COMMISSIONED BY

Redpanda

MCKNIGHT CONSULTING GROUP

DATA STREAMING PERFORMANCE

PRODUCT EVALUATION:

Redpanda Cloud Dedicated, Redpanda BYOC, Confluent Cloud

Performance and Total-Cost-of-Ownership

RELATED RESEARCH

Data Stream Processing Ease of Use and TCO

Benchmark of Data Pipeline and Performance Cost

William McKnight

Colin Sullivan

Prepared by

MCKNIGHT CONSULTING GROUP

McKnight Consulting Group
www.mcknightcg.com

January 2025

Contents

Executive Summary

Over the last 20 years, the data streaming and stream processing industry has emerged as a key component of modern data architectures. It empowers organizations to capture, process, and analyze data in real-time, unlocking new opportunities for data-driven insights and decision-making.

Apache Kafka stands at the forefront of the data streaming and stream processing industry with time-proven high scalability, fault tolerance, and data persistence features. This opens source distributed streaming platform, initially developed by LinkedIn in 2011, has been one of the streaming technologies revolutionizing how organizations handle real-time data and batch data processing.

Kafka's capabilities have made it a de facto standard for building real-time data pipelines, stream processing applications, and event-driven architectures.

Confluent, founded by the creators of Kafka, was the first company to commercialize software to help enable production worthy operations of Kafka in 2016. Since that time, Confluent has added high performance cloud-based services for Kafka, with their best performing managed service, the Confluent Cloud Dedicated Cluster.

Redpanda followed suit in 2019 providing a highly-optimized Kafka server offering similar services. Redpanda’s best performing service today is a new BYOC offering.

The focus of this report is on benchmarking performance between Redpanda and Confluent, considering the total cost of ownership (TCO) with large scale benchmarks. The OpenMessaging Benchmark framework was used as the basis for testing three types of workloads.

Redpanda is 54% less expensive (46% of cost) than Confluent's quoted pricing for this benchmark for a Medium Fan-in Use Case and is 60% less expensive (40% of cost) than Confluent for a large fan-in use case, with nearly equal performance and support. Notably, this comparison is with dynamic instance pricing. In practice, these differences in cost would be magnified when leveraging cloud provider savings plans or discounts with reserved instances.

An organization might run at least two Redpanda clusters for each dedicated Confluent cluster, processing at much more streaming data for the same cost with roughly equivalent levels of support.

Platform Summary

This report compares the Redpanda BYOC and Confluent Cloud streaming platforms. While there are many similarities, there are some distinct differences in the platforms.

Bring Your Own Cloud (BYOC)

Generally, Bring Your Own Cloud (BYOC) is a cloud computing model in which organizations use their own cloud infrastructure to host and manage their applications provided by third party vendors. This contrasts with the traditional SaaS model, in which vendors offer services outside of the cloud infrastructure owned by a company.

Industry wide, there are several benefits to BYOC, including:

Cost-effectiveness: BYOC can be more cost-effective than SaaS offerings as organizations do not have to pay for infrastructure and services that they do not use. Software is launched inside their security perimeter eliminating the need for VPCs and can choose regions that may be more cost effective than others.

Security: BYOC can be more secure than traditional cloud computing, as organizations have more control over their own external security perimeter and can place deployments in VPCs.

Data Sovereignty: When using a BYOC deployment, companies can be assured that their data will remain secure and in a specific location, allowing them to adhere to regulations requiring data to remain in a specific location, country, or region.

However, there are drawbacks associated with many BYOC offerings, although Redpanda has mitigated many of them. See below for a more specific description of Redpanda's BYOC offering. These drawbacks may include:

Complexity: BYOC can be more complex to manage than SaaS offerings, as organizations are responsible for monitoring and upgrading the software deployment. They also permit access to components where human error can cause critical failures.

Vendor Support: In general, it may be more difficult for vendors to support BYOC deployments, as they do not have the same introspection available to diagnose or debug issues.

Overall, BYOC can be a cost-effective, flexible, and secure option for organizations that have the resources and expertise to manage their own cloud infrastructure. However, organizations should carefully consider the challenges associated with BYOC before deciding.

Redpanda has addressed the drawbacks found in many BYOC deployments with management and support tooling enabling it to provide a fully managed service.

Redpanda Fully Managed BYOC

Redpanda BYOC is a deployment option for Redpanda which allows organizations to run Redpanda on their own cloud infrastructure, giving them more control over their data and security. Redpanda's offering provides the benefits of BYOC while avoiding many of the drawbacks.

Redpanda states that their BYOC option is good for organizations that have the resources and expertise to manage their own cloud infrastructure and is also a good option for organizations that have sensitive data or that need to comply with specific regulations. At the time of this report, Redpanda's BYOC offering is supported on AWS, Azure, and Google Cloud providers.

Redpanda mitigates the drawbacks of BYOC through a **fully managed offering:** Clusters are automatically provisioned within a customer's cloud provider account and the deployment is supported by Redpanda engineers to meet a strict availability SLA and to deploy upgrades and hotfixes. A dedicated customer success contact and opportunities for services and training are also included in the offering.

Redpanda Dedicated Clusters

It is worth noting that Redpanda offers a dedicated cluster. Redpanda's dedicated clusters are also for production workloads that require high throughput, performance, reliability, or data isolation. Dedicated clusters are for users that do not want to run clusters in their own VPCs or directly deal with cloud vendors. Like BYOC, they can be selected to run on AWS, Azure, and Google Cloud providers. A dedicated cluster is single-tenant and fully managed in an isolated environment offering higher performance and higher reliability than the serverless offering but does not scale up to the level that BYOC clusters can. Given that we are comparing the most performant offerings of each company we will use Redpanda's BYOC in this comparison.

Confluent Cloud

Confluent offers tiers including Basic, Standard, Enterprise, and Dedicated options. Dedicated is equivalent to Redpanda's Dedicated and BYOC clusters, and can be scaled up by provisioning CKUs, or a unit of horizontal scaling that provides pre-allocated resources. The user configurable limit is 24 CKUs, although up to152 CKUs are available by request.

During the process of benchmarking, Confluent announced a BYOC offering with the acquisition of Warpstream. Guidance by Confluent's Global Field CTO in the article "*Deployment Options for Apache Kafka: Self-Managed, Fully-Managed / Serverless and BYOC (Bring Your Own Cloud)"*[1] suggests BYOC should not be the first choice.

Confluent cloud offers ala carte support services at a monthly subscription, along with guidance and education.

Similarities and Differences

The platforms are quite similar and have a similar flow in onboarding and creating clusters.
Redpanda's BYOC is running in the end user's cloud accounts so some minimal bootstrapping is required (via command line tools), to stage things so Redpanda can take over the bootstrapping. Confluent's Dedicated does not require command line tools at all, as expected with a standard SaaS offering. Both have similar monitoring metrics, tooling to manage clusters, create users, roles, and access control lists. Both platforms offered deployments in the three major cloud providers and options for private VPNs or private links. The skill set required to provision and manage clusters is the same.

While both platforms offer ingress, egress, partitions, and connections as limits, Confluent provides additional limits (e.g. request/second) which was helpful in sizing clusters.

Authentication and Authorization are similar, although Redpanda requires a more secure handshake algorithm (SCRAM) with traditional username/password where Confluent uses PLAIN, a less secure mechanism with an API key/private key.

There are a number of similar features that we don't address in this report as we are focusing on performance of a single cluster.

In this report we compare the two highest performing offerings from each; Redpanda's BYOC and Confluent's Dedicated Cluster.

[1] https://www.kai-waehner.de/blog/2024/09/12/deployment-options-for-apache-kafka-self-managed-fully-managed-serverless-and-byoc-bring-your-own-cloud/

Test Setup

The setup for this Field Test was informed by the Open Messaging Framework[2] spec validation queries. This is not an official OMB benchmark. The queries were executed using the setup, environment, standards, and configurations described below.

Benchmark Data

The data sets used in the benchmark were a workload derived from the well-recognized industry standard Open Messaging Framework.

The Open Messaging Benchmark (OMB) is an open-source benchmarking framework designed to evaluate the performance of messaging and streaming platforms. It provides a standardized way to measure the throughput, latency, and scalability of these systems. OMB includes a set of pre-defined workloads that simulate real-world use cases, such as high-throughput messaging, low-latency messaging, and IoT data ingestion.

Relevance to Data Streaming Platforms
Data streaming platforms require high-throughput and low-latency messaging to handle large volumes of data in real-time. OMB helps evaluate the performance of these platforms in these critical areas. Additionally, OMB's scalability tests help determine a platform's ability to handle growing workloads. By providing a standardized benchmarking framework, OMB enables direct comparisons between different data streaming platforms, helping users choose the best solution for their specific use case.

Benefits for Data Streaming Platform Users
OMB provides a reliable and objective way to evaluate the performance of data streaming platforms, enabling users to make informed decisions about which platform to use. By identifying performance bottlenecks and areas for improvement, OMB helps users optimize their data streaming platforms for better performance and scalability. As an open-source framework, OMB is a cost-effective solution for evaluating data streaming platforms. Overall, the Open Messaging Benchmark is a valuable tool for evaluating the performance of messaging and streaming platforms, including data streaming platforms.

Changes to the Open Messaging Benchmark Code Base

The Open Messaging Benchmark had some limitations that required some changes which are detailed in the Appendix.

[2] More can be learned about the Open Messaging Framework benchmark at https://openmessaging.cloud/docs/benchmarks/.

Field Tests

The goal of the field tests performed was to find equivalent cluster sizes in both Redpanda and Confluent that could handle specific workloads and usage patterns.

To that end, tests were defined that simulate both medium and large-scale deployments with two patterns, fan-in which is common in observability and telemetry use cases, and fan-out, used in high-speed stream processing such as fraud and anomaly detection, real-time market data analysis requiring extremely low latencies.

As every use case is different, edge cases of these patterns were used to test Redpanda and Confluent, with the understanding that many use cases will fall somewhere in-between these usage patterns.

Through both documented limits and much trial and error, we found equivalent cluster sizes to run tests and approximate costs.

Test Procedure

We performed each test with a fresh environment by starting a Redpanda or Confluent cluster, setting up users and permissions, and then deploying benchmark worker nodes. Each test consisted of running a 1 minute warmup, and then a 20 minute test to reasonably ensure each test could maintain throughput without seeing an unrecoverable increase in the backlog of data.

Benchmark Workers

Benchmark workers are the instances that simulate clients with producers and consumers and are used to provide workloads that evaluate server cluster performance. We overprovisioned the benchmark nodes to ensure that they would not be a bottleneck in terms of CPU and network bandwidth. After each test utilization metrics were checked to ensure the test did not exceed resource capacity at any client.

- 8 AWS Instances
  - Size: m7i.16xlarge
  - Network Bandwidth: 18.75 Gbps
  - Memory: 256GB
  - Region: us-east-2
  - Availability Zone: us-east-2a
- Open Messaging Benchmark
  - Git commit: fe3c5a0c4a35997ccc94c1b0e0fc23797ed516db
  - Modifications as described in “Addendum: Changes to the Open Messaging Benchmark Code Base”
  - Slight modifications to use OpenJDK 1.8.
- Kafka Java Client Version: 3.7.1

Redpanda Tiers and Confluent CKUs

Along with documentation, a non-trivial effort in trial and error determined equivalent cluster sizing in terms of deployment. Redpanda and Confluent have various levers in choosing a cluster size, with limits on ingress, egress, logical partitions, and number of connections. Confluent has a few additional limits to consider when sizing clusters, namely connection attempts per second and requests per second. These differences lead to a mismatch in certain limits but, empirically, yielded roughly equivalent performance in benchmark throughput.

Common Configuration

There are several configuration parameters common throughout all tests. These include:

- Replication factor of 3
- Message size of 1024 bytes (1K)
- Consumer backlog size of 0
- 1 minute warm up phase
- 20 minutes of test time

Each test used the optimal # of partitions using the formula of:
Partitions = # Topics * max(producers per topic, consumers per topic).

Fan In (Telemetry/Observability)

Being an observability/telemetry use case the producer was configured to limit linger delay and only required one acknowledgement simulating the need for a lower level of durability in favor of performance and immediately flushing messages from the clients. The replication factor was 3. One high throughput test and one medium throughput test was performed.

Benchmark Driver

Table 1. Fan-In Benchmark Driver Configuration

Producer Configuration	Consumer Configuration
acks=1 linger.ms=1 batch.size=131072	auto.offset.reset=earliest enable.auto.commit=false

High Throughput Fan-In Benchmark

Table 2. High Throughput Fan-In Workload Configuration

Benchmark Parameter	Value
Topics	20
Producers Per Topic	100
Total Producer Count	2000
Producer Rate (Aggregate)	1320000
Consumer Groups Per Topic	1
Consumers Per Group	40
Total Consumer Count	800
Partitions Per Topic	100
Total Partitions	2000
Cluster Ingress (MB/s)	1351.68
Cluster Egress (MB/s)	1351.68

High Throughput Fan-In Cluster Sizes

Ingress and Egress drove the cluster size selection.

Table 3. High Throughput Fan-In Cluster Sizes

Limit	Redpanda Tier 8	Confluent CKU = 24
Ingress	1600 MB/s	1440 MB/s
Egress	3200 MB/s	4320 MB/s
Partitions	90,000	100,000
Connections	360,000	432,000
Connection Attempts	Unlimited	12,000/s
Requests	Unlimited	360,000/s

Medium Throughput Fan-In Benchmark

Table 4. Medium Throughput Fan-In Workload Configuration

Benchmark Parameter	Value
Topics	20
Producers Per Topic	100
Total Producer Count	2000
Producer Rate (Aggregate)	600000
Consumer Groups Per Topic	1
Consumers Per Group	10
Total Consumer Count	200
Partitions Per Topic	100
Total Partitions	2000
Cluster Ingress (MB/s)	614.4
Cluster Egress (MB/s)	614.4

Medium Throughput Fan-In Cluster Sizes

Ingress and Egress drove the cluster size selection.

Table 5. Medium Throughput Fan-In Cluster Sizes

Limit	Redpanda Tier 6	Confluent CKU = 11
Ingress	800 MB/s	660 MB/s
Egress	1600 MB/s	1980 MB/s
Partitions	45,000	49,500
Connections	180,000	198,000
Connection Attempts	Undocumented	5,500/s
Requests	Undocumented	165,000/s

Fan Out - Services

A services test testing fanout for simulating multiple aspects of high speed parallel processing of data with a larger number of topics was performed. A replication factor of 3 was used and the topics were configured with min.insync.replicas=2.

Benchmark Driver

The benchmark driver had to limit the poll records and poll interval to provide steady throughput and stabilize consumer behavior.

Table 6. Fan-out Benchmark Driver Configuration

Producer Configuration	Consumer Configuration
acks=all linger.ms=1 batch.size=1048576	auto.offset.reset=earliest enable.auto.commit=false max.partition.fetch.bytes=10485760 max.poll.records = 50 max.poll.interval.ms = 300000

Fan-Out Benchmark

Table 7. Fan-Out Workload Configuration

Benchmark Parameter	Value
Topics	300
Producers Per Topic	3
Total Producer Count	900
Producer Rate (Aggregate)	90000
Consumer Groups Per Topic	7
Consumers Per Group	10
Total Consumer Count	21000
Partitions Per Topic	70
Total Partitions	21000
Cluster Ingress (MB)	92.16
Cluster Egress (MB)	645.12

Fan-Out Cluster Sizes

Requests per second and the number of consumer groups played an impact in the sizing choice with the Confluent Dedicated Cluster. Clusters under 20 CKUs did not reliably pass the benchmarks or provided poor latencies.

Table 8. High Throughput Fan-out Cluster Sizes

Limit	Redpanda Tier 6	Confluent CKU = 20
Ingress	800 MB/s	1200 MB/s
Egress	1600 MB/s	3600 MB/s
Partitions	45,000	90.000
Connections	180,000	360.000
Connection Attempts	Undocumented	10,000/s
Requests	Undocumented	300,000/s

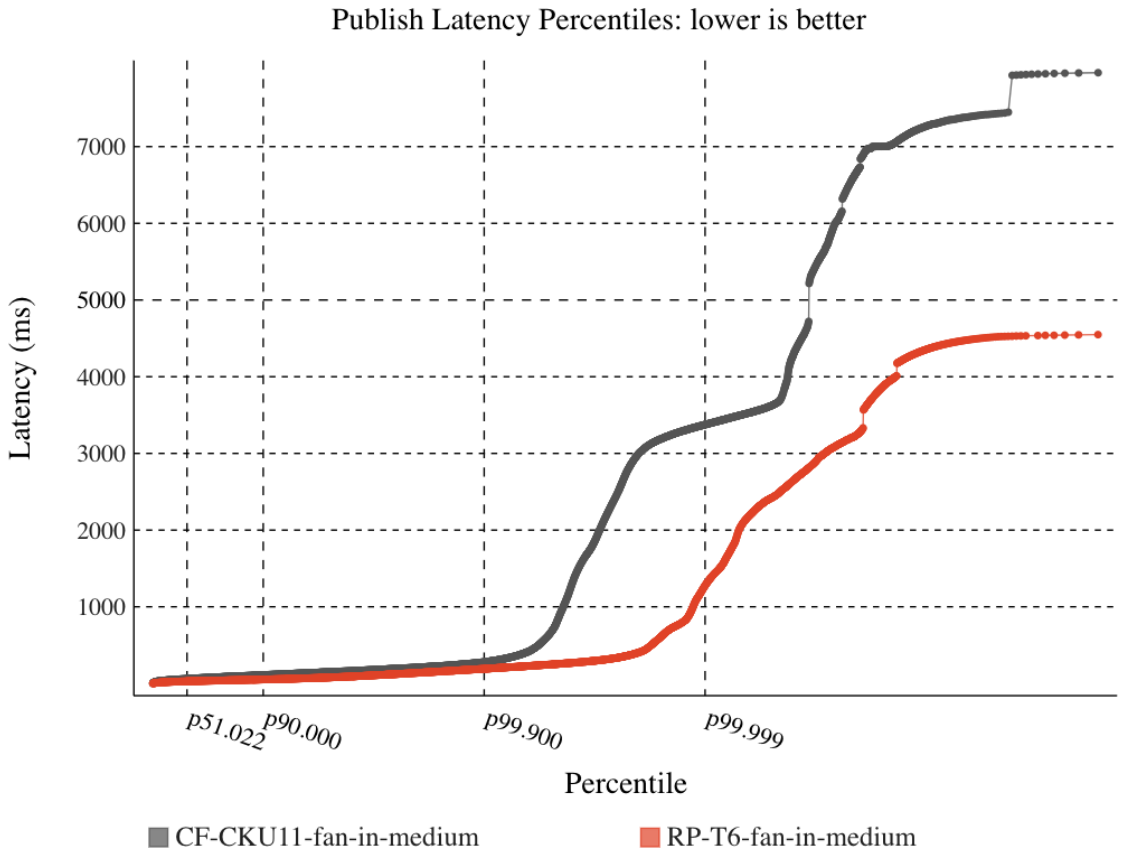
Benchmark Results

Benchmarks were created that take a few common communication patterns and push a steady flow of data against equivalently sized clusters from Redpanda and Confluent. In each benchmark we measure publish latencies, end to end latencies, publish rate, and consumption rate. Publish rate and consumption rate should be roughly equal (otherwise, the test would fail over time indicating we chose a cluster too small). We found notable differences in latencies between the two platforms. Each chart legend uses prefixes where RP/CF represents Redpanda or Confluent and RP-T and CKU represents the tier selection or # of CKUs provisioned.

Medium Fan In (Telemetry/Observability)

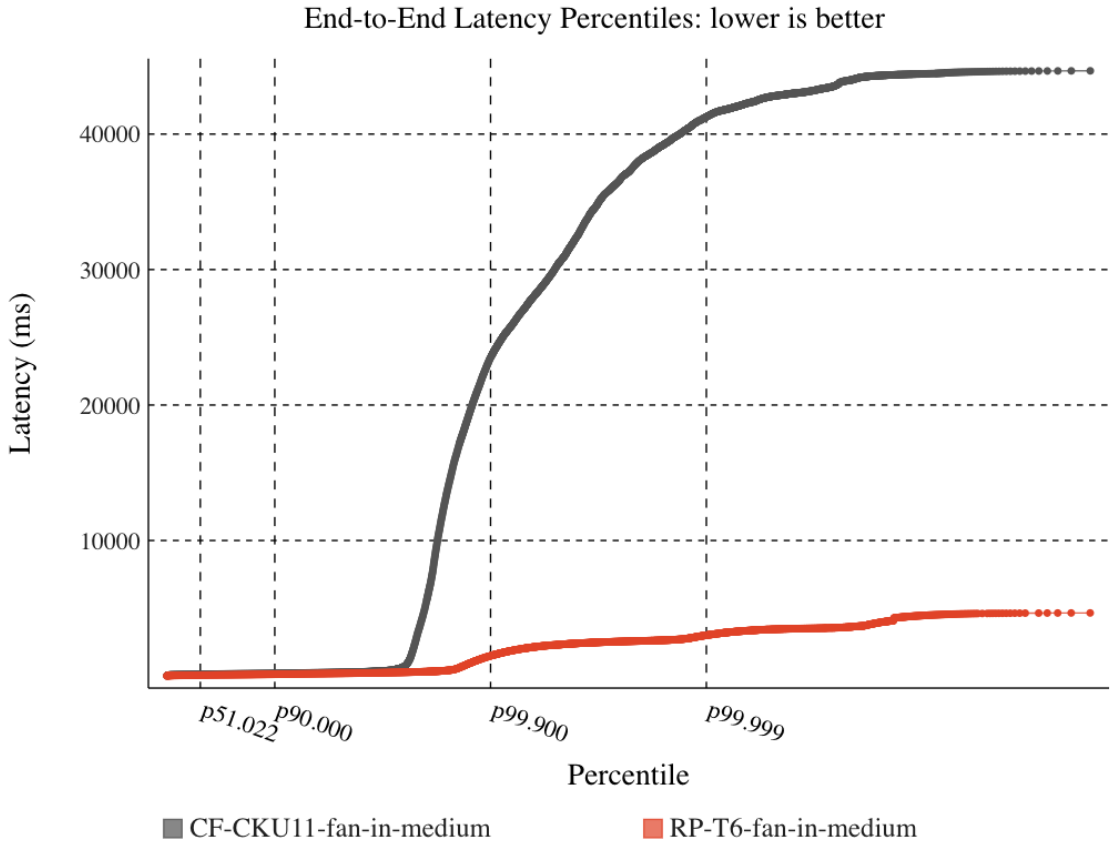
For equivalent cluster sizes that could continuously handle the same throughput, Redpanda had much more consistent performance in terms of latency. Confluent dedicated was consistent throughout much of the test but variations in consume rate resulted in temporary consumer backlogs. The benchmark nodes displayed consistent resource usage with CPU around 25% and no network spikes. Confluent metrics reported a cluster load between 80-85%.

Figure 1. Medium Throughput Fan-In Publish Tail Latency



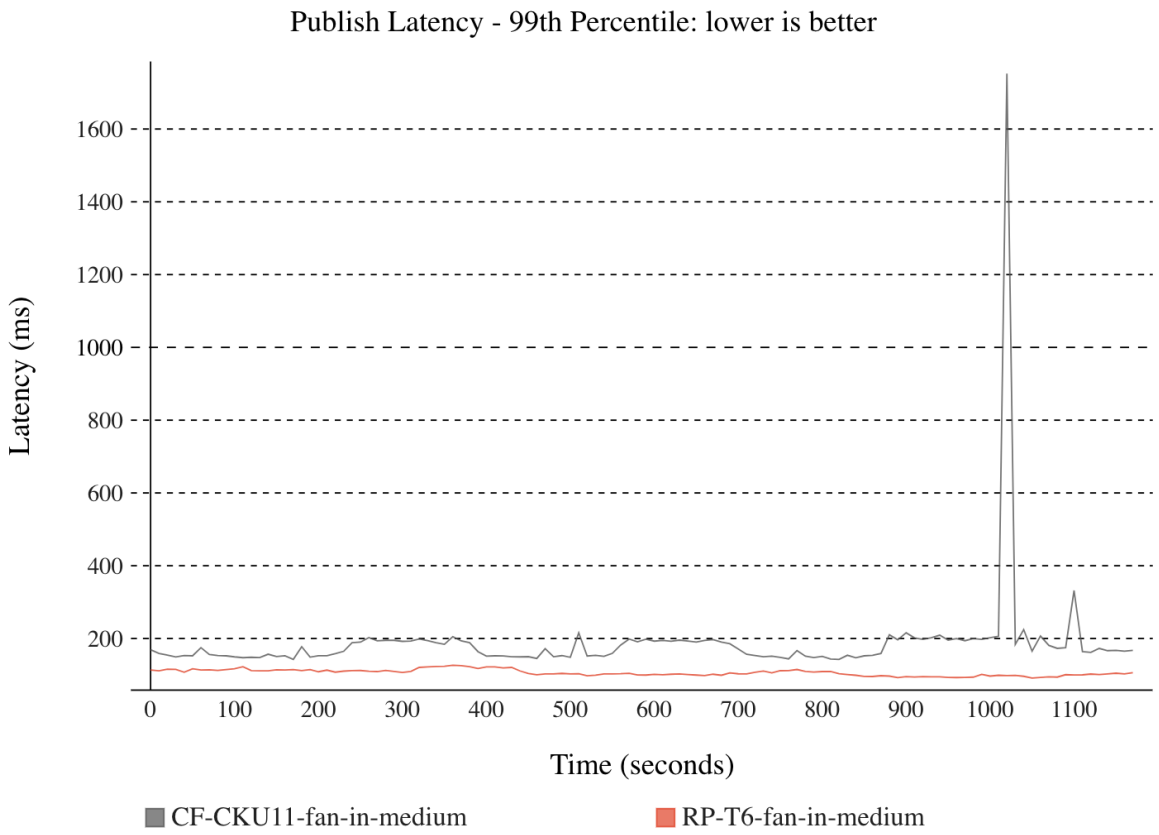
This graph describes tail latencies at P51, P90, P99.0, and P99.999, indicating how close the longer publish times experience in comparison to the majority of publish API calls. Server load will have a significant influence on these metrics. Confluent and Redpanda are equivalent until you pass the 99.9th percentile of longer latency measurements. Redpanda's cluster provided much more consistent publish latencies.

Figure 2. Medium Throughput Fan-In End-to-End Tail Latency



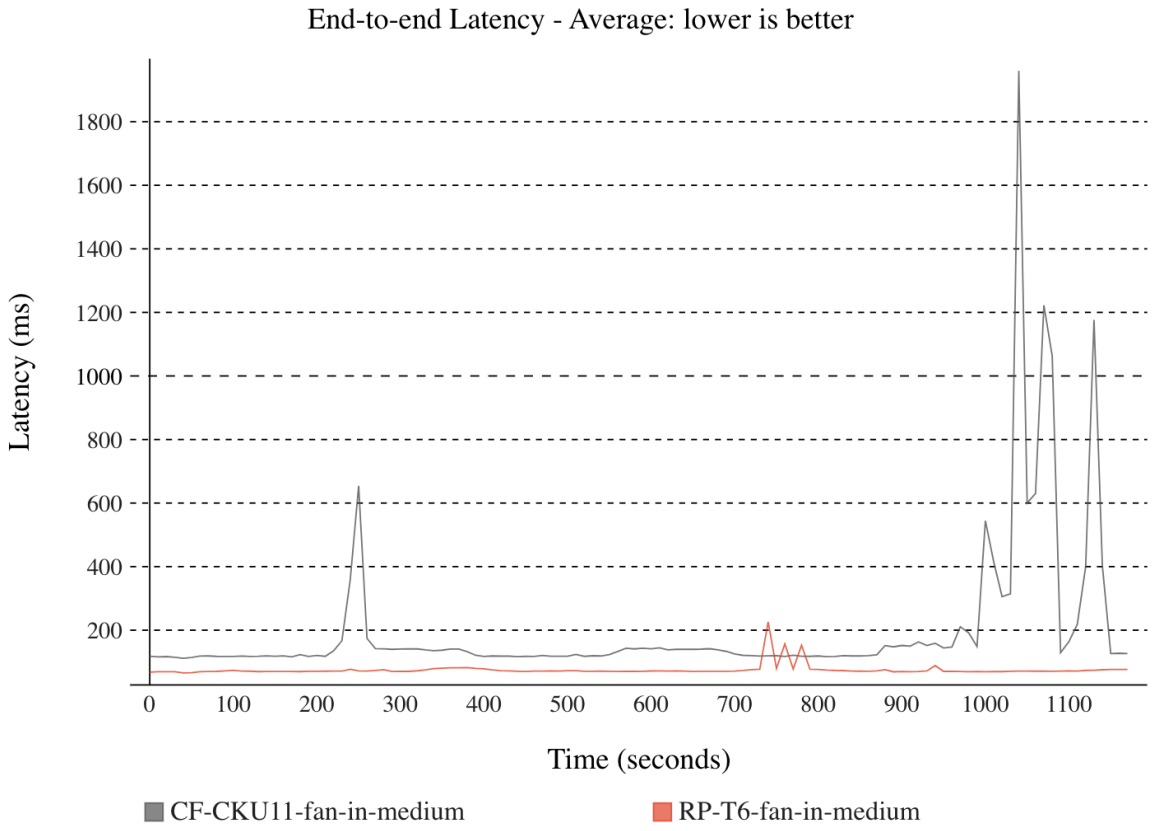
This graph displays tail latencies from the time it took from a message to get from the publisher to the consumer at P51, P90, P99.0, and P99.999, showing the difference in the maximum outliers. Server load and periodic backlogs at the consumer will be reflected in these results here. Both tests succeeded and Confluent and Redpanda's publish latencies are roughly equivalent until you approach the 99.9th percentile. Redpanda's cluster provided much more consistent end to end tail latencies.

Figure 3. Medium Throughput Fan-In Publish Latency



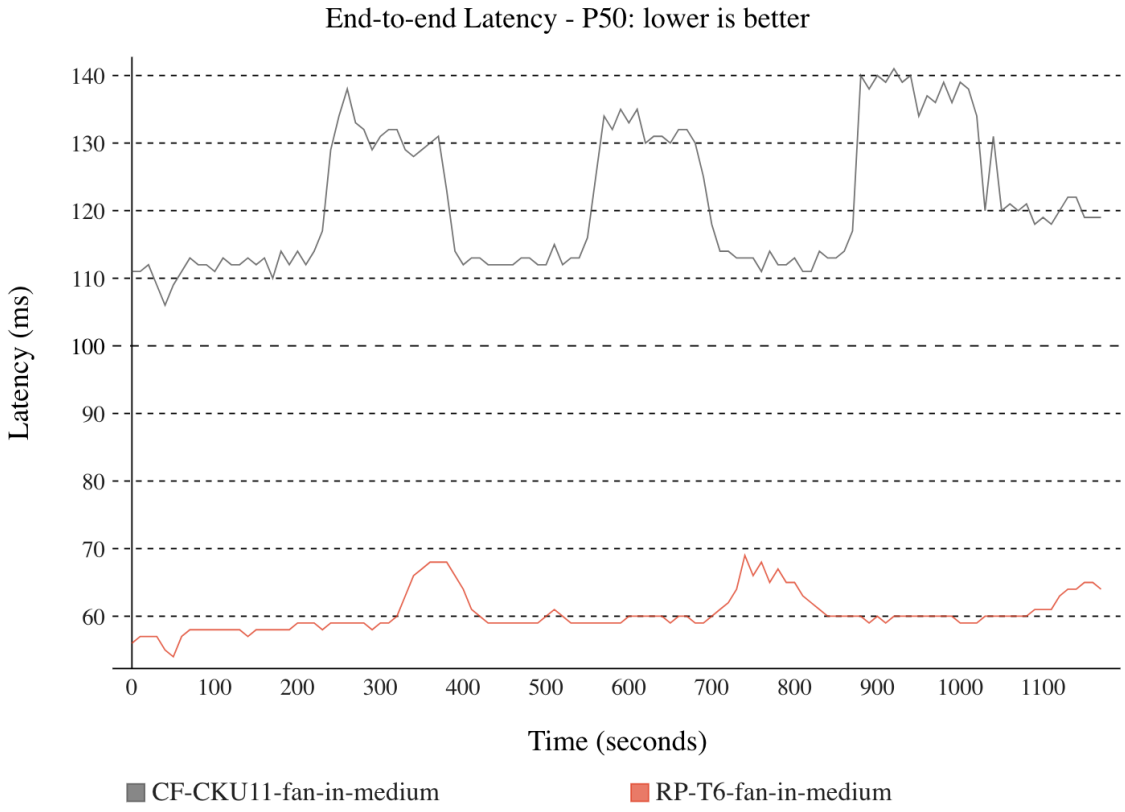
This graph shows publish latency trends throughout the benchmark. The Confluent Dedicated cloud had slightly higher publish latencies and demonstrated more variability in the test, with a large spike toward the end. This large spike contributed to the larger tail latency discrepancies above, while Redpanda's cluster was consistent. The benchmark nodes (not displayed here) remained consistent in resource utilization.

Figure 4. Medium Throughput Fan-In End-to-End Average Latency



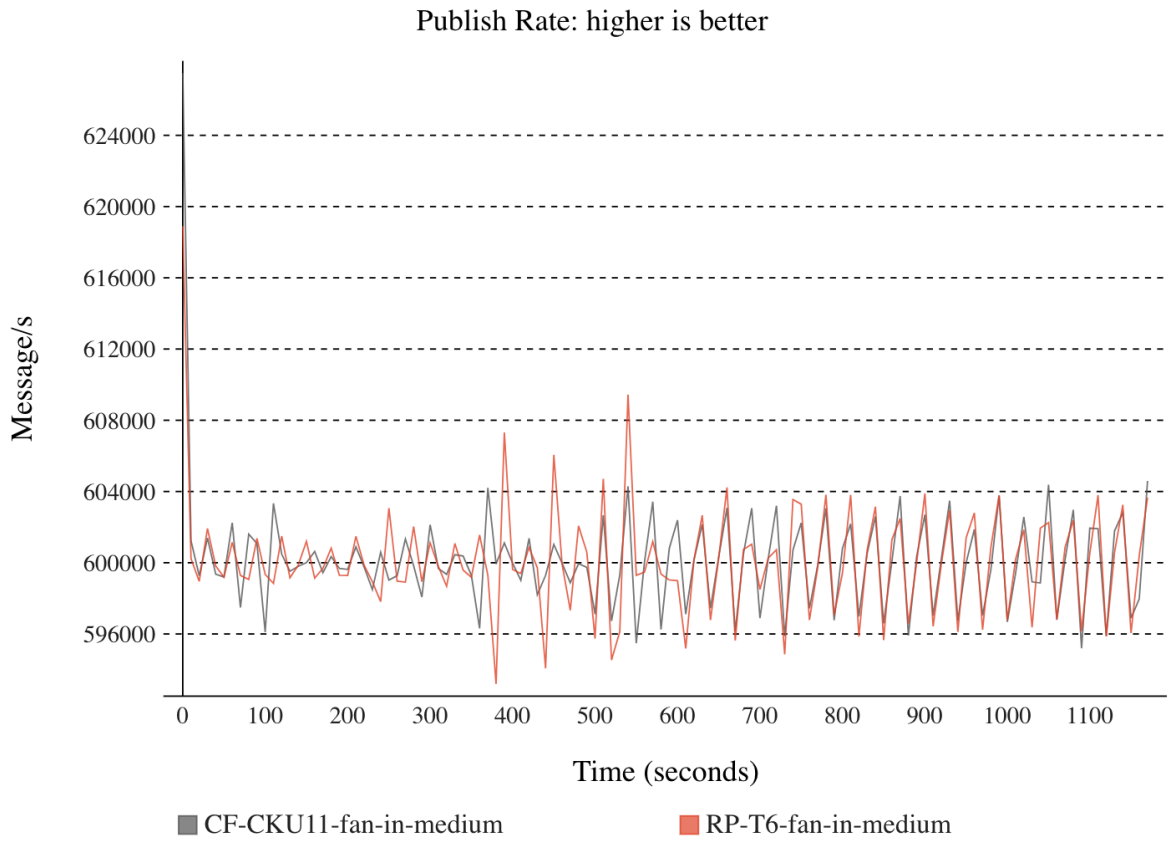
Average end to end latency measured throughout the test is shown here. Redpanda is consistent with a few spikes after 700 seconds into the test. Confluent Dedicated has much larger variation, which correlated to the latency spikes found in publishing, leaning toward the notion that there was some variation in performance with the cluster. With a larger Confluent Dedicated cluster size, this could be mitigated.

Figure 5. Medium Throughput Fan-In P50 Latency



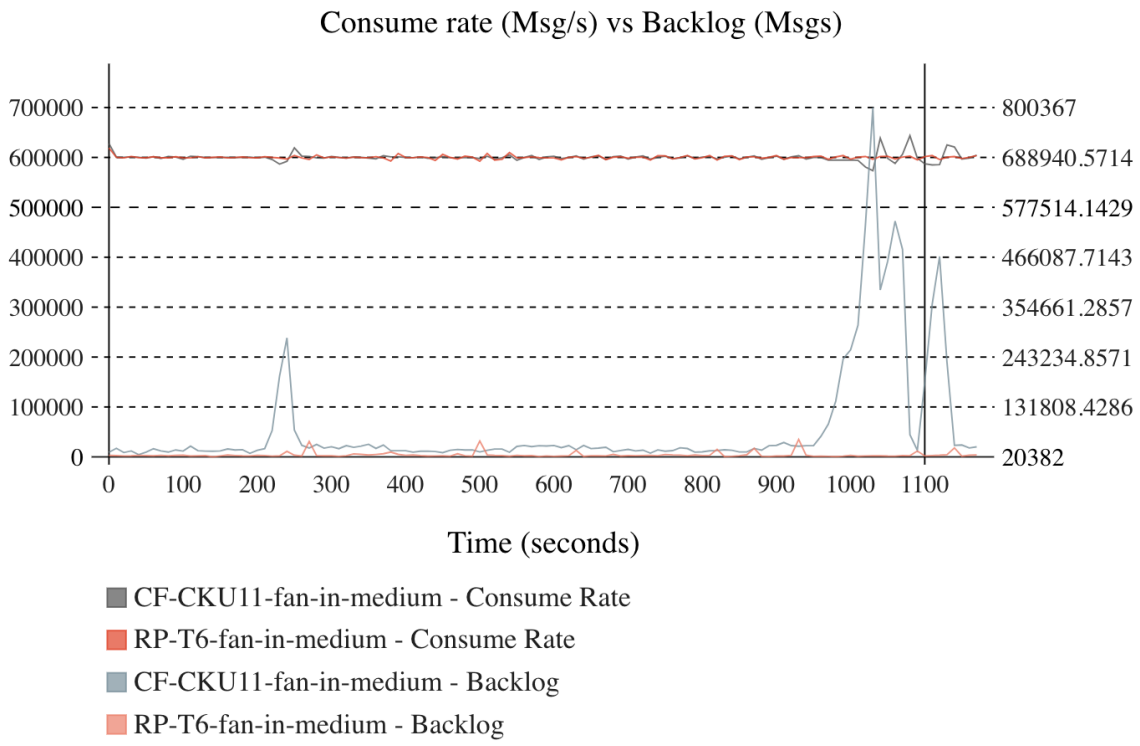
This graph shows the middle range (P50) of end-to-end latency. Both platforms are relatively consistent and within 60-80 milliseconds of each other, although the Confluent Dedicated cluster shows a 30 millisecond cyclic variability. Note that Confluent's servers had latencies around double that of Redpanda's, indicating the Redpanda servers were moving data twice as fast from the producer application to the consumer application.

Figure 6. Medium Throughput Fan-In Publish Rate



This is a simple graph that demonstrates the average publish rate as recorded by the worker nodes in the benchmark publishing metrics. This indicates the publishing applications were behaving consistently and any publish latency spikes seen in the test may have been from buffering vs the application.

Figure 7. Medium Throughput Fan-In Consumer Rates/Backlog

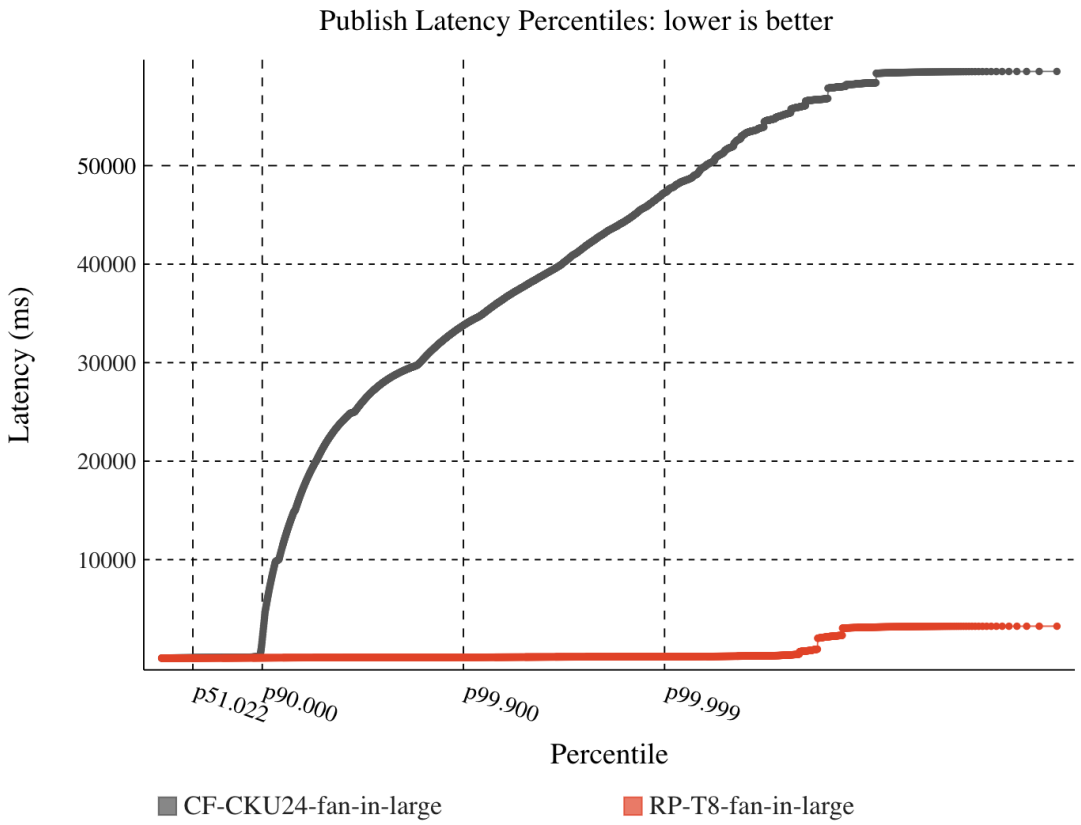


Here we have a graph that tracks both consumption rate and the backlog. Redpanda maintained a consistent data consumption rate and very low backlog. The consumption rate with the Dedicated Confluent cluster kept up the test but had some variability resulting in the backlog growth seen here. These backlogs subsided but contributed to the spikes in the end-to-end latency measurements reflected in the tail latency metrics.

Large Fan In (Telemetry/Observability)

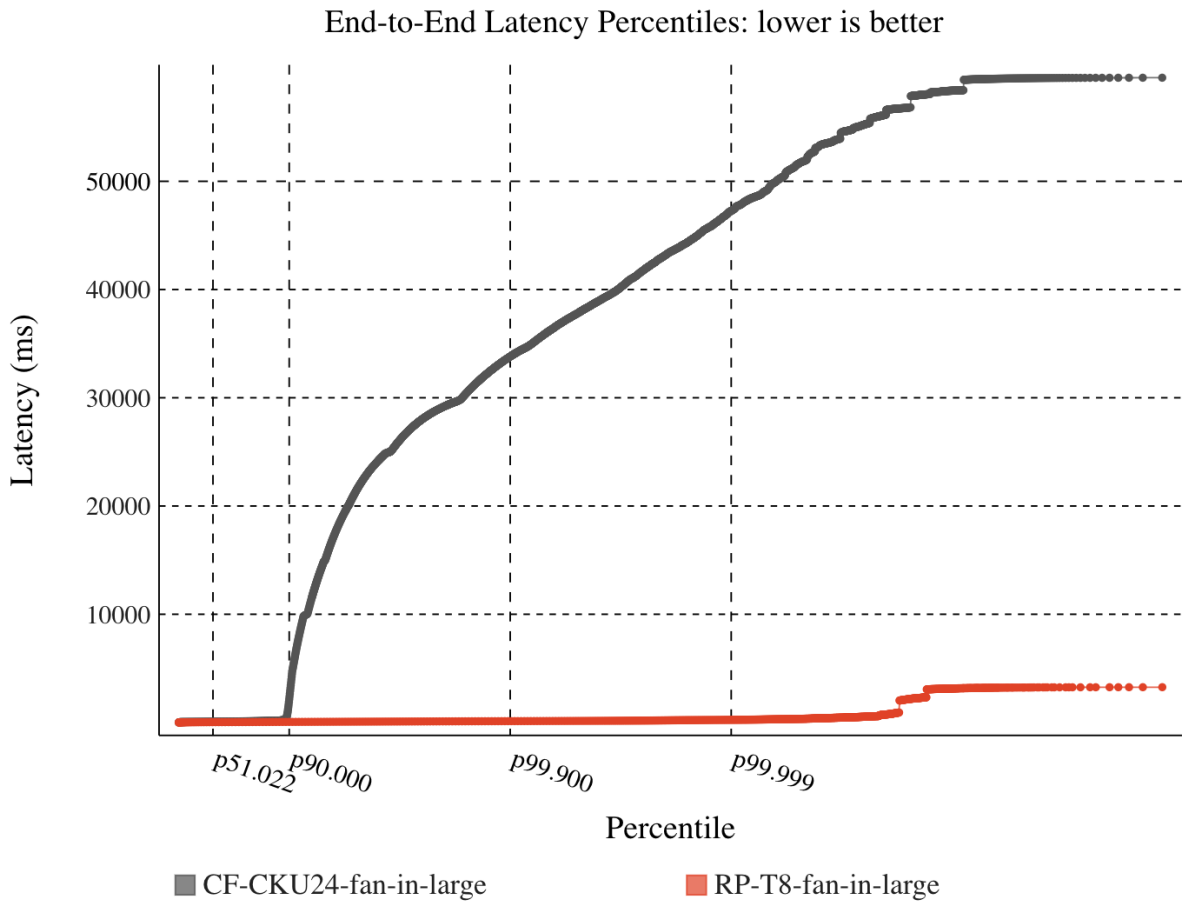
Again, Redpanda had much more consistent performance in terms of latency; in this larger test Redpanda provided lower publish latency as well, which magnified the differences even further. Confluent dedicated was consistent throughout much of the tests but variations in consume rate resulted in temporary consumer backlogs. The benchmark nodes displayed consistent resource usage with CPU around 45% and no network spikes. The confluent cluster was reporting 70-80% load.

Figure 8. High Throughput Fan-In Publish Tail Latency



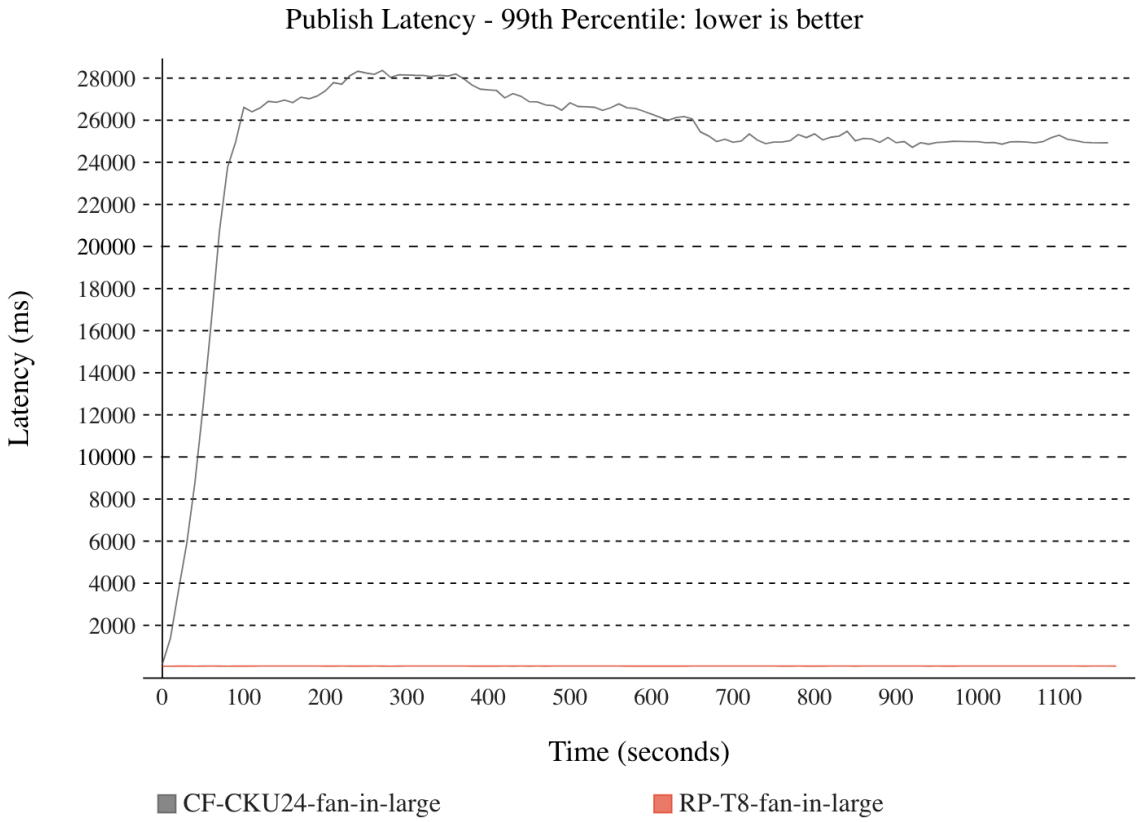
This graph describes tail latencies at P51, P90, P99.0, and P99.999, indicating how close the longer publish times experience in comparison to the majority of publish API calls. Server load will have a significant influence on these metrics. Confluent and Redpanda are equivalent until you pass the 90th percentile of longer latency measurements. Redpanda's cluster provided much more consistent publish latencies. While both tests completed, which was the goal of the benchmark, these results indicate that Confluent's server cluster was under much higher load and provisioning larger clusters could mitigate these tail latencies. While the Confluent Dedicated Cluster could complete the test, in practice, one would likely scale up the cluster to a higher number of CKUs

Figure 9. High Throughput Fan-In End-to-End Tail Latency



As one would expect given the results above, end to end latencies track closely, meaning latency spikes in Confluent Dedicated cluster were occuring on the publish/ingress side.

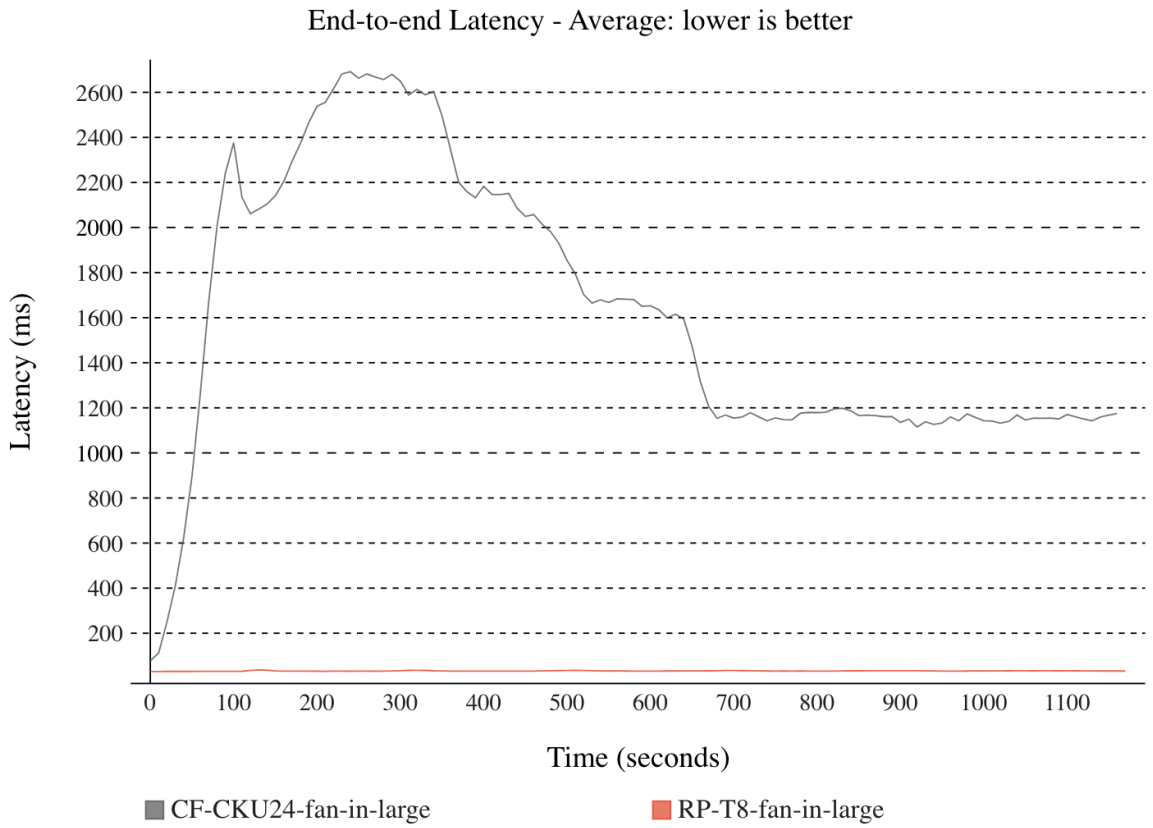
Figure 10. High Throughput Fan-In Publish P99 Latency



These results track with the tail latencies we are seeing in the Confluent's server cluster. Confluent's Dedicated cluster starts off strong but develops a publish delay shortly after starting the test. Regardless, it was able to handle the throughput.

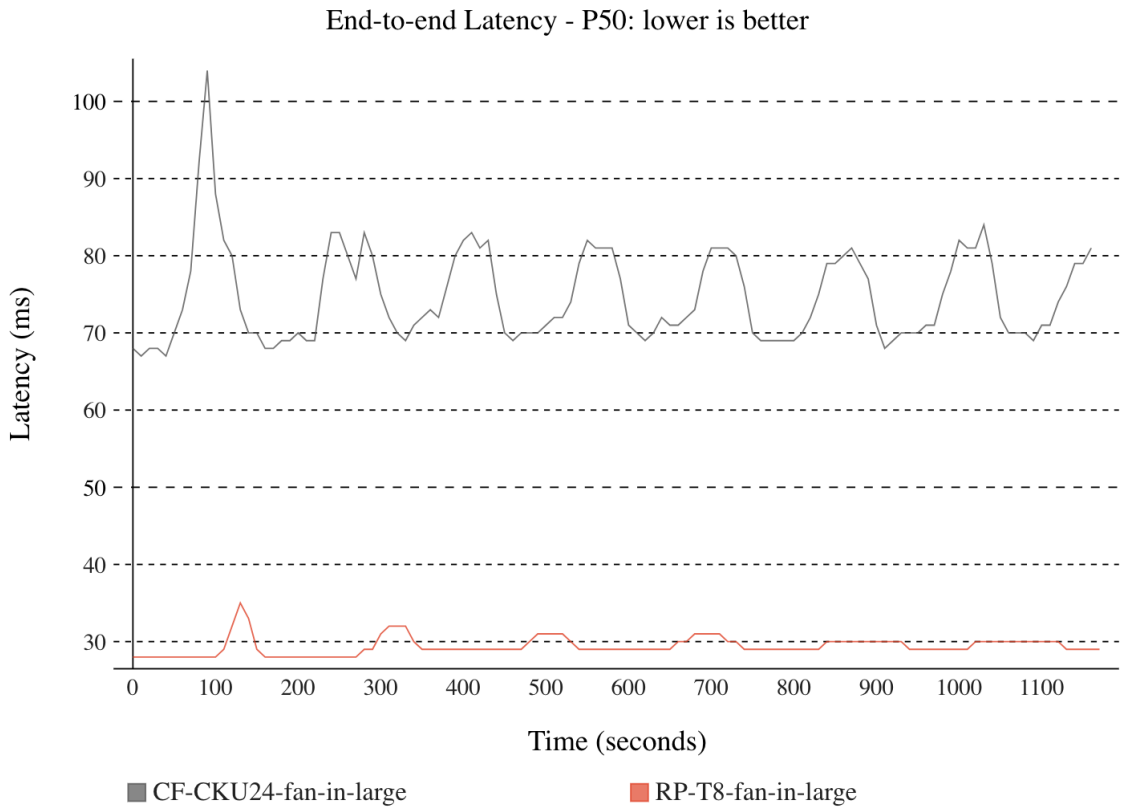
Redpanda's latencies are difficult to see as they are significantly lower at the 99th percentile. These were consistently at 60-65 ms.

Figure 11. High Throughput Fan-In End-to-End Average Latency



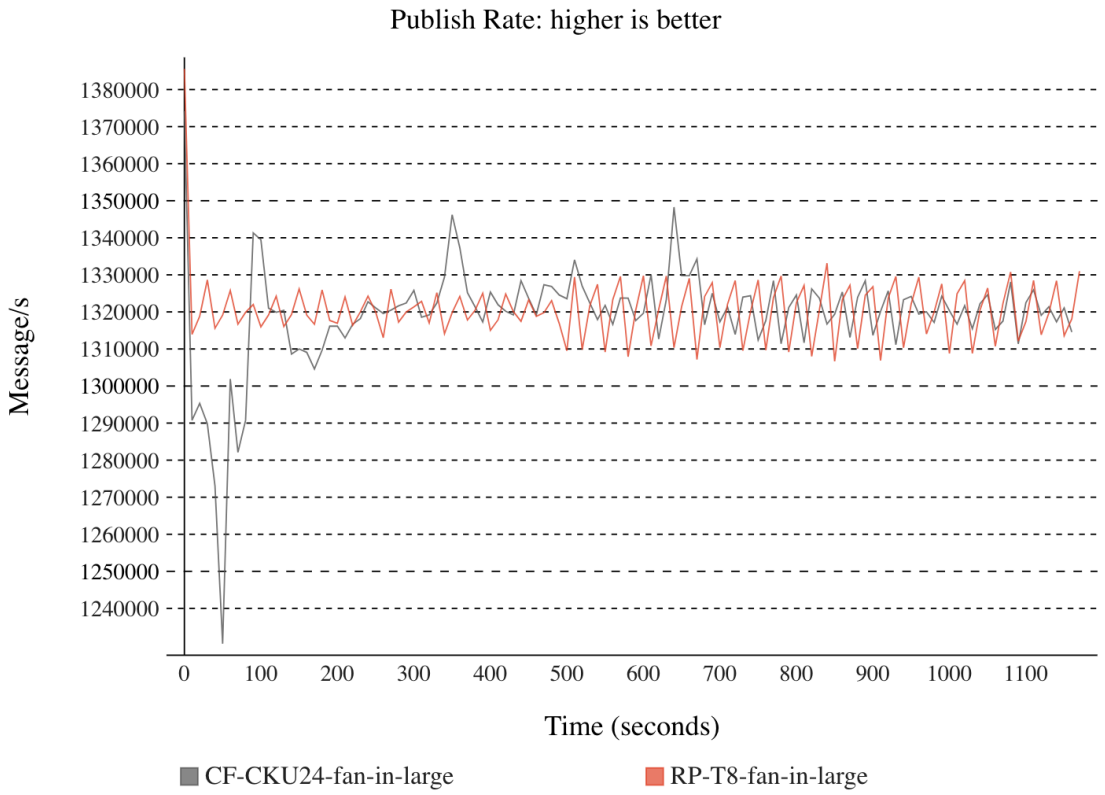
Confluent's dedicated cluster has latency spikes as the test ramps up, then recovers to finish the test with consistent 1200 ms latency measurements. Redpanda's cluster consistently performed with a very low latency with an aggregated end-to-end latency at P50 of 29 ms.

Figure 12. High Throughput Fan-In End-to-End P50 Latency



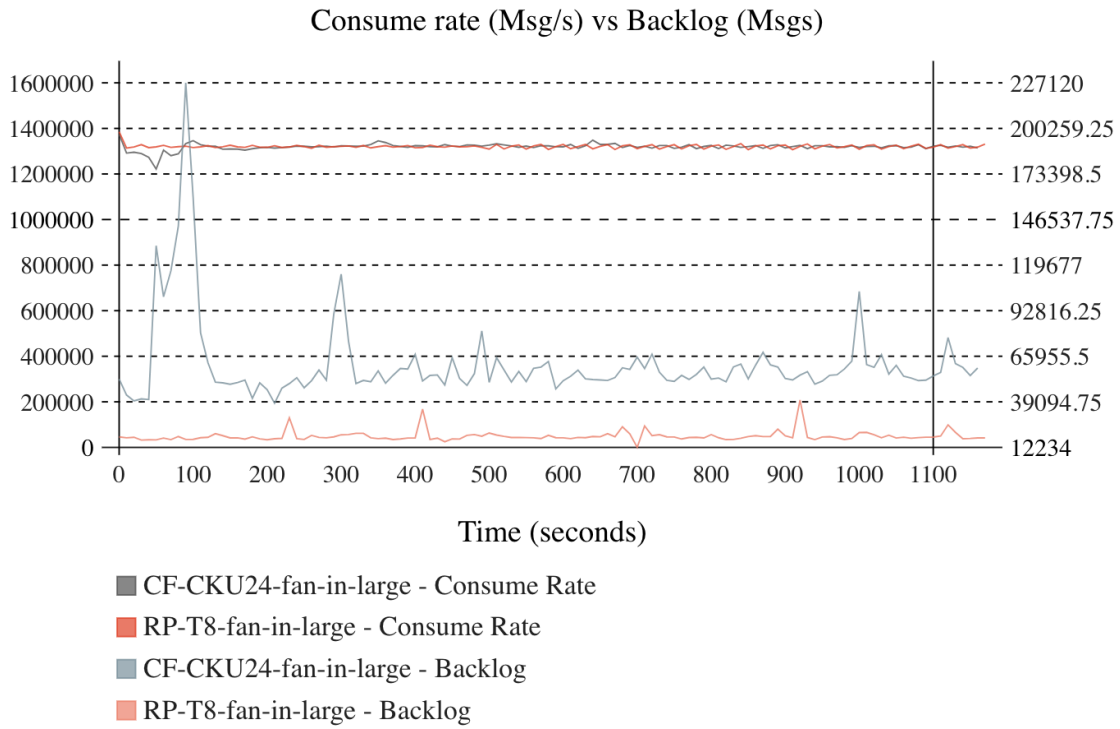
Here we see the P50 range of recorded end-to-end latencies. Redpanda was consistent around 29ms with a few variations. The spike in the beginning of the Confluent benchmark set the cluster back for the rest of the test. Confluent Dedicated had cyclical fluctuation varying between 68ms and 85ms.

Figure 13. High Throughput Fan-In Publish Rate



These results show the average publish rate as recorded by the worker nodes in the benchmark publishing metrics. This indicates the applications were behaving relatively consistent and as before, publish rate dips seen in the test may have been from buffering server cluster or client.

Figure 14. High Throughput Fan-In Consumer Rate/Backlog

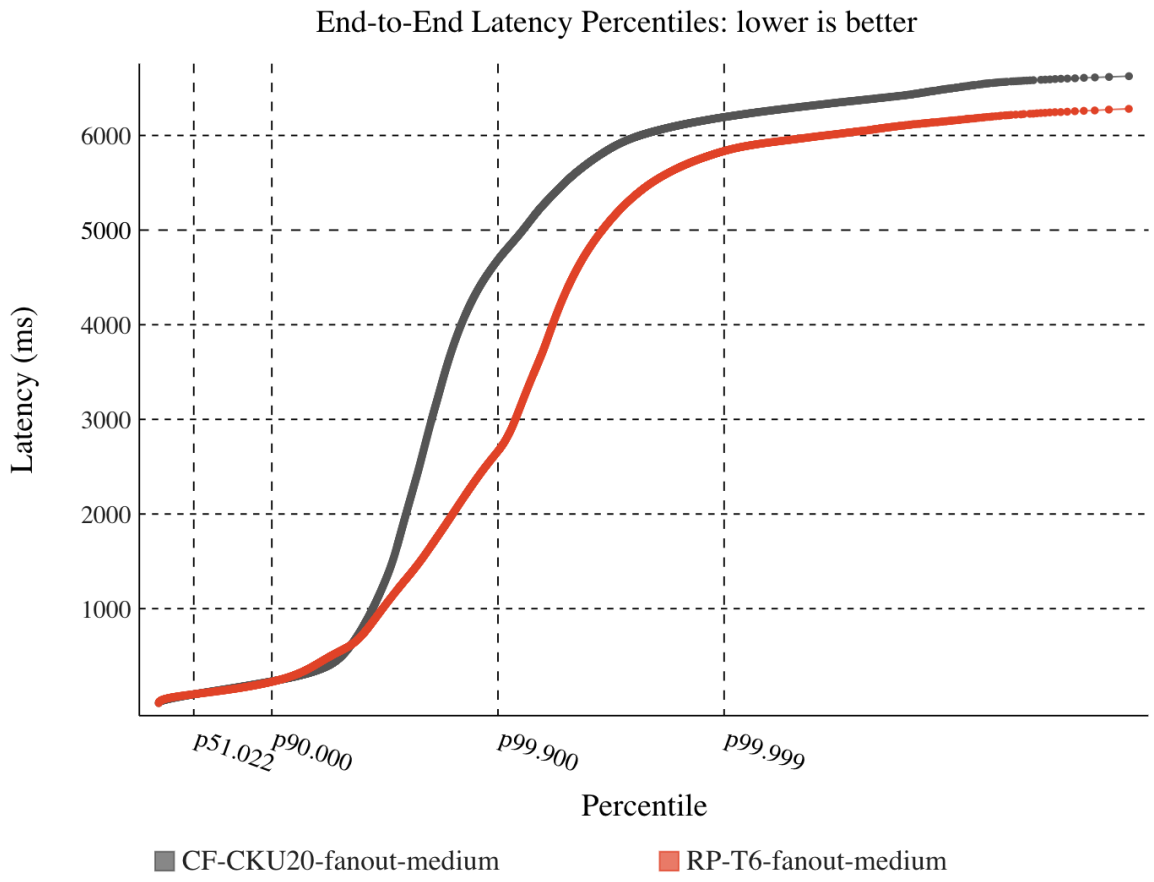


This graph tracks both consumption rate rate and the backlog. Redpanda maintained a consistent data consumption rate and very low backlog, although had more spikes in consumption rate than the mid-sized test. The consumption rate with the Dedicated Confluent cluster kept up the test but had some variability in the beginning resulting in buffering. These backlogs resolved themselves but significantly contributed to differences in the end-to-end tail latency metrics.

Fan Out (Services)

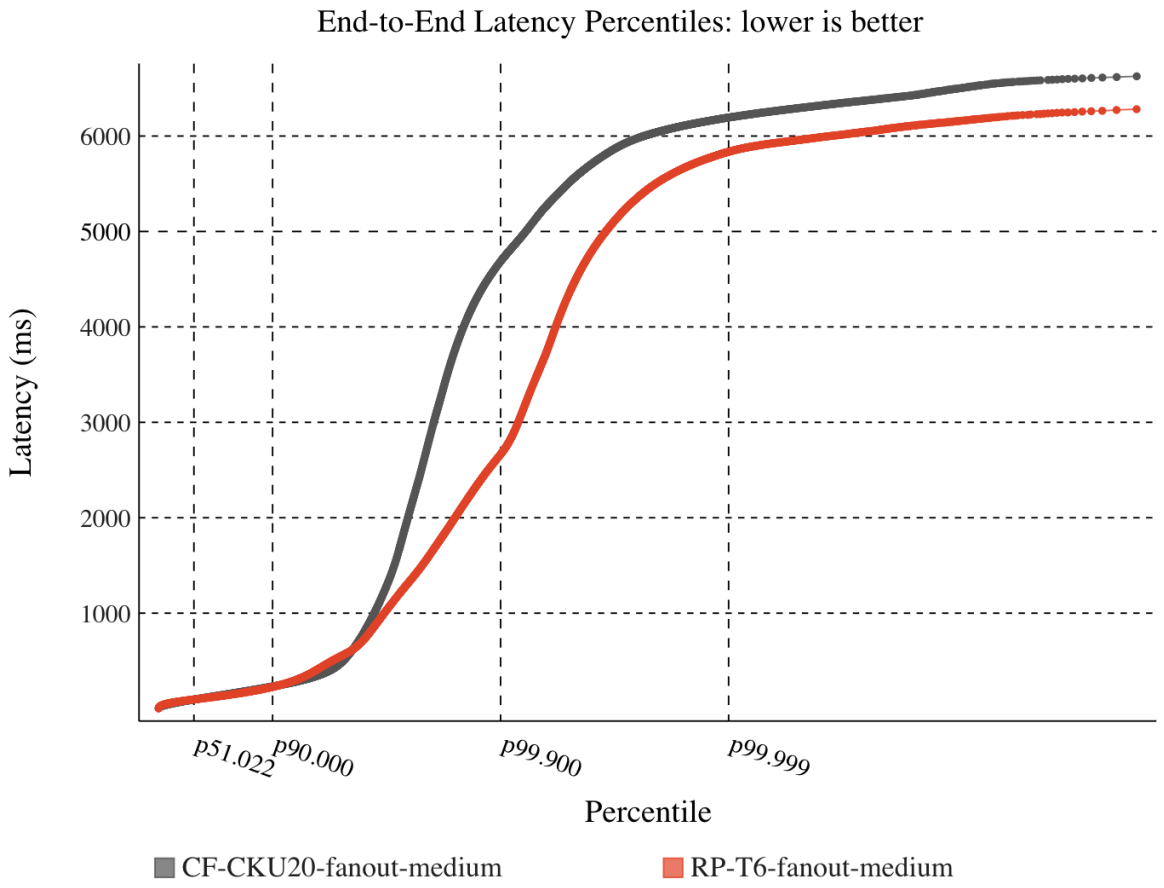
Requests per second limits required provisioning a larger Confluent cluster resulting in much more even metrics here. However, the confluent cluster was reporting 90% load for these benchmarks. The benchmark machines were utilizing about 50% CPU.

Figure 15. Fan-Out Publish Tail Latency



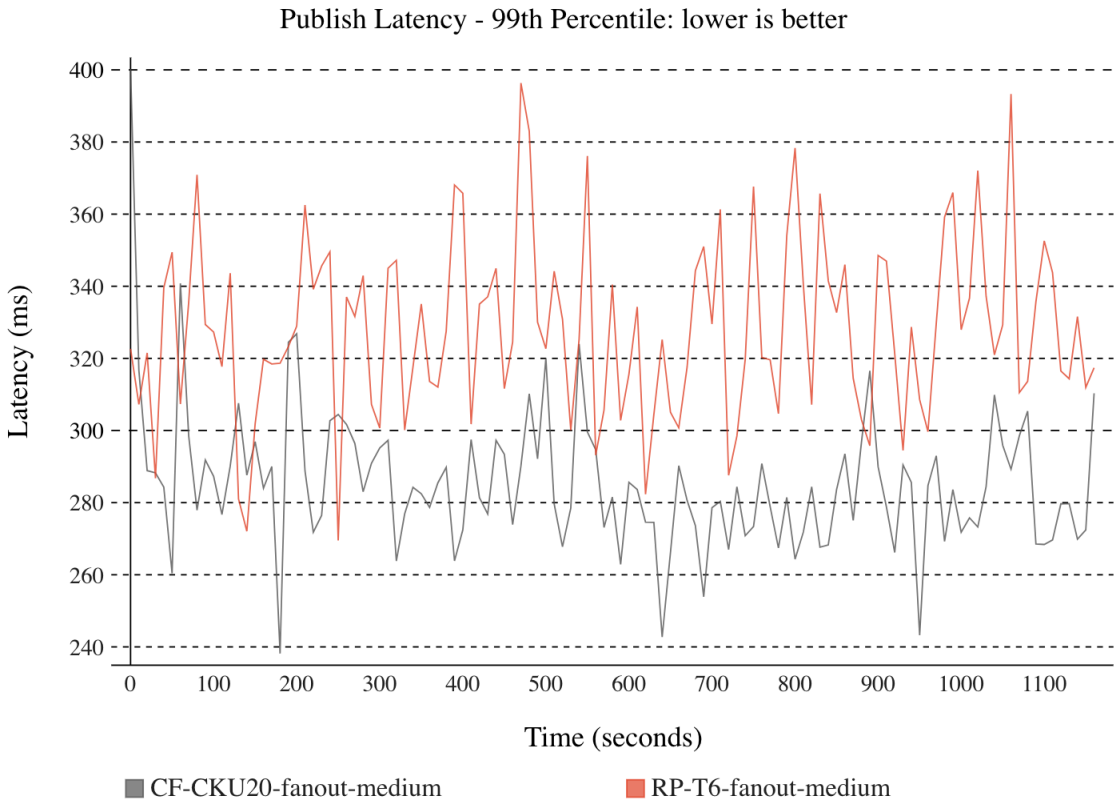
Like the graphs above, this graph describes tail latencies at P51, P90, P99.0, and P99.999, indicating how close the longer publish times experience in comparison to the majority of publish API calls. Server load will have a significant influence on these metrics. In this benchmark Confluent outperformed Redpanda in publish latency, indicating Confluent's Dedicated cluster may perform better with fewer publishers.

Figure 16. Fan-Out End-to-End Tail Latency



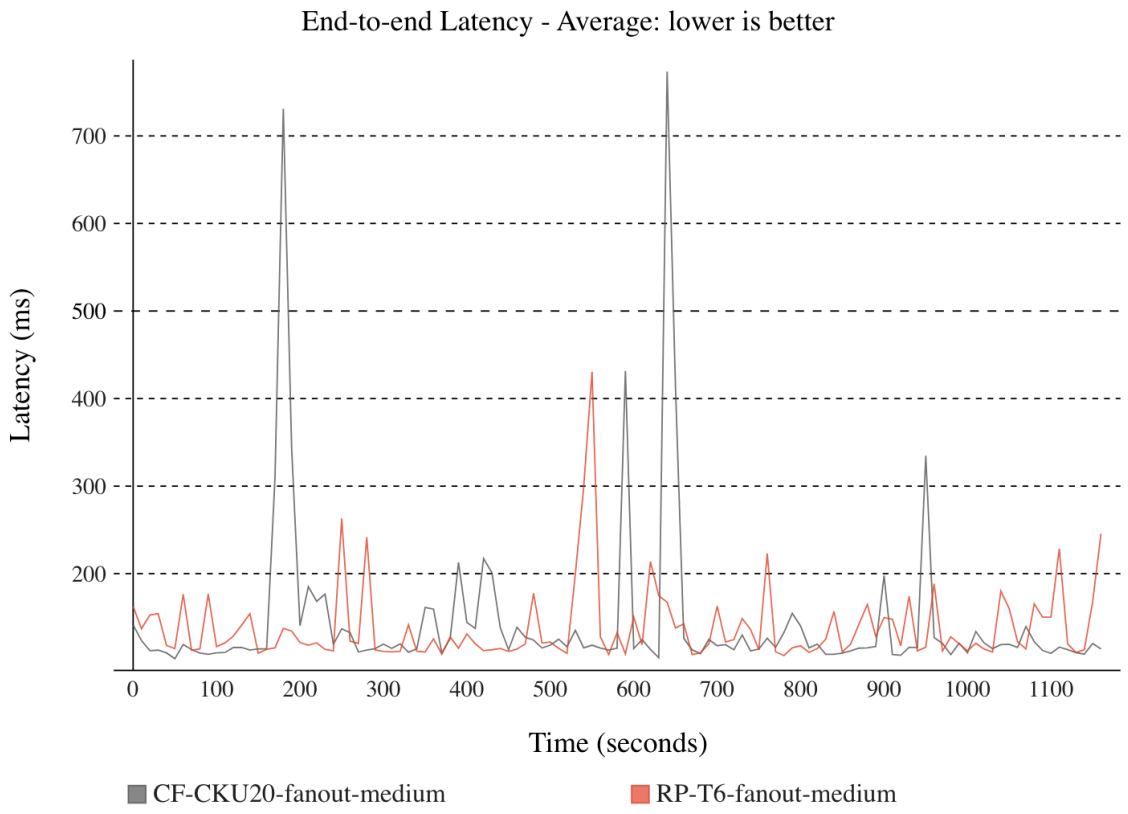
This graph displays tail latencies of the time it took from a message to get from the publisher to the consumer at P51, P90, P99.0, and P99.999, showing the difference in the maximum outliers. Server load and periodic backlogs at the consumer will be reflected in these results here. Both tests succeeded and Confluent and Redpanda's end-to-end latencies are roughly equivalent, with a slight edge for Redpanda.

Figure 17. Fan-Out Publish P99 Latency



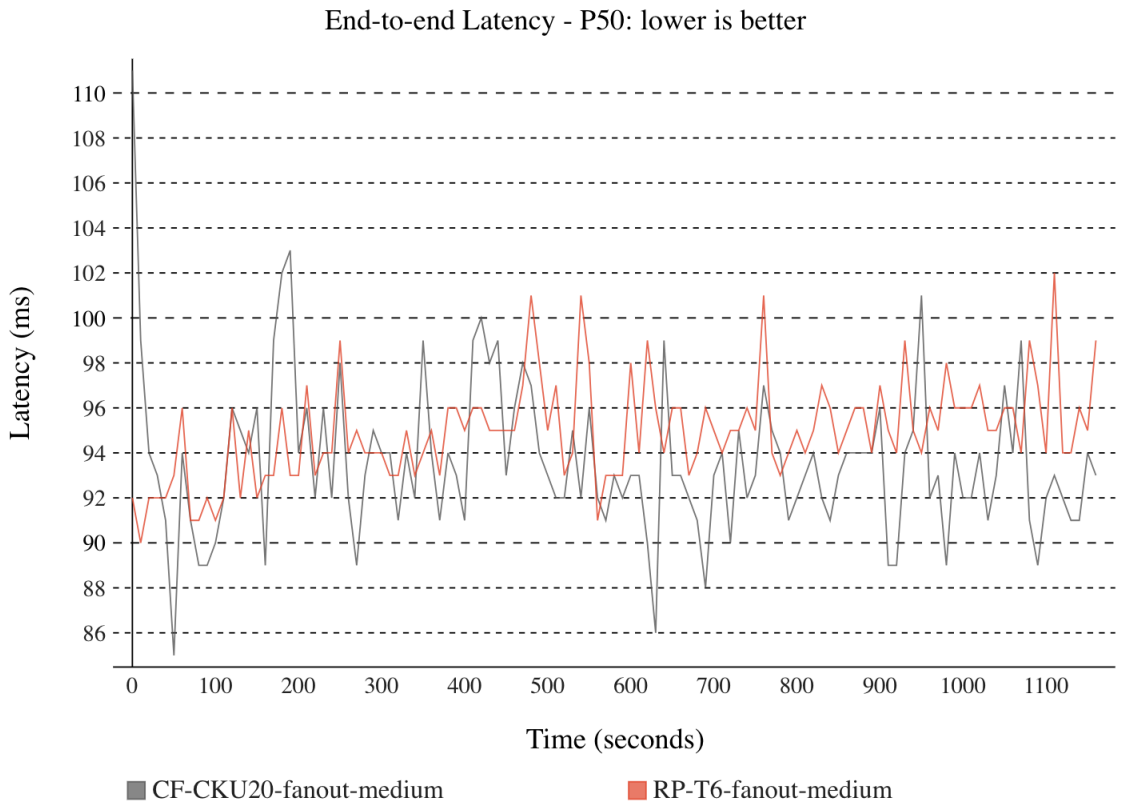
This graph shows the middle range (P50) of publish latency. Both platforms performed roughly the same with Confluent having an advantage in lower publish latency.

Figure 18. Fan-Out End-to-End Average Latency



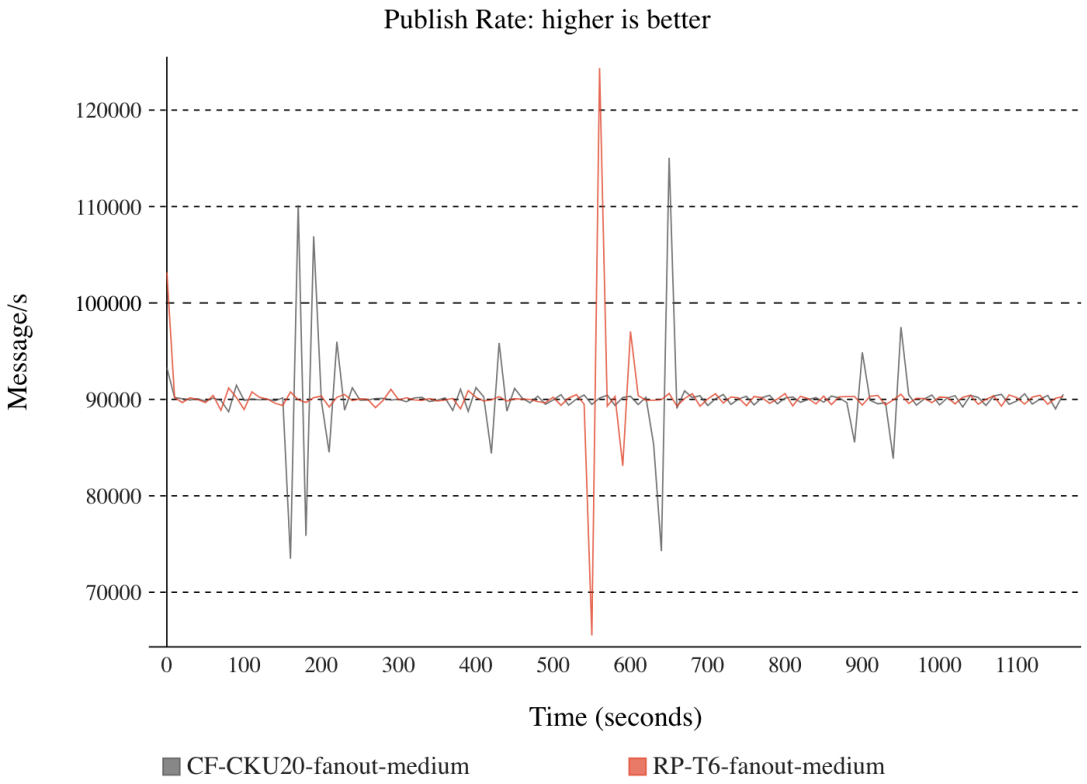
This graph shows the middle range (P50) of end-to-end latency. Both platforms performed similarly although the Confluent Dedicated cluster had a few latency spikes higher than Redpanda.

Figure 19. Fan-Out Publish End-to-End P50 Latency



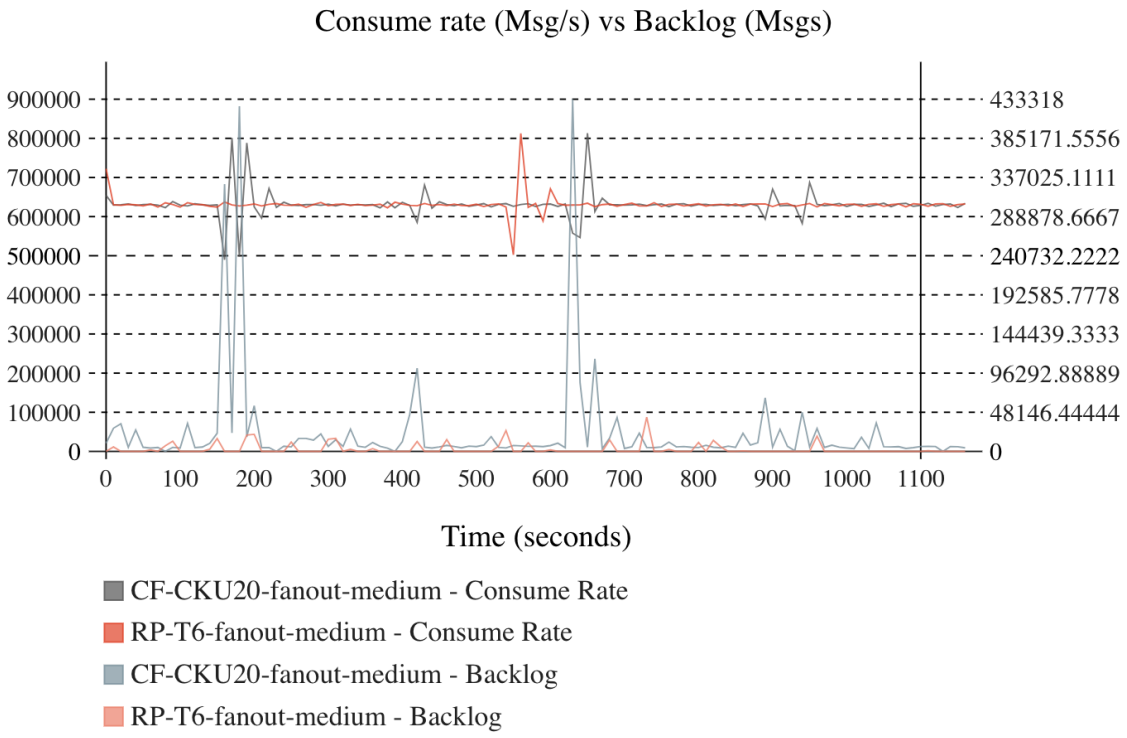
Both platforms demonstrated equivalent performance.

Figure 20. Fan-Out Publish Rate



Both Redpanda and Confluent performed well here. There was one notable drop and spike during the Redpanda benchmark and a few smaller drops/spikes the Confluent's Dedicated benchmark's publish rates.

Figure 21. Fan-Out Consumer Rate/Backlog



Here we see consumption rate with backlog. Consumption rates roughly match the variability in the production rates in the chart above. Notably, the Redpanda BYOC cluster demonstrated less variability in consumption rates than Confluent's Dedicated cluster.

Benchmark Results Summary

Overall, we see that both Redpanda and Confluent can handle equivalent amounts of data through a cluster, publishing and consuming at the same rates, although the Confluent cluster did show more variation than Redpanda.

Redpanda has consistently demonstrated lower tail and average end-to-end latencies. It does not show fluctuations in data delivery rates that Confluent does, thus avoiding periods of backlog growth.

While both clusters can handle significant amounts of ingress and egress, in these benchmarks Redpanda has shown it can deliver better and more consistent results across both fan-in and fan-out usage patterns.

Price-Performance

System cost can be a difficult aspect to compare systems, because vendor platforms vary on their pricing and licensing models. However, both platforms have ways to measure hourly and usage-based pricing that we can use to determine price per performance.

Redpanda Pricing Model

Redpanda's pricing model is licensed annually, by Tier. Each tier represents a set of specifically sized instances to run components of a redpanda deployment. These include a BYOC agent, a connection handling server, data handling servers, and a schema registry instance. Redpanda offers 9 tiers, ranging from a small Tier 1 to a large Tier 9. In our benchmarking, we found Tier 6 able to handle the medium sized tests and Tier 8 was able to handle the largest test the confluent Dedicated cluster could run.

Tier	Annual Price
Tier-6	USD $245,000 / yr
Tier-8	USD $350,000 / yr

Compute

When provisioning a BYOC cluster, Redpanda launches a certain number of instances running their software in your cloud provider, incurring costs. Tier 6 and Tier 8 were used in AWS us-east-2 for the Benchmarks. We'll assume they are running 100% of the time for simple pricing comparisons. We are using dynamic pricing; in practice with reserved instances or using a savings plan would significantly reduce these costs.

Tier 6

Instance Type	Count	USD/hour	Instance Type Cost/HR
t3.small	1	0.0208	0.0208
c5.9xlarge	1	1.5300	1.53
r5.2xlarge	2	0.5040	1.008
img4gn.8xlarge	6	2.9100	17.46
		Hourly	**$20.02**
		Monthly	**$28,827.07**
		Annually	**$345 924.84**

Tier 8

Instance Type	Count	USD/hour	Instance Type Cost/HR
t3.small	1	0.0208	0.0208
c5.18xlarge	1	3.0600	3.06
r5.4xlarge	2	1.0080	2.016
img4gn.8xlarge	12	2.9100	34.92
		Hourly	**$40.02**
		Monthly	**$57,624.19**
		Annually	**$691,490.28**

Storage

S3 is used for storage and widely varies based on data retention, volume, and topic configuration. At the time of this report, pricing is as follows[3]:

- First 50 TB / Month $0.023 per GB
- Next 450 TB / Month $0.022 per GB
- Over 500 TB / Month $0.021 per GB

Price calculations account for Redpanda using local instance storage and S3 for longer term storage. Local storage does not incur costs and S3 includes replication in pricing.

Network

Standard Cloud Provider network costs apply and are billed by the cloud provider accordingly. For this test the following is applied. This is directly from Amazon Web Services[4]:

Data Transfer IN To Amazon EC2 From Internet

All data transfer in $0.00 per GB (AWS ingress has no cost).

Data Transfer OUT From Amazon EC2 To Internet

AWS customers receive 100GB of data transfer out to the internet free each month, aggregated across all AWS Services and Regions (except China and GovCloud). The

[3] https://aws.amazon.com/s3/pricing/

[4] https://aws.amazon.com/ec2/pricing/on-demand/

100 GB free tier for data transfer out to the internet is global and does not apply separately or individually to AWS Regions.

First 10 TB / Month $0.09 per GB
Next 40 TB / Month $0.085 per GB
Next 100 TB / Month $0.07 per GB
Greater than 150 TB / Month $0.05 per GB

Other Costs

Resources such as VPNs, NATs, ingress/egress from other regions and log file storage will accrue some charges. These are dependent on cloud architecture, or in cases may be considered negligible when compared to compute, storage, and network costs.

Confluent Dedicated Pricing Model

Confluent prices by CKU, or Confluent Unit for Kafka, and is easier to predict than Redpanda's BYOC. A CKU is a unit of horizontal scaling for Dedicated Kafka clusters that provide pre-allocated resources.

Confluent cloud allows one to scale up to 24 CKUs in three steps, first provisioning a dedicated cluster, then a 4CKU cluster, and finally up to a 24 CKU cluster. The entire process took around 50 minutes to an hour, although Confluent indicates it may take several hours.

In addition to CKUs, ingress and egress network traffic usage is billed, as well as storage.

Compute Costs

In AWS US East 2, the CKU price is USD $3.8060 per hour.

CKUs	Hourly (USD)	Monthly (USD)	Annually (USD)
11	$41.87	$60,287.04	$723 444.48
20	$76.12	$109,612.80	$1,315,353.60
24	$91.34	$131,535.36	$1,578,424.32

Storage

Confluent charges 0.0001 per GB/Hour

Network

Ingress/Egress from a public cloud:
Read: $0.11 / GB
Write: $0.05 / GB

Medium Fan-In Comparison

With the medium fan-in scenario for one month supporting bursts up to 614MB/sec of ingress and egress, average traffic could be around 50% of that at 300MB/sec. We'll extrapolate the compute, network, and storage values we've discovered in the benchmarks to calculate what an annual comparison could be like.

With a two-day retention of all data, we'll use 50 TB of data and have we'll end up with the following computations:

Throughput
300MB/sec * 60 secs/min * 60 mins/hr * 720 hrs/month = 777,600,000 MB (777,600 GB)

Confluent Network
Ingress: 777600GB * $.05 = $38,880
Egress: 777600GB * $.11 = $85,536
Total: $124,416

Amazon Network
Ingress: 0
Egress:
First 10 TB: 10000 GB at $0.09 = $900 – 100GB free ($9) = $991
Next 40 TB: 40000 GB at $0.085 = $3400
Next 100 TB: 100000 GB at $0.07 = $7000
Remainder: 627600 GB at $.05 = $31380

Total: $42,671/month

Storage
Confluent: 50TB = 50000GB * $0.0001 GB/hr * 720 hours in a month x R3 = $10,800.
Amazon: 50TB = 50000GB * $0.023 GB/month = $1,150.

Annual Expense	Redpanda Tier 6	Confluent CKU 11
Compute	$345,924.86	$723,444.48
Ingress / Egress	$512,052	$1,492,992
Storage	$13,800	$129,600.00
Licensing/Software	$245,000	(N/A - Built into Compute)
Total Annual 100% usage	$1,116,776.86	$2,346,036.48

Redpanda is 52% cheaper (47% of cost) of the Confluent list price for this benchmark. Note that this assumes egress costs for Redpanda cluster. If clients are consuming within the same region of the BYOC cluster, network usage will be less, even zero.

Large Fan-In Comparison

With the large fan-in scenario for one month supporting bursts up to 1350MB/sec of ingress and egress, average traffic could be around 50% of that at 676MB/sec. We'll extrapolate the compute, network, and storage values we've discovered in the benchmarks to calculate what an annual comparison could be like.

With a two-day retention of all data, we'll estimate 120 TB of data and have we'll end up with the following computations:

Throughput
675MB/sec * 60 secs/min * 60 mins/hr * 720 hrs/month = 1,749,600,000MB 1,749,600GB)

Confluent Network
Ingress: 1,749,600GB * $.05 = $87,480
Egress: 1,749,600GB * $.11 = $192,456
Total: $279,936

Amazon Network
Ingress: 0
Egress:
First 10 TB: 10000 GB at $0.09 = $900 – 100GB free ($9) = $891
Next 40 TB: 40000 GB at $0.085 = $3400
Next 100 TB: 100000 GB at $0.07 = $7000
Remainder: 1599600GB at $.05 = $79980
Total: $91,271

Storage
Confluent: 120TB = 120000GB * $0.0001 GB/hr * 720 hrs / month * R3 = $25,920.00.
Amazon: 120TB = 120000GB * $0.023 GB/month = $2,760.

Annual Expense	Redpanda Tier 8	Confluent CKU 24
Compute	$691,490.30	$1,578,424.32
Ingress / Egress	$1,095,252.00	$3,359,232
Storage	$33,120	$311,040.00
Licensing/Software	$350,000	(N/A - Built into Compute)
Total Annual 100% usage	**$2,169,862.30**	**$5,248,696.32**

Redpanda is 58% cheaper (42% of cost) of the Confluent list price for this benchmark. Note that this assumes egress costs for Redpanda cluster. If clients are consuming within the same region of the BYOC cluster, network usage will be less, even zero.

Fan-Out Comparison

With the medium fan-out scenario for one month supporting bursts up to 92MB/sec of ingress and 612 MB/sec egress, average traffic could be around 50% of that at 46MB/sec and 506 MB/sec. As earlier, We'll extrapolate the compute, network, and storage values we've discovered in the benchmarks to calculate what an annual comparison could be like.

With a two-day retention of all data, we'll use 30 TB of data and have we'll end up with the following computations:

Throughput
Ingress: 46MB/sec * 60 secs/min * 60 mins/hr * 720 hrs/month = 119,232,000 MB (119,232 GB / month)
Egress: 506MB/sec * 60 secs/min * 60 mins/hr * 720 hrs/month 1,311,552,000 MB (1,311,553 GB / month)

Confluent Network
Ingress: 119,232 GB * $.05 = $5961.60
Egress: 1,311,553 GB * $.11 = $144,270.83
Total**:** $150,232 / Month

Amazon Network
Ingress: 0
Egress:
First 10 TB: 10000 GB at $0.09 = $900 – 100GB free ($9) = $891
Next 40 TB: 40000 GB at $0.085 = $3400
Next 100 TB: 100000 GB at $0.07 = $7000
Remainder: 1161553 GB at $.05 = $58,077.60
Total: $69,368.60 / Month

Storage
Confluent: 30TB = 30000GB * $0.0001 GB/hr * 720 hours in a month = $2,160.
Amazon: 30TB = 30000GB * $0.023 GB/month = $690.

Annual Expense	Redpanda Tier 6	Confluent CKU 20
Compute	$345,924.84	$1,315,353.60
Ingress / Egress	$832,423.80	$1,802,784.84
Storage	$8,280	$77,760.00
Licensing/Software	$245,000	(N/A - Built into Compute)
Total Annual 100% usage	**$1,431,574.64**	**$3,195,901.44**

Redpanda is 55% cheaper (45% of cost) of the Confluent list price for this benchmark. Note that this assumes egress costs for Redpanda cluster. If clients are consuming within the same region of the BYOC cluster, network usage will be less, even zero.

TCO Calculations

Calculating the total cost of ownership (TCO) in projects is something that happens formally or informally for most enterprise programs. It is also occurring with much more frequency than ever. Sometimes, well-meaning programs will use TCO calculations to justify a program but the measurement of the actual TCO can be a daunting experience, especially if the justification was assessed lightly. This section will focus on the platform costs, including the ever-important support costs. As each platform is fully managed, operations, and maintenance costs are not included.

Costs will fall into these categories:

Infrastructure – Infrastructure is the cloud hardware required to run the platform. In most scenarios this is a separate cost through one of the major cloud providers. In the case of this study, we account for infrastructure costs with Redpanda BYOC which is billed to the customer, and it is safe to assume the infrastructure outside of the server cluster will remain the same regardless of the platform chosen. We calculated the operational cost of the Redpanda Tier (AWS compute, storage, and network) which achieved comparable performance with the Confluent Dedicated Cluster.

Software – Software is the cost of running the platform from the vendor. In this case, we use the on-demand hourly rates for software for Confluent where software fees are built-in into the pricing and we'll use the annual licensing fees from Redpanda.

Support and Training - Along with software, enterprises purchase support packages ensuring they get the expertise needed to diagnose and debug the inevitable problems that occur, in order to meet their SLAs. Redpanda includes support, training, and education in their licensing fees; Confluent provides these ala carte. Support will be included in the TCO calculations.

People Time-Effort (Consulting and FTE) – Consulting models vary widely, but many projects utilize consulting to a high degree for the initial implementation, and Employees will contribute to the initial implementation and largely to the maintenance of these projects. We've made the determination that similar skill sets and time would be contributed to using either Redpanda's BYOC or Confluent's Dedicated cluster, and both offerings are fully managed, so we consider people-time-effort roughly equal and thus will not apply them in our calculations.

When costs reach certain amounts typically custom plans are created so we would expect actual pricing to vary in practice. However, companies can view this as a starting point to understand the performance capabilities and price points of each system.

Infrastructure Costs (Annual)

As noted above, infrastructure costs for the Kafka enabled applications should be the same. Confluent does not incur any additional infrastructure costs while Redpanda's BYOC does.

Benchmark	CKUs	Confluent Cost / Yr.	Redpanda Tier	Redpanda Cost / Yr. (USD)
Medium Fan-In	11	N/A	6	$871,776.86
Medium Fan-Out	20	N/A	6	$1,186,628.06
Large Fan-in	24	N/A	8	$1,819,862.30

Software/Service Costs (Annual)

We will track Confluent's Dedicate CKU usage, network, and storage as software costs. In addition to infrastructure costs, Redpanda requires licensing fees which are included here.

Benchmark	CKUs	Confluent Cost / Yr.	Redpanda Tier	Redpanda Cost / Yr. (USD)
Medium Fan-In	11	$2,346,036.48	6	$245,000
Medium Fan-Out	20	$3,195,901.44	6	$245,000
Large Fan-in	24	$5,248,696.32	8	$350,000

Support Costs

Confluent allows support to be purchased ala carte, although the support packages differ. Redpanda's customer success package includes 24x7 support, training, a dedicated customer success engineer and product updates so most closely mirrors the premium level of support for Confluent, for which pricing is unavailable.

Stepping down, Confluent has a business tier, which provides response times roughly equivalent Redpanda's with a shorter response time for critical issues. We will use available pricing for the business tier in our calculations with the understanding that support may end up costing more with Confluent for those that want to leverage premium services Confluent Offers.

While Redpanda provides more guidance and reviews, the SLA's of the support plans are similar, with Redpanda offering a 2 hour SLA for level 1 (critical) support, 4 business hours for level 2, and 8 business hour level 3 support. Confluent's business tier offers 1 hour SLAs for level 1, 4 hours, and 8 hours, based on severity.

Redpanda Support Costs

Built into annual Subscription (Software/Services).

Confluent Business Tier Support Costs

Greater of $1,000 per month, 10% of first $50,000 of usage, 8% of next $50,000 of usage, 6% of next $900,000 of usage, 3% of usage over $1M per month

Benchmark	CKUs	Annual Software Cost	Annual Support Costs
Medium Fan-In	11	$2,346,036.48	$176,762.19
Medium Fan-Out	20	$3,195,901.44	$227,754.09
Large Fan-in	24	$5,248,696.32	$364,587.02

People Time Effort

We will exclude people-time-effort from our calculations as both platforms are fully managed, incurring negligible operational costs by the customer and technical personnel will require the exact same skillset (Kakfa client API). Operators will, regardless of the platform, need a similar skillset is required to operate clusters:

- Kafka knowledge at a level to determine partition count, requests per second, connection attempts per second, ingress and egress rates, connection counts.
- An understanding of Kafka Users or API keys, ACLs and roles.

- For Redpanda, the ability to use command line tooling with cloud provider security to provide access to your infrastructure.
- For most cases, the ability to set up a VPN and private links in the cloud provider of choice.

In setting up the benchmarks, the effort to get a dedicated cluster and BYOC cluster was roughly the same, with provisioning a Redpanda BYOC cluster in about 45 minutes. Provisioning and expanding a confluent cluster usually took between 40-60 minutes, although the Confluent process to expand clusters indicates it may take several hours in cases. As this will be done infrequently, in the context of the scope of this report this time is certainly negligible.

Some effort will be required to monitor the cluster for load and errors; these costs are small compared to the other costs.

Total Cost of Ownership

To break down the total cost of ownership, we've established that People-Time Effort will be roughly equal, and only a small percentage of the costs here, leaving infrastructure, software and services, and support costs.

Medium Fan-in Use Case

Item	Confluent	Redpanda
Infrastructure	N/A	$871,776.86
Software/Services	$2,346,036.48	$245,000
Support	$176,762.19	Included in Software
Total	**$2,522,798.67**	**$1,116,776.86**

Redpanda is 55% cheaper (45% of cost) of the Confluent list price for this benchmark. Note that this assumes egress costs for Redpanda cluster. If clients are consuming within the same region of the BYOC cluster, network usage will be less, even zero. For an enterprise, this means you could scale Redpanda up to a higher tier or add another cluster.

Effectively you can run **two** Redpanda BYOC clusters for the price of **one** Confluent Dedicated Cluster.

Large Fan-in Use Case

Item	Confluent	Redpanda
Infrastructure	N/A	$1,819,862.30
Software/Services	$5,248,696.32	$350,000
Support	227,754.09	Included in Software
Total	$5,476,450.41	$2,169,862.30

The price differential in this case is interesting. **Redpanda is 60% cheaper** (**40% of cost)**, while providing roughly equivalent support and performance. For every dedicated confluent cluster an enterprise could run three Redpanda clusters allowing you to process at least 3x streaming data for the same price.

In this comparison, you can run **two** Redpanda BYOC clusters for the price of **one** Confluent Dedicated cluster, with room to increase your cluster size.

Medium Services Use Case

Item	Confluent	Redpanda
Infrastructure	N/A	$1,186,628.06
Software/Services	$3,195,901.44	$245,000
Support	364,587.02	Included in Software
Total	$3,560,488.46	$1,431,628.06

Finally, our last use case **Redpanda is 59% cheaper** (**41% of cost)**, while providing roughly equivalent support and performance. For every dedicated confluent cluster an enterprise could run three Redpanda clusters allowing you to again process at least 3x streaming data for the same price.

In this comparison, you can run **two** Redpanda BYOC clusters for the price of **one** Confluent Dedicated cluster

Conclusion

Both services proved they can handle enterprise level workloads at scale. Using the smallest cluster size possible for each benchmark, Redpanda demonstrated it can handle load better at the edges of their tiers with up to **60% savings**, including egress costs and even with the more costly higher dynamic cloud provider pricing, while Confluent can achieve similar performance when provisioning higher CKUs clusters. If confluent clients are running in the same cloud region, egress costs are zero and the savings are greater. Depending on the use case, a company can run two Redpanda BYOC clusters for the price they would spend on a single Confluent Cluster.

A company can pay less for a larger Redpanda BYOC Cluster and deliver data to more applications faster and cheaper than Confluent. This translates to faster business decisions and completion of more transactions in a period of time. Ultimately, this provides a competitive advantage.

Companies should be cognizant of hidden costs with BYOC, and these are not necessarily easy to project. However, in our testing that was scoped to basic cluster performance, we've found Redpanda's highest performing offering (BYOC) to be significantly more cost effective to run than Confluent's highest performing offering, the Dedicated Cluster.

Confluent is a great choice for those who can value their larger toolset and ecosystem, and value additional features and connectors over cluster price.

Redpanda BYOC is a great choice for those who wish to run large scale workloads in their own cloud environment, have a committed cloud spend, or require consistent performance and lower latencies.

With each platform, higher levels of service can be purchased to meet throughput and latency requirements.

To summarize, with the ability to provision 2 Redpanda BYOC clusters for the cost of each Confluent Dedicated cluster, businesses can process significantly more data with Redpanda while allowing a budget for scaling and future proofing their system.

Appendix: Changes to the Open Messaging Benchmark Code Base

The Open Messaging Benchmark had some limitations that required some changes for tests to complete.

Worker Close Timeout

First, the open messaging benchmarks have an issue in that they close the consumers before writing the output file, timing out when there are large numbers of consumers.As the worker.stopAll() call is also issued later in the benchmark, it is superfluous for the Kafka driver and could be removed.

Here is the workaround we used:

```
$ git diff
diff --git a/benchmark-
framework/src/main/java/io/openmessaging/benchmark/WorkloadGenerator.java
b/benchmark-
framework/src/main/java/io/openmessaging/benchmark/WorkloadGenerator.java
index dc91353..818f1c6 100644
--- a/benchmark-
framework/src/main/java/io/openmessaging/benchmark/WorkloadGenerator.java
+++ b/benchmark-
framework/src/main/java/io/openmessaging/benchmark/WorkloadGenerator.java
@@ -143,7 +143,6 @@ public class WorkloadGenerator implements AutoCloseable
{
        TestResult result = printAndCollectStats(workload.testDurationMinutes,
TimeUnit.MINUTES);
        runCompleted = true;

-       worker.stopAll();
        return result;
    }
```

General HTTP Timeout

Confluent Dedicated Cloud service takes a significant amount of time to create topics as it appears to rate-limit topic creation. The change below is not necessarily a fix, but instead provides plenty of time for the benchmark to setup and get kicked off.

```
diff --git a/benchmark-
framework/src/main/java/io/openmessaging/benchmark/worker/HttpWorkerClient.jav
a b/benchmark-
framework/src/main/java/io/openmessaging/benchmark/worker/HttpWorkerClient.jav
a
index 513b041..ab56f9c 100644
--- a/benchmark-
framework/src/main/java/io/openmessaging/benchmark/worker/HttpWorkerClient.jav
a
+++ b/benchmark-
framework/src/main/java/io/openmessaging/benchmark/worker/HttpWo
rkerClient.java
@@ -58,7 +58,7 @@ public class HttpWorkerClient implements Worker {
     private final String host;

     public HttpWorkerClient(String host) {
-
this(asyncHttpClient(Dsl.config().setReadTimeout(600000).setRequestTimeout(6000
00)), host);
+
this(asyncHttpClient(Dsl.config().setReadTimeout(6000000).setRequestTimeout(600
0000)), host);
     }

     HttpWorkerClient(AsyncHttpClient httpClient, String host) {
```

About Redpanda

Redpanda is a revolutionary streaming data platform that's changing the game for real-time data processing and analytics. Built on a unique, thread-per-core architecture, Redpanda combines the simplicity of a message queue with the power of a streaming database, allowing users to unify event-driven architecture, stream processing, and data integration in a single, Kafka-compatible platform. With Redpanda, developers can build scalable, fault-tolerant, and highly performant data pipelines that unlock new use cases and revenue streams, from real-time analytics and event-driven microservices to IoT data processing and AI/ML model training. By providing a streamlined, easy-to-use alternative to traditional streaming data solutions, Redpanda is empowering businesses to transform their data infrastructure, unlock new insights, and drive innovation in the digital age.

For more, visit **http://www.redpanda.com**.

About McKnight Consulting Group

Information Management is all about enabling an organization to have data in the best place to succeed to meet company goals. Mature data practices can integrate an entire organization across all core functions. Proper integration of that data facilitates the flow of information throughout the organization which allows for better decisions – made faster and with fewer errors. In short, well- done data can yield a better run company flush with real-time information... and with less costs.

However, before those benefits can be realized, a company must go through the business transformation of an implementation and systems integration. For many that have been involved in those types of projects in the past – data warehousing, master data, big data, analytics - the path toward a successful implementation and integration can seem never-ending at times and almost unachievable. Not so with McKnight Consulting Group (MCG) as your integration partner, because MCG has successfully implemented data solutions for our clients for over a decade. We understand the critical importance of setting clear, realistic expectations up front and ensuring that time-to-value is achieved quickly.

MCG has helped over 100 clients with analytics, big data, master data management and “all data” strategies and implementations across a variety of industries and worldwide locations. MCG offers flexible implementation methodologies that will fit the deployment model of your choice. The best methodologies, the best talent in the industry and a leadership team committed to client success makes MCG the right choice to help lead your project.

MCG, led by industry leader William McKnight, has deep data experience in a variety of industries that will enable your business to incorporate best practices while implementing leading technology. See www.mcknightcg.com.