InfoWorld

Review: Redpanda gives Kafka a run for its money

The Kafka-compatible distributed event streaming platform excels in latency and performance and offers a glimpse into the future of streaming with inline WebAssembly transforms and more.

By Martin Heller
InfoWorld | May 25, 2022

AT A GLANCE

Redpanda

★★★★½

Apache Kafka is an open-source Java/Scala distributed event streaming platform for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications. As I have explained, one downside of Kafka is that setting up large Kafka clusters can be tricky. Another downside is that Kafka uses the Java virtual machine (JVM), which introduces lag because of memory garbage collection. Adding even more complexity, Kafka has until recently required Apache ZooKeeper for distributed coordination, and it requires a separate schema registry process.

Redpanda (previously called Vectorized) is a Kafka plug-in replacement written primarily in C++ using the Seastar asynchronous framework, and the Raft consensus algorithm for its distributed log. Redpanda does not require using ZooKeeper or the JVM, and its source is available on GitHub under the Business Source License (BSL). It's not technically open source as defined by the Open Software Foundation, but that doesn't matter to me because I have no plans to offer Redpanda as a service.

Redpanda vs. Kafka

As you might expect from the reimplementation in C++, Redpanda has significantly lower latency and higher performance than Kafka. It's also much easier to install and tune.

Figure 1 shows latency charts for Redpanda and Kafka. The left-hand chart shows average latency versus time, and the right-hand chart shows latency versus percentile. Redpanda's caption isn't exactly false, but it does exaggerate. I'd rephrase it and say that Kafka's average latency is 6 to 10 times higher than Redpanda's, and that Kafka's tail latency is up to 40 times higher than Redpanda's.

Redpanda's architecture and optimizations

Kafka has a complicated architecture that is designed to scale, as shown in Figure 2 below. Redpanda has a simpler architecture, shown in Figure 3, but still outperforms Kafka by a large factor, especially when it comes to latency.

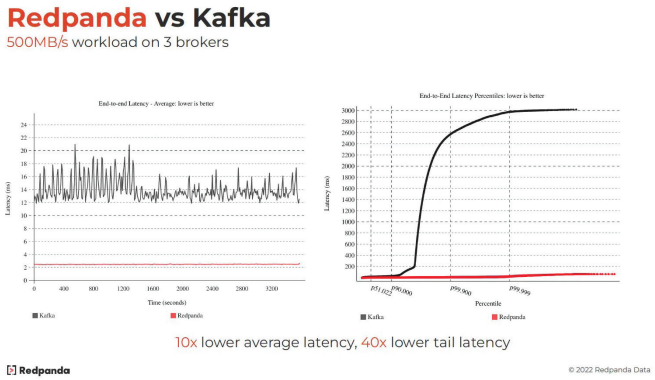

Figure 1. Latency charts for Redpanda and Kafka. IDG

Redpanda boasts a number of optimizations over Kafka, starting with jettisoning the JVM and ZooKeeper, and continuing from there. Even beyond its reimplementation in C++, Redpanda uses an asynchronous, shared-nothing, thread-per-core model, with no locking, minimal context switching, and thread-local memory access. It scales well, both vertically (bigger, faster nodes) and horizontally (more nodes).

The Raft consensus algorithm speeds writes to a cluster, and

> "Redpanda goes beyond the Kafka protocol into the future of streaming with inline WebAssembly transforms and geo-replicated hierarchical storage/shadow indexing.

Redpanda does automatic leader and partition balancing. In production mode, Redpanda auto-tunes. Simple one-shot tuning and configuration sets kernel parameters, and auto-detects and optimizes for available hardware.

Kafka relies on the Linux page cache to accelerate disk I/O,

which has issues such as flushing the cache after a backup. Redpanda bypasses the Linux page cache to avoid its design flaws; instead, it uses custom memory management and I/O scheduling.

Redpanda goes beyond the Kafka protocol into the future of streaming with inline WebAssembly transforms and geo-replicated hierarchical storage/shadow indexing. WebAssembly (WASM) is a high-performance, system-independent byte code system that is compiled from other languages. Alexander Gallego, the founder and CEO of Redpanda, has said that "What JavaScript did for the web in the late '90s is what WebAssembly can do for server-side applications." Redpanda uses WASM to perform data transformations on streams without needing to use an external processor such as Apache Flink.

According to Redpanda, Shadow Indexing is a multi-tiered remote storage solution that provides the ability to archive log segments to a cloud object store in real time as the topic is being produced. You can recover a topic that no longer exists in the cluster, and replay and read log data as a stream directly from cloud storage even if it doesn't exist in the cluster. Shadow Indexing provides a disaster recovery plan that takes advantage of infinitely scalable storage systems, is easy to configure, and works in real time.

Redpanda supports observability via Prometheus and Grafana. It has a metrics endpoint, and the rpm generate command can create configuration for both Prometheus and Grafana.

Architectural overview

As shown in Figure 2, Kafka's architecture is designed to scale. Each component is given its own servers, and if any layer becomes overloaded, you can scale it independently by adding nodes to that specific layer. For example, when adding applications that use the Confluent REST proxy, you may find that the REST proxy no longer provides the required throughput, while the underlying Kafka brokers still have spare capacity. In this case, you can scale your entire platform simply by adding REST proxy nodes.

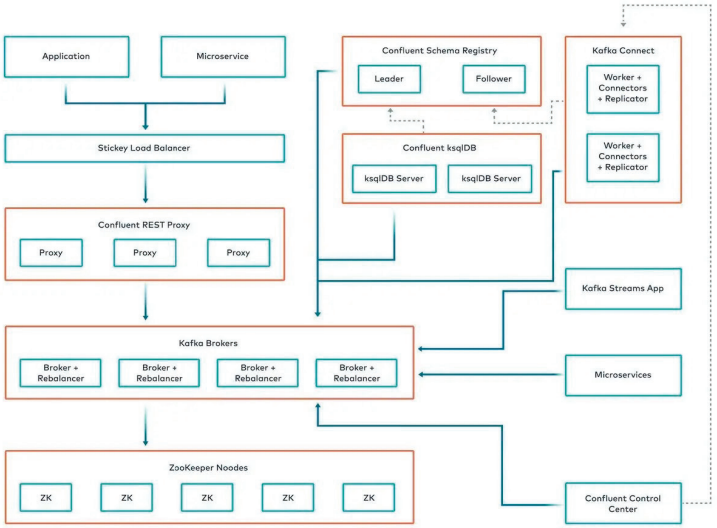

Figure 2. Confluent Kafka large-cluster architecture diagram. IDG

The diagram in Figure 3 shows a three-node Redpanda cluster, where each cluster supports the Kafka API, an HTTP proxy, the Kafka schema registry, and a WebAssembly engine.

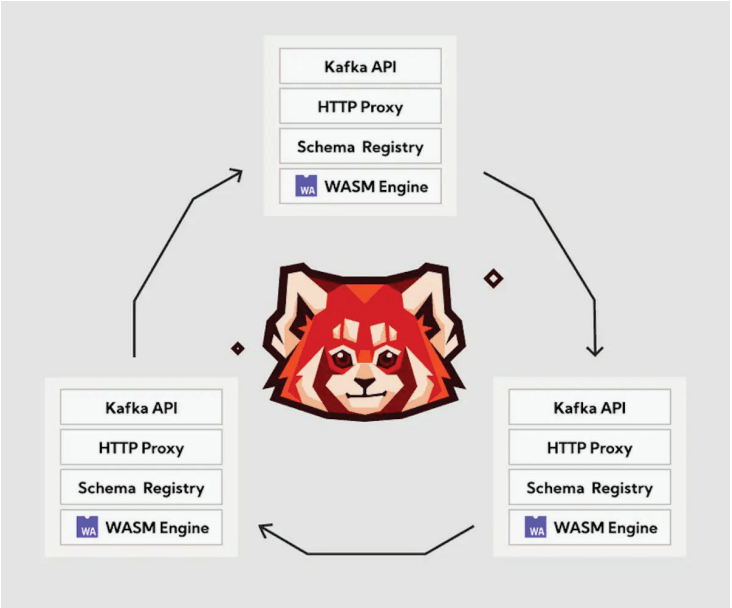

Figure 3. High-level architecture of Redpanda. IDG

Installing and testing Redpanda

You can install Redpanda on Linux and on Docker or Kubernetes containers running on macOS or Windows. You can also run Redpanda in either your own cloud or Redpanda's fully managed cloud. My first instinct, based on the difficulty of installing and managing Kafka clusters, was to use the Redpanda cloud. I was convinced to try installing Redpanda in Docker on one of my Macs; it turned out to be painless once I repaired and updated my old Homebrew installation.

Installation on macOS

As you can see from the log below, this installation wasn't a big deal: Homebrew installed RPK, which controls Redpanda and Docker. You can ignore Homebrew's complaints about my older macOS version: this is not a formula that struggles with macOS High Sierra. (That's the latest version that this iMac can run. Thanks, Apple.)

```
Martins-iMac:~ mheller$ brew install redpanda-data/tap/redpanda
==> Tapping redpanda-data/tap
Cloning into '/usr/local/Homebrew/Library/Taps/redpanda-data/hor
remote: Enumerating objects: 333, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 333 (delta 0), reused 0 (delta 0), pack-reused 32
Receiving objects: 100% (333/333), 37.32 KiB | 1.96 MiB/s, done
Resolving deltas: 100% (160/160), done.
Tapped 1 formula (15 files, 63.9KB).
Warning: You are using macOS 10.13.
We (and Apple) do not provide support for this old version.
You will encounter build failures with some formulae.
Please create pull requests instead of asking for help on Homeb
Twitter or any other official channels. You are responsible for
any issues you experience while you are running this
old version.

==> Downloading https://github.com/vectorizedio/redpanda/releas
==> Downloading from https://objects.githubusercontent.com/gith
######################################################################
==> Installing redpanda from redpanda-data/tap
==> Caveats
Redpanda - The fastest queue in the west!
```

```
This installs RPK which, with Docker, enables the running of a
for testing purposes.

You can start a 3 node cluster locally using the following comma

    rpk container start -n 3

You can then interact with the cluster using commands like the

    rpk topic list

When done, you can stop and delete the cluster with the followi

    rpk container purge

For information on how to setup production evironments, check o
installation guide here: https://vectorized.io/documentation/se
==> Summary
🍺  /usr/local/Cellar/redpanda/21.11.15: 3 files, 23.6MB, built
==> Running `brew cleanup redpanda`...
Disable this behaviour by setting HOMEBREW_NO_INSTALL_CLEANUP.
Hide these hints with HOMEBREW_NO_ENV_HINTS (see `man brew`).
```

Testing

Once installed, the only tricky part of using RPK in this scenario is adding the correct list of node addresses to the commands. Fortunately, the rpk container start command echoes a fleshed-out command line for the rpk cluster info command. You can copy and paste the same broker list for all rpk commands against the same cluster. In a production scenario you'd use DNS or an environment variable, as well as some Redpanda configuration to avoid needing a brokers list.

As you can see, I was successful in exercising both the produce and consume functions. Once I was done, I shut down the whole cluster with rpk container purge (not shown below).

```
Martins-iMac:~ mheller$ rpk container start -n 3
Downloading latest version of Redpanda
Starting cluster
Waiting for the cluster to be ready...
  NODE ID  ADDRESS
  0        127.0.0.1:60042
  1        127.0.0.1:60051
  2        127.0.0.1:60052

Cluster started! You may use rpk to interact with it. E.g:

rpk cluster info --brokers 127.0.0.1:60042,127.0.0.1:60051,127.(

Martins-iMac:~ mheller$ rpk cluster info --brokers 127.0.0.1:600
BROKERS
=======
ID    HOST       PORT
0*    127.0.0.1  60042
1     127.0.0.1  60051
2     127.0.0.1  60052

Martins-iMac:~ mheller$ rpk topic create twitch_chat --brokers 1
TOPIC        STATUS
twitch_chat  OK

Martins-iMac:~ mheller$ rpk topic produce twitch_chat --brokers
this is a test^D
Produced to partition 0 at offset 0 with timestamp 165150792490
How do you stream to Redpanda^D
Produced to partition 0 at offset 1 with timestamp 165150795726
Now is the time for all good men to come to the aid of their pa
Produced to partition 0 at offset 2 with timestamp 165150798657
^C

Martins-iMac:~ mheller$ rpk topic consume twitch_chat --brokers
{
  "topic": "twitch_chat",
  "value": "this is a test",
  "timestamp": 1651507924907,
  "partition": 0,
  "offset": 0
}
{
  "topic": "twitch_chat",
  "value": "How do you stream to Redpanda",
  "timestamp": 1651507957268,
  "partition": 0,
  "offset": 1
}
{
  "topic": "twitch_chat",
  "value": "Now is the time for all good men to come to the aid
  "timestamp": 1651507986574,
  "partition": 0,
  "offset": 2
}
^C

Martins-iMac:~ mheller$ rpk version
v21.11.15 (rev 7325762b6f9e1586efc60ab97b8596f08510b31a)

Martins-iMac:~ mheller$ rpk help
rpk is the Redpanda CLI & toolbox.

Usage:
  rpk [command]

Available Commands:
  acl         Manage ACLs and SASL users.
  cluster     Interact with a Redpanda cluster.
  container   Manage a local container cluster.
  generate    Generate a configuration template for related serv
  group       Describe, list, and delete consumer groups and man
  help        Help about any command
  plugin      List, download, update, and remove rpk plugins.
  redpanda    Interact with a local or remote Redpanda process
  topic       Create, delete, produce to and consume from Redpan
  version     Check the current version.
  wasm        Deploy and remove inline WASM engine scripts.

Flags:
  -h, --help      help for rpk
  -v, --verbose   Enable verbose logging (default: false).

Use "rpk [command] --help" for more information about a command.

Martins-iMac:~ mheller$ rpk topic -h
Create, delete, produce to and consume from Redpanda topics.

Usage:
  rpk topic [command]

Available Commands:
  add-partitions Add partitions to existing topics.
  alter-config   Set, delete, add, and remove key/value configs
  consume        Consume records from topics.
  create         Create topics.
  delete         Delete topics.
  describe       Describe a topic.
  list           List topics, optionally listing specific topics
```

```
  produce        Produce records to a topic.

Flags:
      --brokers strings          Comma-separated list of broker i
      --config string            Redpanda config file, if not set
  -h, --help                     help for topic
      --password string          SASL password to be used for aut
      --sasl-mechanism string    The authentication mechanism to
      --tls-cert string          The certificate to be used for T
      --tls-enabled              Enable TLS for the Kafka API (no
      --tls-key string           The certificate key to be used f
      --tls-truststore string    The truststore to be used for TL
      --user string              SASL user to be used for authent

Global Flags:
  -v, --verbose   Enable verbose logging (default: false).

Use "rpk topic [command] --help" for more information about a co
```

The first three nodes shown in Figure 4 were started by rpk container start.

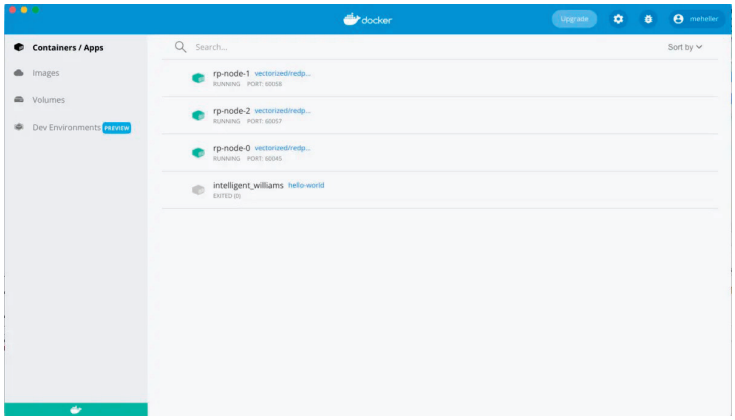

Figure 4. The Docker dashboard shows all the running containers. IDG

Redpanda production deployment options

You can deploy Redpanda for production on bare metal with the Redpanda installation binary, with Terraform and/or Ansible, and on Kubernetes or remote K8s. These options span all hyperscale clouds as well as on-prem deployment. If you're deploying for production you have to watch out for situations where Redpanda runs out of storage. You should also use rpk iotune to test the node hardware and set the proper Redpanda parameters. But if you don't, and you're running on Amazon Web Services or Google Cloud Platform, Redpanda can detect the instance type and start with near optimal settings. You can also tell Redpanda what instance type you're using.

As shown in Figure 5, Redpanda offers both self-managed and cloud deployments. Self-managed community deployments are free. Self-managed enterprise and bring-your-own-cloud deployments are licensed per-core. The fully managed Redpanda cloud service is charged based on usage.

Redpanda Jepsen testing

Redpanda recently collaborated with Kyle Kingsbury on a Jepsen test. The conclusions, as reported by Redpanda, were "Redpanda is a safe system without known consistency problems. The consensus layer is solid. The idempotency and transactional layers had issues that we have already fixed. The only consistency findings we haven't addressed reflect

Deployment Options

	Self-Managed On-prem or any cloud Bare-metal, VM, Kubernetes		Redpanda Cloud aws Google Cloud	
	Community Source Available	Enterprise Support + Features	BYOC Bring Your Own Cloud	FMC Fully Managed Cloud Service
Monitoring/Ops	-	-	✓	✓
Support	-	24x7, 8x5	24x7, 8x5	24x7, 8x5
Tiered Storage	-	✓	✓	✓
Licensing	free	per-core	per-core	usage-based

Redpanda © 2022 Redpanda Data

Figure 5. Redpanda offers both self-managed and cloud deployments. IDG

unusual properties of the Apache Kafka protocol itself, rather than of a particular implementation." Essentially, the Kafka protocol has much looser rules about transactions than those that apply to relational databases, which makes it much faster.

Conclusion

Overall, Redpanda is a Kafka plug-in replacement, written in C++, that has significantly lower latency and higher performance than Apache Kafka. It's also much easier to install and tune.

Redpanda offers both self-managed and cloud deployments. Self-managed community deployments are free, have source available, and are licensed under the BSL, which is almost but not quite open source. Self-managed enterprise and bring-your-own-cloud deployments are licensed per-core. The fully managed Redpanda cloud service is charged based on usage.

Redpanda competes with Apache Kafka, Confluent Kafka, and all the Kafka-based cloud services, including Amazon MSK and Confluent Cloud. If low latency is your highest priority, then choosing Redpanda is a no-brainer. If cost is a big consideration and you need support, then you'll have to contact Redpanda sales to size and cost out a cluster.

AT A GLANCE

Redpanda

★★★★½

Redpanda is a Kafka plug-in replacement written in C++ that has significantly lower latency and higher performance than Kafka. It's also much easier to install and tune.

Pros

- Runs faster and has lower latency than Apache Kafka
- Available free under the source-available BSL license
- Much easier to install and tune than Apache Kafka

Cons

- The BSL license is not quite open source
- You'll need to contact Redpanda sales for pricing on the enterprise and cloud versions

Redpanda

For more information:
www.redpanda.com
hi@redpanda.com
@redpandadata
https://redpandacommunity.slack.com/