

S4Proposal

S4 Proposal

Abstract

S4 (Simple Scalable Streaming System) is a general-purpose, distributed, scalable, partially fault-tolerant, pluggable platform that allows programmers to easily develop applications for processing continuous, unbounded streams of data.

Proposal

S4 is a software platform written in Java. Clients that send and receive events can be written in any programming language. S4 also includes a collection of modules called Processing Elements (or PEs for short) that implement basic functionality and can be used by application developers. In S4, keyed data events are routed with affinity to Processing Elements (PEs), which consume the events and do one or both of the following: (1) *emit* one or more events which may be consumed by other PEs, (2) *publish* results. The architecture resembles the Actors model, providing semantics of encapsulation and location transparency, thus allowing applications to be massively concurrent while exposing a simple programming interface to application developers.

To drive adoption and increase the number of contributors to the project, we may need to prioritize the focus based on feedback from the community. We believe that one of the top priorities and driving design principle for the S4 project is to provide a simple API that hides most of the complexity associated with distributed systems and concurrency. The project grew out of the need to provide a flexible platform for application developers and scientists that can be used for quick experimentation and production.

S4 differs from existing Apache projects in a number of fundamental ways. Flume is an Incubator project that focuses on log processing, performing lightweight processing in a distributed fashion and accumulating log data in a centralized repository for batch processing. S4 instead performs all stream processing in a distributed fashion and enables applications to form arbitrary graphs to process streams of events. We see Flume as a complementary project. We also expect S4 to complement Hadoop processing and in some cases to supersede it. Kafka is another Incubator project that focuses on processing large amounts of stream data. The design of Kafka, however, follows the pub-sub paradigm, which focuses on delivering messages containing arbitrary data from source processes (publishers) to consumer processes (subscribers). Compared to S4, Kafka is an intermediate step between data generation and processing, while S4 is itself a platform for processing streams of events.

S4 overall addresses a need of existing applications to process streams of events beyond moving data to a centralized repository for batch processing. It complements the features of existing Apache projects, such as Hadoop, Flume, and Kafka, by providing a flexible platform for distributed event processing.

Background

S4 was initially developed at Yahoo! Labs starting in 2008 to process user feedback in the context of search advertising. The project was licensed under the Apache License version 2.0 in October 2010. The project documentation is currently available at <http://s4.io>.

Rationale

Stream computing has been growing steadily over the last 20 years. However, recently there has been an explosion in real-time data sources including the Web, sensor networks, financial securities analysis and trading, traffic monitoring, natural language processing of news and social data, and much more.

As Hadoop evolved as a standard open source solution for batch processing of massive data sets, there is no equivalent community supported open source platform for processing data streams in real-time. While various research projects have evolved into proprietary commercial products, S4 has the potential to fill the gap. Many projects that require a scalable stream processing architecture currently use Hadoop by segmenting the input stream into data batches. This solution is not efficient, results in high latency, and introduces unnecessary complexity.

The S4 design is primarily driven by large scale applications for data mining and machine learning in a production environment. We think that the S4 design is surprisingly flexible and lends itself to run in large clusters built with commodity hardware.

S4 enables application programmers to focus more on the application and less on the infrastructure. S4 also provides a consistent graph oriented programming model that, if widely adopted, will facilitate sharing of basic component across developers.

Initial Goals

The basic S4 infrastructure is complete and can be used in real-world applications. However, many additional components need to be developed and improved. Some areas we hope to focus on in Apache:

- Add a reliable communication protocol option to the communication layer for low bandwidth control messages that require guaranteed delivery.
- Higher-performance serialization and inter-node communication.
- Functionality to save the state of PEs at runtime transparently and restore it at startup.
- Intelligent load shedding strategies.
- Dynamic load balancing to make it possible to add and remove nodes from the cluster without data loss.
- Dynamic application loading and unloading.
- Migration to a pure object-oriented design that takes advantage of Java static typing using Generics in the framework code. (Keep it simple for the application developer.)
- Eliminate string identifiers and XML configuration.
- Adopt JSR 330 (Dependency Injection for Java).
- Add real-time query support.
- Add a cluster management system.

Clearly this is a long list but sets the high level roadmap for the project.

Current Status

The project has been under development at Yahoo! since late 2008, and it was open sourced in October 2010. Since then we have received patches from developers, started a discussion forum, and improved the documentation.

Meritocracy

The S4 project was initially developed at Yahoo! Labs, a research-oriented organization that values original ideas and individual contributions. The design evolved in a bottom up fashion, where decisions were driven by the application and the long-term viability and flexibility of the platform. Once the project became open-source it continued to be managed by those who were actively doing the work.

Community

S4 is currently in use internally at Yahoo!, and since it was released as an open source project it has received positive feedback and contributions from developers.

Core Developers

S4 developers span a few companies and work on a voluntary basis. We expect to have developers from other organizations joining the team in the next few months, especially if S4 joins the Apache Incubator project. Being an Apache Incubator project is likely to attract the attention of more talented developers.

One interesting aspect of the current group of developers is the diverse background:

- Kishore Gopalakrishna was the main developer of the communication layer and the integration with Zookeeper. He has been an active contributor to Hadoop.
- Flavio Junqueira has a background in distributed computing. He is a committer of [ZooKeeper](#), a [ZooKeeper](#) PMC member, and a committer of [BookKeeper](#);
- Matthieu Morel has extensive background in distributed systems, he likes theory and loves to implement things. He has been the main designer and implementor of S4 checkpointing.* Anish Nair has been the project's main customer. With his background on natural language processing and algorithms he developed the applications that drove the S4 design including processing of social feeds and real-time recommendation engines.
- Leo Neumeyer has a background in signal processing and statistical modeling but has been advocating clean simple software design throughout his career. At Yahoo! he conceived and championed the S4 project as a solution to improve monetization in search advertising.
- Bruce Robbins has been the main S4 developer, taking the concept from idea to releases. Bruce engineering experience ranges from programming Mainframe computers to assembly code.

Alignment

S4 brings stream processing capabilities that complement Hadoop's batch processing capabilities.

Known Risks

Orphaned Products

S4 has been used in production at Yahoo! and is being evaluated by other organizations. The developers have continued to support the project on their own time. We believe that adoption will increase significantly as more tools and documentation become available. As the project evolves, we may see new ideas that we may want to adopt or, if it makes sense and is practical, we may want to merge two or more open source projects. We believe that there is a clear need to have a well supported open source stream processing platform and therefore, there is low risk of the project becoming orphan. However, we are open to combining projects in order to have fewer projects with a more active community. Ultimately, this will be decided by the design ideas, the implementation quality, and the adoption.

Inexperience with Open Source

The S4 code was open sourced by Yahoo! under Apache 2.0 license. One committer of the S4 project, Flavio Junqueira, is intimately familiar with the Apache model for open-source development and is experienced with working with new contributors. Flavio is both a committer a PMC member for [ZooKeeper](#). The other developers have had experience as contributors in other open-source projects. Most of the original S4 developers continue to be committers.

Homogeneous Developers

The initial set of committers for S4 represent four different companies: A9, LinkedIn, Quantbench, and Yahoo!. This set is diverse enough for a starting project.

Reliance on Salaried Developers

Some committers are contributing as part of their jobs, but as we move to a more diverse set of developers we expect a good mix of salaried and volunteer time.

Relationships with Other Apache Projects

S4 relies on the following Apache projects:

- BCEL (bytecode generation library)
- commons cli (command line interface)
- commons logging (needed by some other dependency)
- log4j
- commons jexl (expression processing)
- zookeeper
- Maven and its usual plug-ins (build time only)

Compared to existing projects, S4 complements existing functionality in a few ways summarized below:

- Flume: S4 processes streams in a distributed fashion and enables applications to form arbitrary graphs of processing elements. Flume focuses on accumulating streams of logs in a centralized repository for batch processing;
- Kafka: Kafka is a pub/sub messaging layer that interposes generation of events and processing, while S4 itself forwards events and processes them in a stream fashion.
- Hadoop: Hadoop focuses on batch processing of large data sets, while S4 is a platform for stream processing of events. We would like to implement extensions that enable processing in both platforms with the same code.

An Excessive Fascination with the Apache Brand

The project has already received a significant amount of attention and so far has been associated with Yahoo!. We would like, however, to foster the development of a community around S4 that evolves independently of the interests of a single company. Given the reliance of S4 on some Apache projects and the principles promoted by the foundation, we find it a suitable home for the project.

Documentation

- S4 Website: <http://s4.io>
- S4 documentation: <http://docs.s4.io/>
- S4 Forum: <http://groups.google.com/group/s4-project/topics>
- S4 Mailing list (with archives): <http://groups.google.com/group/s4-project>

Source and Intellectual Property Submission Plan

The S4 source code is already licensed under Apache Software License 2.0. The source code is available at <https://github.com/s4>

External Dependencies

- asm (3-clause BSD license)
- json (json.org's own license <http://www.crockford.com/JSON/license.html> which is acceptable as per Apache FAQ: <http://www.apache.org/legal/resolved.html#json>)
- kryo (4-clause BSD license)
- spring framework (Apache license - v 2)
- codehaus jackson (Apache license)
- junit (Common Public License - v 1.0)

Cryptography

None

Required Resources

Mailing lists

- s4-dev
- s4-user
- s4-private (with moderated subscriptions)
- s4-commit

Subversion Directory

<https://svn.apache.org/repos/asf/incubator/s4>

Issue Tracking

JIRA S4 (S4)

Initial Committers

- Kishore Gopalakrishna (kg at s4 dot io)
- Flavio Junqueira (fpj at s4 dot io)
- Matthieu Morel (mm at s4 dot io)
- Anish Nair (an at s4 dot io)
- Leo Neumeyer (leo at s4 dot io)
- Bruce Robbins (br at s4 dot io)

Affiliations

- Kishore Gopalakrishna, LinkedIn
- Flavio Junqueira, Yahoo!
- Matthieu Morel, Yahoo!
- Anish Nair, A9
- Leo Neumeyer, Quantbench
- Bruce Robbins, Yahoo!

Sponsors

Champion

- Patrick Hunt

Nominated Mentors

- Patrick Hunt
- Owen O'Malley
- Arun Murthy

Sponsoring Entity

- Apache Incubator PMC