TVM & THE APACHE SOFTWARE FOUNDATION
MARKUS WEIMER
MEMBER, APACHE SOFTWARE FOUNDATION
ARCHITECT, MICROSOFT ML PLATFORM

Why I am here

Pays me to be here
AGENDA

- Intro about the ASF
- The Apache Way
- Community choices
- Next steps
Open.
The Apache Software Foundation provides support for the Apache Community of open-source software projects, which provide software products for the public good.

Innovation.
The Apache projects are defined by collaborative consensus based processes, an open, pragmatic software license and a desire to create high quality software that leads the way in its field.

Community.
We consider ourselves not simply a group of projects sharing a server, but rather a community of developers and users.

Source: https://www.apache.org/
HISTORY: APACHE HTTPD

1992: Admins trading patches for NCSA webserver

1995: First release of httpd

1999: Non-profit, membership based corporation
      Protects contributor and intellectual property

2016: Most widely deployed webserver for 19 years [1]

WHAT IS THE ASF? -- A NON-PROFIT COOPERATION

- **Board**: Appoint
- **Members**: Elect
- **Officers (PMC Chairs)**
- **PMC**
- **Contributors**
- **Committers**

PROJECT LIFECYCLE

Incubation
- Curriculum to learn responsibilities/procedures necessary to protect foundation:
  Diverse community, accept new contributors, develop “in the open”
- Searching for a purpose and a role in the ecosystem.

Mature
- More users/legacy/visibility
- Massive ego; “conquered the web”, etc.
- Propose new abstraction/interface, rather than deep changes to core function

Attic
- Project retirement stage
- Code still available, may be “complete”, but community has moved on
AGENDA

Intro about the ASF

The Apache Way

Community choices

Next steps
CORE PRINCIPLES

- Collaborative software development
- Respectful, honest, technical-based interaction
- Consistently high-quality software
- Commercial-friendly standard license
- Faithful implementation of standards
- Security as a mandatory feature

Community over code
MERITOCRACY AND ROLES

- **User**
  - Use the software
  - Ask questions

- **Developer**
  - Give answers
  - Add code, graphics, testing, ...

- **Committer**
  - @apache.org
  - Tactical decision making

- **PMC Member**
  - Project stewardship

- **PMC Chair**
  - Board Liaison
  - VP @ ASF

- **Member**
  - ASF owners

- **PMC Member**
  - Project stewardship

- **PMC Chair**
  - Board Liaison
  - VP @ ASF

- **Member**
  - ASF owners
Be easy to work with

Don’t be a sycophant
Disagree respectfully
Focus on the technical details

Implied, occasionally explicit aversion to “leaders”

Roles are responsibilities; authority necessary to perform function
e.g., releases are lock-free; a struggling RM can (should) be preempted
Leads rotate, authority partitioned to subdomains
**Explicit consensus**

- Releases, new committers / PMCs, rules changes, ...
- [DISCUSS], [VOTE], [VOTE][RESULT] email threads

**Implicit consensus**

- Code changes, documentation, ...
- Given by absence of veto

**Veto powers**

- Every committer has veto powers against code changes based on technical reasons
- Pro tip: never argue whether the reasons are technical

---

**CONSENSUS DRIVEN DEVELOPMENT**
Voting should be relatively rare. The goal is that the project proceeds by consensus, and its members negotiate outcomes with one another. The votes (+1, -1, …) are sometimes used in discussions to express opinions outside of a [VOTE] thread.
As public as possible:

- user@ for usage questions
- dev@ for development discussion
- private@ for personal and legal matters

Easy to search and archive

Plain text, Markdown where needed
Quoting with `>`
“If it didn’t happen on the mailing list, it didn’t happen”
LESSON: THE APACHE PROCESS LEADS TO BETTER SOFTWARE

“If it isn’t on the mailing list, it didn’t happen”

- **The bad:**
  - Higher latency per issue: Discussing things over email takes longer
  - This can be trained and gets get better over time.

- **The good:**
  - Full visibility, no need for meetings to “bring everybody on the same page”
  - Documents the decision process, not just the outcome
  - Every developer (even the shy ones) have equal influence.

- **The phenomenal:**
  - Quality goes up (code as well as discussion becomes part of your CV)
  - Throughput goes up (less blockage, uncertainty)
AGENDA

- Intro about the ASF
- The Apache Way
- Community choices
- Next steps
As communities form, they make their own rules
In practice, people copy / merge the rules from past projects
REEF copied from Spark and Hadoop, which in turn spawned from Lucene

REEF
- Looks towards Spark and Hadoop

Spark
- Community overlap with Hadoop

Hadoop
- Started out of Lucene

Lucene
CTR VS. RTC

Commit then review (CTR)

- Every committer does the code changes they see fit.
- Other committers review and undo changes.
- Stabilization happens during release.
- Messy commit log

Review then commit (RTC)

- Every change is reviewed by another committer.
- Every change is merged by another committer.
- Always releasable software
- Extremely clean commit log
- Used by every Big Data project in Apache

C:\s\reef [master]> git log --oneline

b5c807a [REEF-1373] Convert IClock to an interface
6b67524 [REEF-1420] Dispose IGroupCommClient/Network Service from IMRU tasks
af8b908 [REEF-1369] Remove obsolete RuntimeClock.RegisterObserver
f2b9b84 [REEF-1421] Transport Client inner thread is not canceled when the object is disposed
1a2f120 [REEF-1410] Validate Task constructor failure => FailedTask Event
dbd628a [REEF-1414] Condition to EvaluatorExitLogger is inverted for RuntimeStop
414d4b4 [REEF-1392] Adding IObserver<ICloseEvent> for IMRU tasks
b564736 [REEF-1345] Define IMRU task exceptions
d5a671b [REEF-1403] Deadlock between ContextRuntime.StartTask and HeartBeatManager.OnNext(Alarm)
f5ae659 [REEF-1306] Remove OnDriverReconnect from parameter in DriverConfiguration
bf5caab [REEF-1396] Fix testFailureRestart to validate that the restarted Evaluators are received
4c0207e [REEF-1409] Upgrade HDP2.4 docker image to use 2.4.2 repository
947b18e [REEF-1398] Update version to 0.16.0-SNAPSHOT
b495935 [REEF-1400] Update update_website.py script to include pom.xml file
e03fa36 [REEF-1401] Fix Driver Restart to correctly use the right folder
JIRAS, CODE REVIEWS AND COMMITS

- Design discussions happen on JIRA (yay!)
  - People are comfortable and encourage many open JIRAs
  - Example [MapReduce-279] (YARN) was open for 3.5 years.
  - Often, the committer closes issue (2nd best option)

- Code reviews happen on GitHub

- Commit messages link back to both.
  - Committer who does the merge also rebases to current `master`
  - Commits are squashed to create a linear commit history
  - “True history” is preserved on GitHub
[REEF-1410] Validate Task constructor failure => FailedTask Event

This addressed the issue by
  * Fixing heartbeat failure if task fails to start.
  * Adding a test to verify Task failure message.

JIRA:
[REEF-1410](https://issues.apache.org/jira/browse/REEF-1410)

Pull Request:
This closes #1019
RELEASE MANAGER CREATES A BRANCH, BEGS FOR HELP

MAJORITY OF PMC VOTES, BECOMES THE NEXT RELEASE (FOUNDATION RULE)

VERSION ASSIGNED DURING THE VOTE
AGENDA

Intro about the ASF

The Apache Way

Community choices

Next steps
NEXT STEPS

WRITE A PROPOSAL

FORM AN INITIAL PMC AND COMMITTER GROUP

SEND THE PROPOSAL TO THE APACHE INCUBATOR
THANKS FOR YOUR TIME!

- weimer@apache.org
- Markus.Weimer@Microsoft.com
- @markusweimer
- http://markusweimer.com
RESOURCES

- https://www.apache.org/foundation/how-it-works.html
- https://wiki.apache.org/hadoop/HowToContribute
- https://wiki.apache.org/hadoop/HowToCommit
- https://issues.apache.org/jira/browse/HADOOP (YARN/MAPREDUCE/HDFS/...)
- http://producingoss.com/ (Karl Fogel, Subversion)