Pravega Release Notes – 0.8.0

Table of Contents

Pravega Release Notes – 0.8.0 .... ***1***

Pravega in a nutshell **1**

What's new? **2**

Key-Value Tables **2**

Schema Registry **3**

Performance Improvements **3**

Configuration Names **4**

Other New Features and Updates **4**

Bug Fixes **5**

Ecosystem Updates/Links **5**

Testing **5**

Published artifacts **5**

Pravega in a nutshell

Pravega is a software defined streaming storage system that enables applications to write streams of events or bytes in an append-only manner. It offers several desirable features for stream applications, including the ability to auto-scale streams, provide transaction semantics, and even build replicated state machines via a state synchronizer.

Pravega has three core components:

- **Controller**: manages the lifecycle of streams and transactions
- **Segment store**: implements the functionality in the critical path of reads and writes
- **Client**: library for applications that use Pravega

Pravega is an open-source system released under the Apache 2.0 License. The source code is available in a GitHub repository: http://github.com/pravega/pravega.

What's new?

This release contains exciting new features and significant enhancements over the previous major release line (0.7.x).

This page provides an overview of the major ones. For a full set of changes, please refer to our GitHub release.

Key-Value Tables

In 0.8, we have introduced an entirely new primitive that enables efficient insertions, updates and lookups of arbitrary key-value pairs. Using the same underlying infrastructure as Streams, **Key-Value Tables** provide the same features one would expect of a typical key-value store, plus some more.

Here are a few highlights:

- Distributed Key-Value Store is built on top of append-only Pravega Segments.
- Keys are sharded uniformly across underlying Segments or can be grouped by means of *Key Families*.
- Conditional or unconditional updates and removals.
- Single or multiple entry batch (atomic) updates or removals.
- Single or multiple entry retrieval.
- Listing of keys/entries in bitwise lexicographical order (by key).
- Java Map wrapper provided for easy integration with existing applications.

Key-Value Tables are built on top of Table Segments - the same underlying primitive that Pravega has used to store its own Stream, Segment and Transaction metadata since release 0.5. The Pravega Controller and Segment Store rely on the consistency, durability and atomicity guarantees of Table Segments for all their metadata needs - the same guarantees are also extended to Key-Value Tables as well.

For more details, please refer to PDP-39-Key Value Tables.

As of 0.8, Key-Value Tables are released as **Experimental**. While fully supported, we will continue to make improvements over the upcoming releases and it is possible that we refine some of the APIs too.

Schema Registry

With Pravega 0.8, we are also releasing Pravega **Schema Registry** 0.1. The Schema Registry service is designed to store and manage schemas for unstructured data stored in Pravega streams.

Here are a few highlights:

- The service provides a RESTful interface to store and manage schemas under schema groups.
- Users can choose from different compatibility policies for safely evolving their schema.
- The service has built-in support for popular serialization formats in Avro, Profobuf and JSON schemas. However, users can store schemas of other formats too.
- In v0.1 we have compatibility policies enabled only for Avro. Subsequent releases will support compatibility checks for Protobuf and JSON.
- The service stores and manages encoding information in form of codec information. Codecs could correspond to different compression or encryption used while encoding the serialized data at rest. The service generates unique identifiers for schemas and codec information pairs that users may use to tag their data.

For more details, visit the Schema Registry repository.

As of 0.1, Schema Registry is released as **Experimental**. While fully supported, we will continue to make improvements over the upcoming releases and it is possible that we refine some of the APIs too.

Performance Improvements

Pravega 0.8 contains significant improvements over the previous release. These are the main improvements included in 0.8.

- Issue 4774: (Client) Saves significant compute cycles for a reader by minimizing and optimizing costly calls for managing its internal state.
- Issue 4653: (Segment Store) Control the number of container starts/recoveries performed in parallel.
- Issue 4294: Remove Netty from the client.

Configuration Names

Pravega 0.8 changes the names of almost all configuration items, based on a new uniform configuration naming scheme.

For instance the property `controller.minRebalanceIntervalSeconds` is replaced by `controller.rebalance.interval.seconds.min` and `pravegaservice.storageImplementation` is replaced by `pravegaservice.storage.impl.name`. See this Wiki page for new names of old properties.

Old properties are deprecated but will continue to work for now. So, upon upgrade from previous version your Pravega deployment will continue to work. You can use a combination of new and old properties to Configure Pravega components, although it is highly discouraged. If you specify a property using both new and old names, the old property will take precedence.

Fresh installations must use new properties (even though old properties will work too for now). For upgrade from previous versions, we highly encourage migrating to new properties at the earliest opportunity.

See issue 4712 for more details.

Other New Features and Updates

- Issue 4893: (Controller, Metrics) Adds metrics for created and deleted scopes.
- Issue 4910: (Segment Store, Metrics) Adds two new metrics to report the size distribution of appends and reads.
- Issue 4680: (Client) Add a new truncation API to ByteStreamWriter.
- Issue 4791: (Client, Controller) Add a new controller API to fetch segments for a given epoch
- Issue 4845: (Client, Usability) StreamInfo obtained via io.pravega.client.admin.impl.StreamManager indicates if the Stream is sealed.
- Issue 4846: (Library upgrades) Upgrade Netty to 4.1.50Final.
- Issue 4883: (Library upgrades) Upgrade gRPC to 1.24.2.
- Issue 4531: (Library upgrades) Upgrade Bookeeper client to 4.9.2.

Bug Fixes

- Issue 4631: (Controller, Metrics) Report failed transaction commit metric only in case of non-retryable failures.
- Issue 4847: (Client) Improve sealed stream handling of EventStreamWriter and ByteStreamWriter.

Ecosystem Updates/Links

- Connectors
  - Pravega Flink Connectors
  - Pravega Hadoop Connectors
  - Pravega Boomi Connector
- Kubernetes Operators
  - Pravega Operator: v0.5.1
  - Pravega Bookkeeper Operator: v0.1.2
  - Pravega Zookeeper Operator: v0.2.8 and v0.2.9-rc0

Testing

We have tested Pravega extensively:

- We have currently over 2,000 Junit test cases that we exercise on a regular basis.
- We have currently 30 system test cases that we exercise on a regular basis. Our system tests currently run on both Kubernetes and Docker Swarm.
- We have executed longevity workloads on different infrastructure: Google Cloud, Amazon Web Services, and on-premises PKS.
- We have performed a number functional and non-functional tests, such as killing pods.

Published artifacts

GitHub:

https://github.com/pravega/pravega/releases/tag/v0.8.0

Docker image:

https://hub.docker.com/r/pravega/pravega

Maven Central:
http://central.maven.org/maven2/io/pravega/