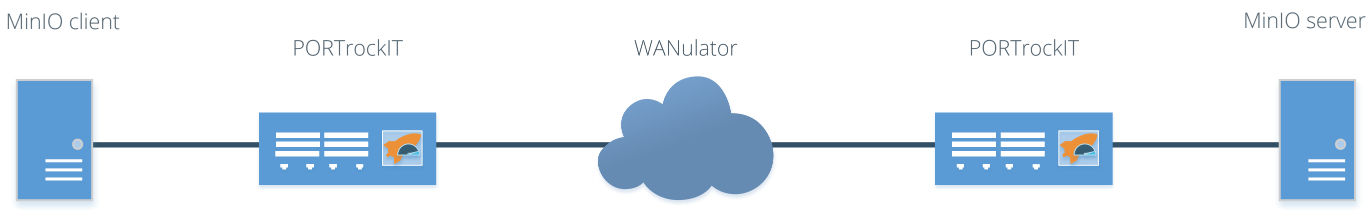


Figure 2: Accelerated environment setup



WHAT THE DATA TELLS US

LATENCY

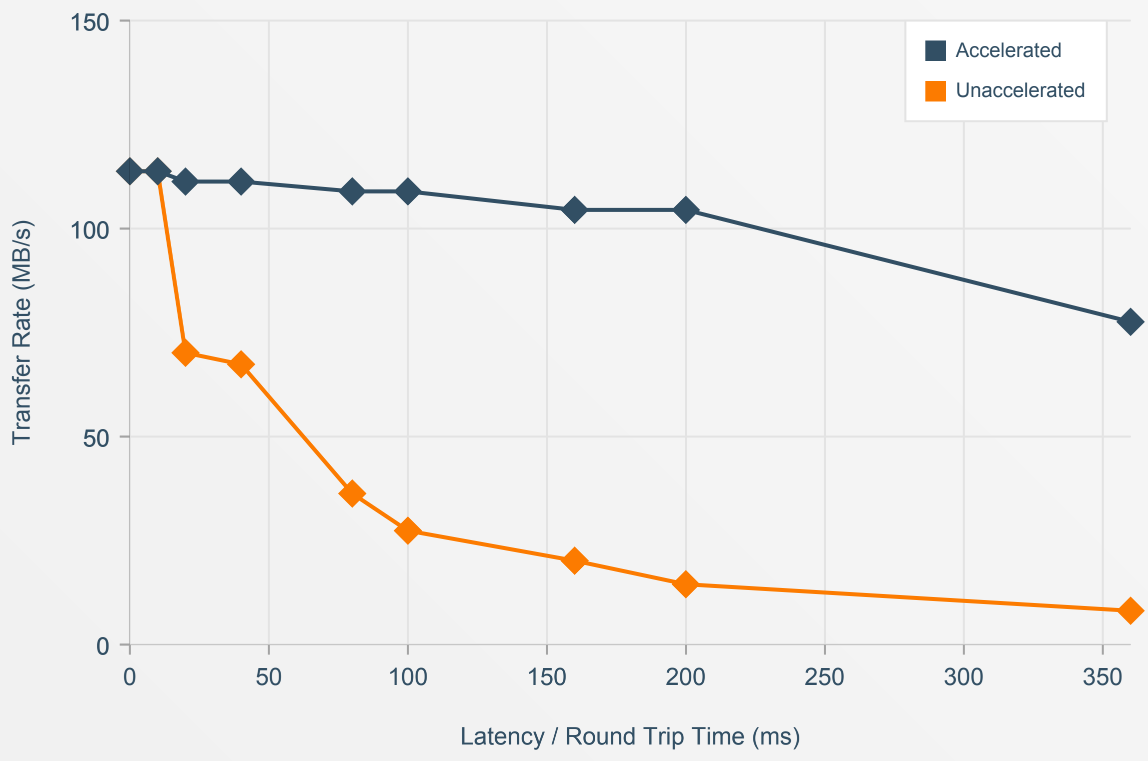
The first test simulated a scenario with no packet loss, at latencies ranging from 0 ms to 360 ms round trip time (RTT). A 5 GB test file was transferred across the WAN via both GET and PUT requests, first via the unaccelerated architecture, and then again via the accelerated architecture with PORTrockIT.

Looking at Figures 3 and 4, the results show that performance on the unaccelerated architecture degraded as latency increased.

GET requests were affected significantly more than PUT requests: in the scenario with 360 ms of latency, the transfer speed for a GET request was just 8.13 MB/s, while the PUT request fared better at 29.60 MB/s.

The accelerated architecture performed similarly at low levels of latency, but showed almost no degradation in performance as latency increased. Even at 360 ms of latency, the transfer rate for 82.58 MB/s for PUTs (64% faster than the unaccelerated architecture), and 77.58MB/s for GETs (89% faster than the unaccelerated architecture).

Figure 3: Chart of latency at 0% packet loss for MinIO GETs



PACKET LOSS

Next, the team decided to investigate the impact of introducing packet loss at different levels of latency. Two scenarios were considered: a network with 0.1% packet loss, and a network with 0.5% packet loss.

Figures 5, 6, 7 and 8 all show that the unaccelerated architecture saw severe performance degradation for both GET and PUT requests from the combination of latency and packet loss. In both scenarios, the accelerated architecture performed considerably better on both GETs and PUTs.

Even in the most challenging scenario (360 ms of latency with 0.5% packet loss) the accelerated architecture achieved a transfer rate of 38.21 MB/s for GET requests – more than 99 times faster than the unaccelerated transfer rate of 0.37 MB/s. In the same scenario, PUT requests ran at 27.98 MB/s on the accelerated architecture, compared to just 1.23 MB/s on the unaccelerated architecture – more than 95 times faster.

Figure 4: Chart of latency at 0% packet loss for MinIO PUTs

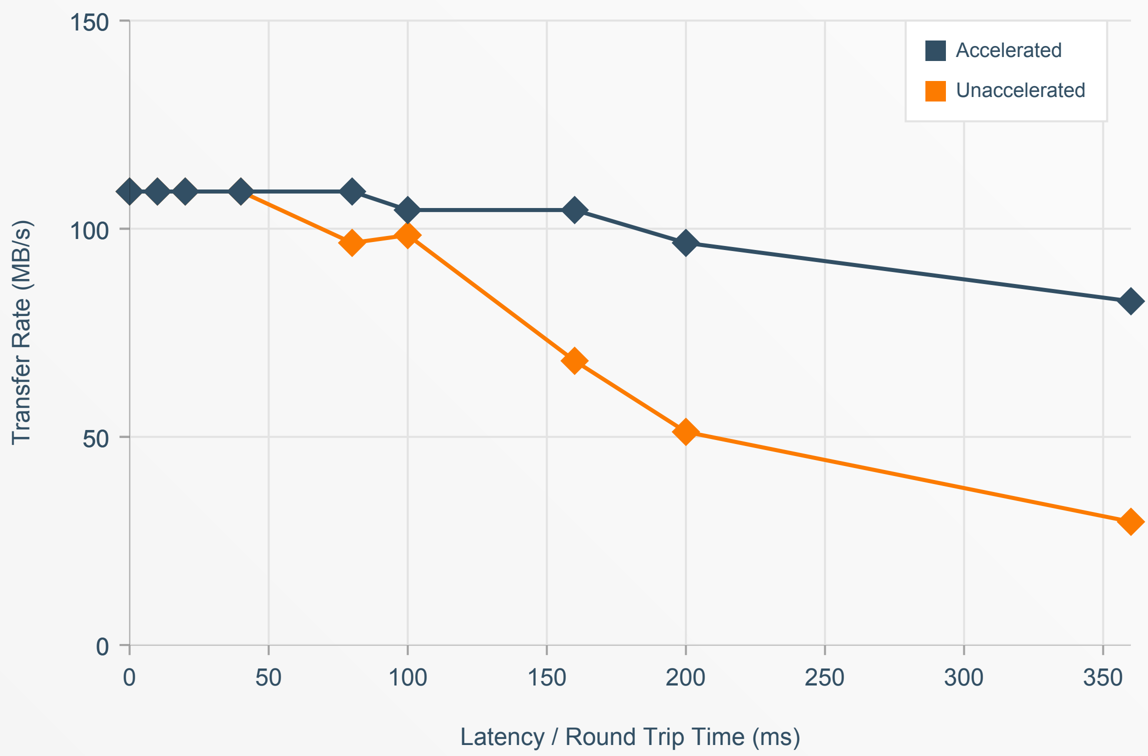


Figure 5: Chart of latency at 0.1% packet loss for MinIO GETs

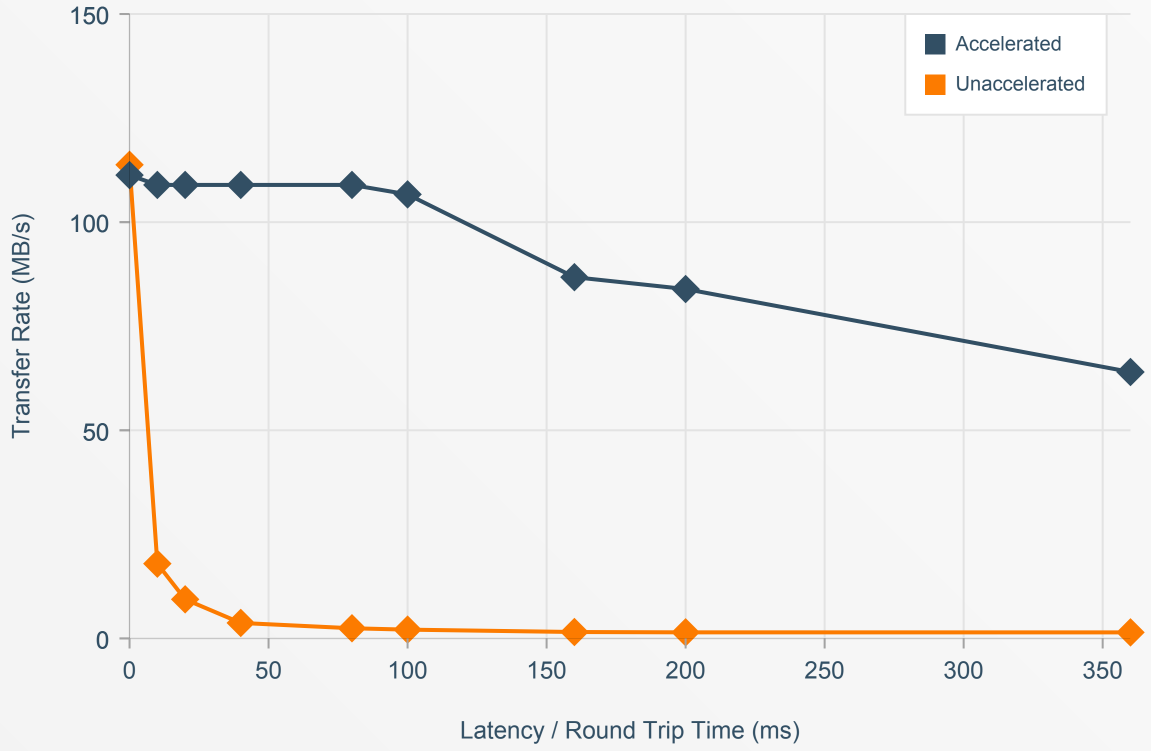


Figure 6: Chart of latency at 0.1% packet loss for MinIO PUTs

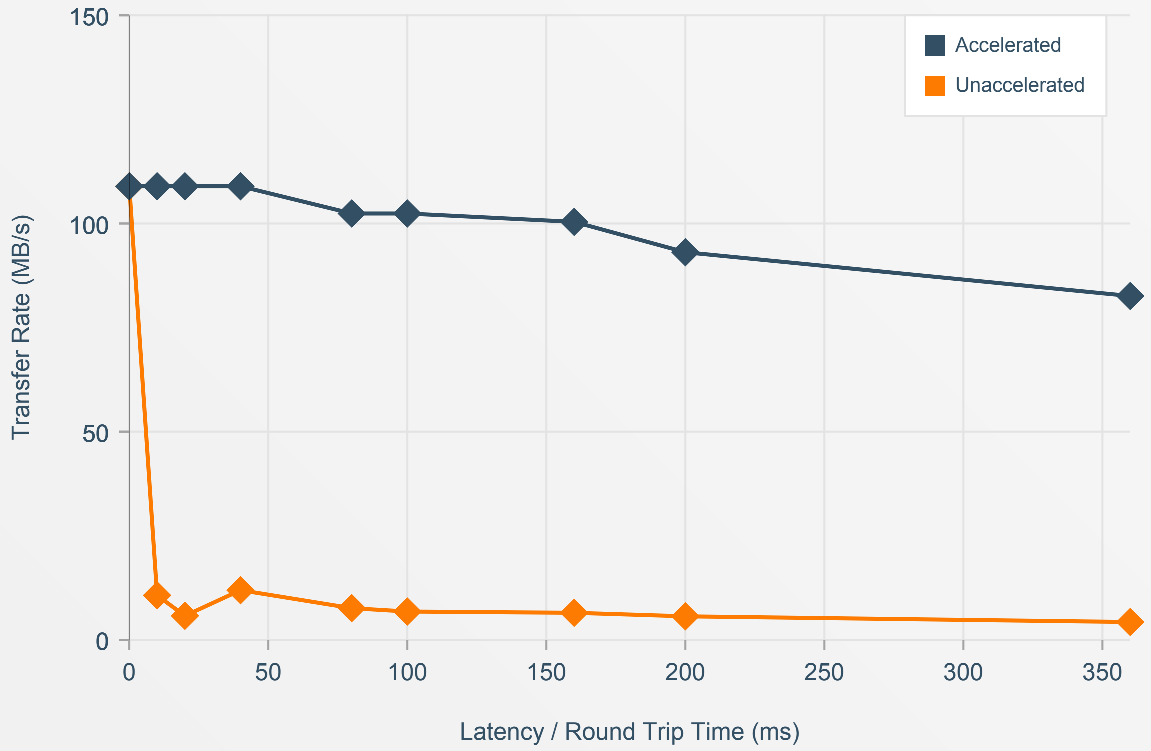


Figure 7: Chart of latency at 0.5% packet loss for MinIO GETs

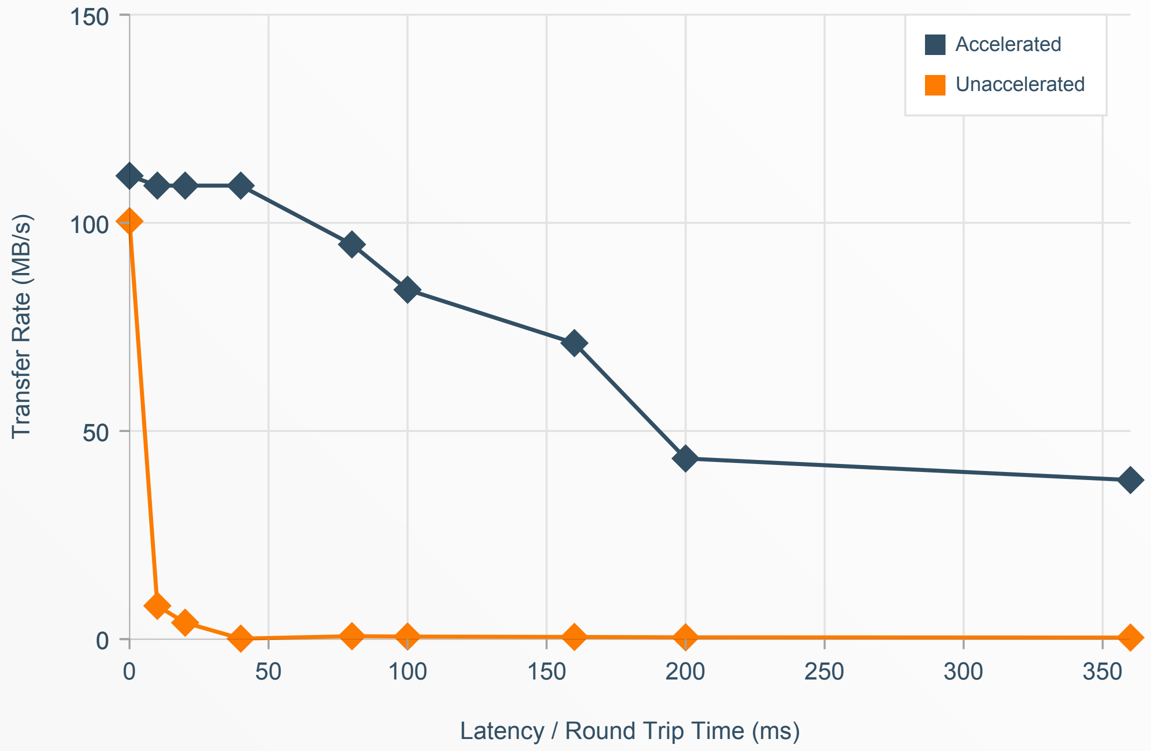
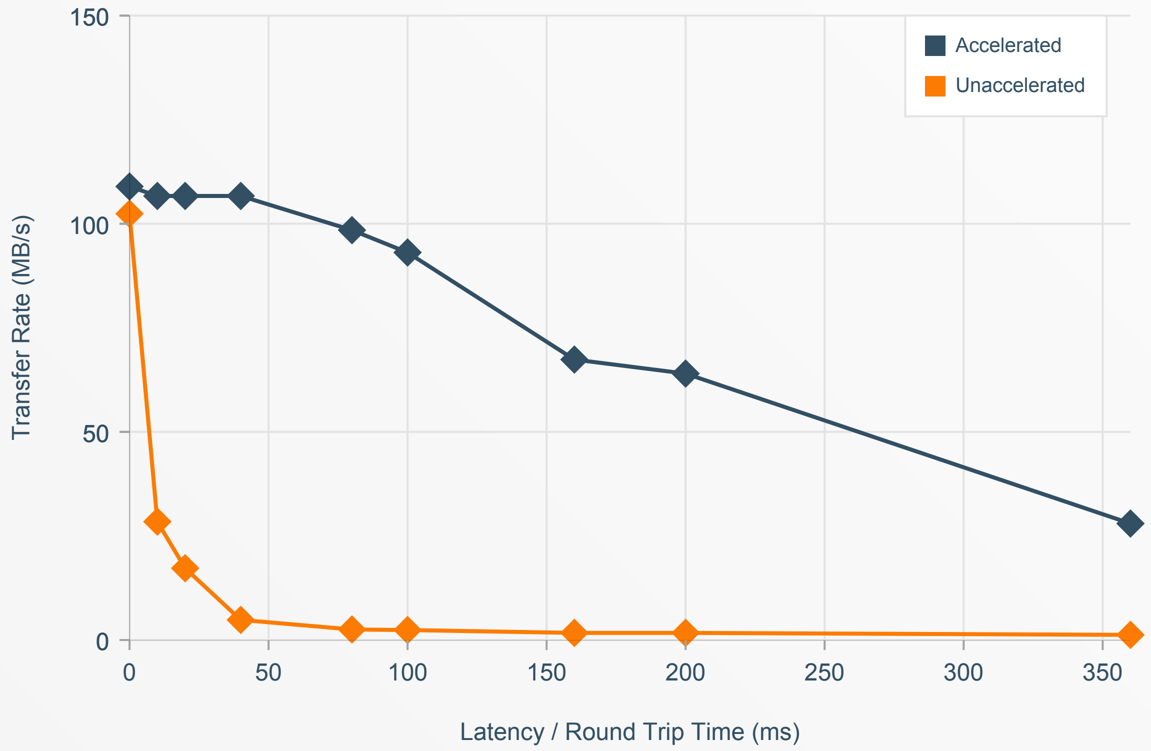


Figure 8: Chart of latency at 0.5% packet loss for MinIO PUTs



REALISING THE BUSINESS BENEFITS

MinIO is potentially an ideal solution for companies that need to host and control their own object store.

However, in the common case where a company has one or more office locations that need to connect to a MinIO cluster at a remote data centre, it's vital to ensure that network latency and packet loss will not degrade performance.

By adding PORTrockIT to their MinIO network architecture, companies can resolve these issues instantly.

PORTrockIT offers plug-in-and-go technology that can be implemented quickly with minimal impact on the rest of the IT infrastructure – keeping deployment costs and risk to a minimum.

Furthermore, by maximising the performance of existing infrastructure, PORTrockIT reduces the need to invest in expensive high-bandwidth connections or more powerful servers – enabling significant cost-avoidance.

Depending on the company's use case for MinIO, the introduction of PORTrockIT can also deliver significant higher-level business benefits, as shown in the table below.

Business use case	Results of PORTrockIT acceleration
Internal business applications	• Increase responsiveness, boosting user productivity • Accelerate business processes, increasing efficiency • Give decision-makers instant access to the data they need
Data replication processes	• Ensure that tasks run reliably and finish on time • Eliminate the risk of impacting other business systems • Keep business data fully protected at all times
Analytics processes	• Accelerate training cycles for machine learning models • Enable faster iterations, creating higher-quality models • Deliver accurate insight to the business more quickly

ABOUT THE AUTHOR

George Trossell is a Systems Engineer at Bridgeworks, specialising in technical presales and solution architecture.

Over the course of his career, George has gained expertise across a broad spectrum of engineering and commercial disciplines – from developing and testing applications and tools to providing sales and technical support. In particular, George is passionate about monitoring and measuring the performance of systems and software, making him the ideal person to lead Bridgeworks' testing and benchmarking initiatives.

George has a Master's degree in electrical, electronic and communications engineering from the University of Kent, and is a Member of the Institution of Engineering and Technology.

TAKE THE NEXT STEPS

To learn more about PORTrockIT and other smart networking solutions from Bridgeworks, please visit www.4bridgeworks.com, or call us on +44 (0) 1590 615 444.

BRIDGEWORKS