DRATProposal

Apache DRAT Proposal

Abstract

Apache Distributed Release Audit Tool (DRAT) is a distributed, parallelized (Map Reduce) wrapper around Apache RAT™ to allow it to complete on large code repositories of multiple file types where Apache™ RAT hangs forever.

Proposal

Apache DRAT is a distributed, parallelized (Map Reduce) wrapper around Apache RAT™ (Release Audit Tool). RAT is used to check for proper licensing in software projects. However, RAT takes a prohibitively long time to analyze large repositories of code, since it can only run on one JVM. Furthermore, RAT isn't customizable by file type or file size and provides no incremental output. This wrapper dramatically speeds up the process by leveraging Apache OODT™ to parallelize and workflow the following components:

- Apache Solr™ based exploration of a CM repository (e.g., Git, SVN, etc.) and classification of that repository based on MIME type using Apache Tika™.
- A MIME partitioner that uses Apache Tika™ to automatically deduce and classify by file type and then partition Apache™ RAT jobs based on sets of 100 files per type (configurable) – the M/R "partitioner"
- A throttle wrapper for RAT to MIME targeted Apache RAT™. – the M/R "mapper"
- A reducer to "combine" the produced RAT logs together into a global RAT report that can be used for stats generation. – the M/R "reducer"

Background and Rationale

As a part of the Apache Software Foundation (ASF) project, Apache Creadur™, a Release Audit Tool (RAT) was developed especially in response to demand from the Apache Software Foundation and its hundreds of projects to provide a capability for release auditing that could be integrated into projects. The primary function of the RAT is automated code auditing and open-source license analysis focusing on headers. RAT is a natural language processing tool written in Java to easily run on any platform and to audit code from many source languages (e.g., C, C++, Java, Python, etc.). RAT can also be used to add license headers to codes that are not licensed.

In the summer of 2013, our team ran Apache RAT on source code produced from the Defense Advanced Research Projects Agency (DARPA) XDATA national initiative whose inception coincided with the 2012 U.S. Presidential Initiative in Big Data. XDATA brought together 24 performers across academia, private industry and the government to construct analytics, visualizations, and open source software mash-ups that were transitioned into government projects and to the defense sector. XDATA produced a large Git repository consisting of ~50,000 files and 10s of millions of lines of code. DARPA XDATA was launched to build a useful infrastructure for many government agencies and ultimately is an effort to avoid the traditional government-contractor software pipeline in which additional contracts are required to reuse and to unlock software previously funded by the government in other programs. All XDATA software is open source and is ingested into DARPA's Open Catalog that points to outputs of the program including its source code and metrics on the repository. Because of this, one of core products of XDATA is the internal Git repository. Since XDATA brought together open source software across multiple performers, having an understanding of the licenses that the source codes used, and their compatibilities and differences was extremely important and since there repository was so large, our strategy was to develop an automated process using Apache RAT. We ran RAT on 24-core, 48 GB RAM Linux machine at the National Aeronautics and Space Administration (NASA)'s Jet Propulsion Laboratory (JPL) to produce a license evaluation of the XDATA Git repository and to provide recommendations on how the open source software products can be combined to adhere to the XDATA open source policy encouraging permissive licenses. Against our expectations, however, RAT failed to successfully and quickly audit XDATA's large Git repository. Moreover, RAT provided no incremental output, resulting in solely a final report when a task was completed. RAT's crawler did not automatically discern between binary file types and another file types. It seemed that RAT performed better by collecting similar sets of files together (e.g., all Javascript, all C++, all Java) and then running RAT jobs individually based on file types on smaller increments of files (e.g., 100 Java files at a time, etc). The lessons learned navigating these issues have motivated to create "DRAT", which stands for "Distributed Release Audit Tool". DRAT directly overcomes RAT's limitations and brings code auditing and open source license analysis into the realm of Big Data using scalable open source Apache technologies. DRAT is already being applied and transitioned into the government agencies. DRAT currently exists at Github under the ALv2 under Chris Mattmann's GitHub account. Chris Mattmann was the PI of DARPA XDATA at JPL.

Current Status

DRAT is feature-complete, and is about to release a v 1.0. Further development on DRAT will occur at a minimum at JPL (we are going to use it on our huge internal GitHub enterprise repository, and offer it as a service to the lab). In addition, DRAT currently is used in the NSF EarthCube program to assess repositories, and there are a handful of users of the system. We are a small community, but there is already great interest in the system.

Meritocracy

8/10 Current developers for the project are all ASF members, experienced with ASF processes and procedures. We know how to grow an Apache community and to develop a meritocratic free and open source project.

Community

The current DRAT community is primarily a handful of JPL'ers, along with former JPL'ers and USC students who have worked on the project. Those students and former JPL'ers now are employed at places like Apple, Google, and various startups and all expect to continue to the project. Beyond these contributors and community we look to take advantage of Google Summer of Code, along with continue to build community in Mattmann's graduate courses at USC, and at JPL and the NSF and Earth Science Information Partners (ESIP) Federation community. In addition we have already generated some interest and contributions from the Wicket community. We expect OODT people to be interested and to contribute to DRAT since it is a canonical use of Apache OODT. In addition, Tika, Solr and Lucene folks will likely be interested to contribute, and of course last but not least, folks working on RAT /Creadur we expect to be contributors to the project.

Core Developers

Chris Mattmann remains the core developer of DRAT, along with more recently a set of graduate students and JPL'ers who have since moved on to Apple, Google and private enterprise. Those developers are still engaged with the project. We also get contributions from students and JPL'ers using the effort, and we are looking to grow the core set of developers by being an Apache project.

Alignment

The project is already operating using Apache meritocracy principles, and our philosophy for community and code aligns well with the ASF.

Known Risks

Orphaned products

JPL making a commitment to run DRAT on our internal code repos, and the NSF and DARPA have already invested handsomely in DRAT as a service for their code repositories. There is funding going forward for the next few years to work on DRAT. Having DRAT at Apache will hopefully embolden the project to expand into other communities, e.g., Wicket (perhaps to be a canonical example of a Wicket app?), or Creadur/RAT (perhaps via a Maven plugin). We also expect to have DRAT interactive reports become the norm at the ASF rather than RAT as it is currently mostly used which is based on individual projects. There is little chance for orphaning, but if we reach that point we will follow the Apache process around Attic/etc.

Inexperience with Open Source

All proposed PMC members are seasoned open source veterans at the ASF or in the broader open source community.

Relationships with Other Apache Products

- RAT - the core license analysis toolkit that inspired DRAT. We use RAT as a "map reduce mapper".
- OODT - the data management and workflow framework to perform license analysis, record intermediate Rat logs, and catalog and archive source code files, and to run Tika on them. OODT is the "glue" for the project.
- Tika - performs extractions and MIME detections that allow DRAT to "split/partition" the code base in the map reduce style. Tika is responsible for classifying source code files during repository analysis.
- Lucene / Solr - used to create an index of metadata properties about the source code repo and to query those properties.
- Wicket - our core "operational cockpit" for DRAT, Proteus, is built on Wicket.

Homogeneous Developers

Our developers are experienced working together, and have done so across various Apache projects, OODT, Tika, Solr, etc., for years and years. We have like mind and common interest in DRAT.

Reliance on Salaried Developers

Right now DRAT relies on our current salaried developers to push forward on development. We are looking to grow its community and to not only rely on such contributions in part by bringing the project to the ASF.

A Excessive Fascination with the Apache Brand

We love Apache. We used lots of Apache great tools to put DRAT together. We know the value of being at the ASF and DRAT is a perfect example of a project that would thrive within the ASF.

Documentation

Documentation including code, a wiki, and publications surrounding DRAT can be found at http://github.com/chrismattmann/drat/.

Initial Source

Documentation including code, a wiki, and publications surrounding DRAT can be found at http://github.com/chrismattmann/drat/

External Dependencies

DRAT depends on a number of Apache projects:

- OODT - ALv2
- Lucene - ALv2
- RAT - ALv2
- Solr - ALv2
- Tika - ALv2
- Wicket - ALv2
- Data Driven Documents (D3) - BSD 3 clause
- AngularJS - MIT license

and their associated dependencies.

As all dependencies are managed using Apache Maven, none of the external libraries need to be packaged in a source distribution.

Required Resources

Developer and user mailing lists

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

A gitbox repository at:

https://github.com/apache/drat.git

Issue tracking

We will use the GitHub issue tracker.

Initial Committers

The following is a list of the planned initial Apache committers.

- Chris Mattman
- Tyler Palsulich
- Brian Fox
- Paul Ramirez
- Lewis John McGibbney
- Karanjeet Singh
- Steven Francus
- Michael Joyce
- Tom Barber
- Wayne Burke

Affiliations

NASA JPL

- Chris Mattmann
- Paul Ramirez
- Lewis John McGibbney
- Michael Joyce
- Tom Barber
- Wayne Burke

Sonatype

- Brian Fox

Apple

- Karanjeet Singh

Google

- Tyler Palsulich

Chronaly

- Steven Francus

Podling Name Search

- PODLINGNAMESEARCH-130

Sponsors

Champion

Chris Mattmann (NASA/JPL)

Nominated Mentors

- Chris Mattmann
- Paul Ramirez
- Lewis John McGibbney
- Brian Fox

[others]

Sponsoring Entity

The Apache Board (pTLP) or...the Apache Incubator.