ToriiProposal

Torii

Abstract

Torii provides applications with a mechanism to interactively and remotely access Apache Spark.

Proposal

Torii enables interactive applications to access Apache Spark clusters. More specifically:

- Applications can send code-snippets and libraries for execution by Spark
- Applications can be deployed separately from Spark clusters and communicate with the Torii using the provided Torii client
- Execution results and streaming data can be sent back to calling applications
- Applications no longer have to be network connected to the workers on a Spark cluster because the Torii acts as each application's proxy
- Work has started on enabling Torii to support languages in addition to Scala, namely Python (with PySpark), R (with SparkR), and SQL (with SparkSQL)

Background & Rationale

Apache Spark provides applications with a fast and general purpose distributed computing engine that supports static and streaming data, tabular and graph representations of data, and an extensive library of machine learning libraries. Consequently, a wide variety of applications will be written for Spark and there will be interactive applications that require relatively frequent function evaluations, and batch-oriented applications that require one-shot or only occasional evaluation.

Apache Spark provides two mechanisms for applications to connect with Spark. The primary mechanism launches applications on Spark clusters using spark-submit (http://spark.apache.org/docs/latest/submitting-applications.html); this requires developers to bundle their application code plus any dependencies into JAR files, and then submit them to Spark. A second mechanism is an ODBC/JDBC API (http://spark.apache.org/docs/latest/sql-programming-guide.html#distributed-sql-engine) which enables applications to issue SQL queries against SparkSQL.

Our experience when developing interactive applications, such as analytic applications integrated with Notebooks, to run against Spark was that the spark-submit mechanism was overly cumbersome and slow (requiring JAR creation and forking processes to run spark-submit), and the SQL interface was too limiting and did not offer easy access to components other than SparkSQL, such as streaming. The most promising mechanism provided by Apache Spark was the command-line shell (http://spark.apache.org/docs/latest/programming-guide.html#using-the-shell) which enabled us to execute code snippets and dynamically control the tasks submitted to a Spark cluster. Spark does not provide the command-line shell as a consumable service but it provided us with the starting point from which we developed Torii.

Current Status

Torii was first developed by a small team working on an internal-IBM Spark-related project in July 2014. In recognition of its likely general utility to Spark users and developers, in November 2014 the Torii project was moved to GitHub and made available under the Apache License V2.

Meritocracy

The current developers are familiar with the meritocratic open source development process at Apache. As the project has gathered interest at GitHub the developers have actively started a process to invite additional developers into the project, and we have at least one new developer who is ready to contribute code to the project.

Community

We started building a community around Torii project when we moved it to GitHub about one year ago. Since then we have grown to about 70 people, and there are regular requests and suggestions from the community. We believe that providing Apache Spark application developers with a general-purpose and interactive API holds a lot of community potential, especially considering possible tie-in's with Notebooks and data science community.

Core Developers

The core developers of the project are currently all from IBM, from the IBM Emerging Technology team and from IBM's recently formed Spark Technology Center.

Alignment

Apache, as the home of Apache Spark, is the most natural home for the Torii project because it was designed to work with Apache Spark and to provide capabilities for interactive applications and data science tools not provided by Spark itself.

The Torii also has an affinity with Jupyter (jupyter.org) because it uses the Jupyter protocol for communications, and so Jupyter Notebooks can directly use the Torii as a kernel for communicating with Apache Spark. However, we believe that the Torii provides a general-purpose mechanism enabling a wider variety of applications than just Notebooks to access Spark, and so the Torii's greatest affinity is with Apache and Apache Spark.

Known Risks

Orphaned products

We believe the Torii project has a low-risk of abandonment due to interest in its continuing existence from several parties. More specifically, the Torii provides a capability that is not provided by Apache Spark today but it enables a wider range of applications to leverage Spark. For example, IBM uses (and is considering) the Torii in several offerings including its IBM Analytics for Apache Spark product in the Bluemix Cloud. There are also a couple of other commercial users who are using or considering its use in their offerings. Furthermore, Jupyter Notebooks are used by data scientists and Spark is gaining popularity as an analytic engine for them. Jupyter Notebooks are very easily enabled with the Torii and so there is another constituency for it.

Inexperience with Open Source

The Torii project has been running as an open-source project (albeit with only IBM committers) for the past several months. The project has an active issue tracker and due to the interest indicated by the nature and volume of requests and comments, the team has publicly stated it is beginning to build a process so they can accept third-party contributions to the project.

Relationships with Other Apache Products

The Torii has a clear affinity with the Apache Spark project because it is designed to provide capabilities for interactive applications and data science tools not provided by Spark itself. The Torii can be a back-end for the Zeppelin project currently incubating at Apache. There is interest from the Torii community to develop this capability and an experimental branch has been started.

Homogeneous Developers

The current group of developers working on Torii are all from IBM although the group is in the process of expanding its membership to include members of the GitHub community who are not from IBM and who have been active in the Torii community in GutHub.

Reliance on Salaried Developers

The initial committers are full-time employees at IBM although not all work on the project full-time.

Excessive Fascination with the Apache Brand

We believe the Torii benefits Apache Spark application developers, and we are interested in an Apache Torii project to benefit these developers by engaging a larger community, facilitating closer ties with the existing Spark project, and yes, gaining more visibility for the Torii as a solution.

Documentation

Comprehensive documentation including "Getting Started", API specifications and a Roadmap are available from the GitHub project, see https://github.com/ibm-et/Torii/wiki.

Initial Source

The source code resides at https://github.com/ibm-et/Torii.

External Dependencies

The Torii depends upon a number of Apache projects:

- Spark
- Hadoop
- Ivy
- Commons

The Torii also depends upon a number of other open source projects:

- ZeroMQ (LGPL with Static Linking Exception, http://zeromq.org/area:licensing)
- Akka (MIT)
- JOpt Simple (MIT)
- Spring Framework Core (Apache v2)
- Play (Apache v2)
- SLF4J (MIT)
- Scala
- Scalatest (Apache v2)
- Scalactic (Apache v2)
- Mockito (MIT)

Required Resources

Mailing lists

- private@torii.incubator.apache.org (with moderated subscriptions)
- commits@torii.incubator.apache.org
- dev@torii.incubator.apache.org

Git Repository

- https://git-wip-us.apache.org/repos/asf/incubator-torii.git

Issue Tracking

- A JIRA issue tracker: https://issues.apache.org/jira/browse/TORII

Initial Committers

- Leugim Bustelo (lbustelo AT us DOT ibm DOT com)
- Jakob Odersky (odersky AT us DOT ibm DOT com)
- Luciano Resende (lresende AT apache DOT org)
- Robert Senkbeil (rcsenkbe AT us DOT ibm DOT com)
- Corey Stubbs (cstubbs AT us DOT ibm DOT com)
- Miao Wang (wangmiao AT us DOT ibm DOT com)
- Sean Welleck (swelleck AT us DOT ibm DOT com)

Affiliations

All of the initial committers are employed by IBM.

Sponsors

Champion

- Sam Ruby (rubys AT apache DOT org)

Nominated Mentors

- Luciano Resende (lresende AT apache DOT org)
- Reynold Xin (rxin AT apache DOT org)
- Hitesh Shah (hitesh AT apache DOT org)
- Julien Le Dem (julien AT apache DOT org)

Sponsoring Entity

We would like to propose the Apache Incubator to sponsor this project.