This document contains categories of contributions and behaviors relevant to becoming an Apache Mesos Project Committer. As a committer candidate you can use this list to collect and aggregate data on what you have achieved so far. Neither is this an exhaustive list nor do all categories need to be addressed, since there can be reasonably diverse committer profiles. But some categories are always crucial to have covered, as for instance writing high quality source code and performing meaningful code reviews. Eventually, your nominator can hand in your filled-out list to the PMC to facilitate their reviewing your case.

Nomination date: _____

Candidate name: Joseph Wu

JIRA user name: kaysoky

Github user name: kaysoky

Nominator: _____
Name of the committer who nominates the candidate

Understanding of the Apache philosophy (The Apache Way) and Mesos project's goals
Demonstrated by:

- Community
  - Collaborative design and implementation of Maintenance Primitives.
  - Collaborative design of Optimistic Offers.
  - Collaborative design of the ContainerLogger Module.
  - Review of the Master Detector/Contender module: https://issues.apache.org/jira/browse/MESOS-4610
- Merit
  - Shipping high quality code that deals with complex and varied components. Ex.:
    - Flaky tests: https://reviews.apache.org/r/40849/
    - Libprocess finalization: https://reviews.apache.org/r/40266/
    - Stout JSON parsing: https://reviews.apache.org/r/38028/
    - Maintenance primitives: https://reviews.apache.org/r/36571/
    - ContainerLogger: https://reviews.apache.org/r/41002/
  - Test-driven development:
    - SSL + Scheduler library: https://issues.apache.org/jira/browse/MESOS-3753
    - Mitigating test crashing: https://issues.apache.org/jira/browse/MESOS-4633
- Openness & Charity
  - Collaborative and public feature implementation & review:
    - Optimistic offers: https://issues.apache.org/jira/browse/MESOS-1607
    - Maintenance primitives: https://issues.apache.org/jira/browse/MESOS-1474
    - Windows: https://issues.apache.org/jira/browse/MESOS-3094
    - CMake: https://issues.apache.org/jira/browse/MESOS-898
    - Container log rotation: https://issues.apache.org/jira/browse/MESOS-4136
  - Contribution to documentation:
    - Sandbox: https://reviews.apache.org/r/39949/
    - Messages.proto: https://reviews.apache.org/r/39457/
    - Maintenance: https://reviews.apache.org/r/37585/

        - Libprocess initialization ordering: https://reviews.apache.org/r/46806/

Commitment to the Mesos project
Demonstrated by:

- 120+ commits over just under a year.
- Talks given to the community:
    - Maintenance Primitives Release talk
    - ContainerLogger Release talk
    - (Upcoming) MesosCon NA 2016 Maintenance talk
- Being helpful on mailing lists and IRC. i.e.
    - http://www.mail-archive.com/user%40mesos.apache.org/msg06008.html
    - http://www.mail-archive.com/dev%40mesos.apache.org/msg35411.html
    - http://wilderness.apache.org/channels/?f=mesos/2016-02-23
    - http://wilderness.apache.org/channels/?f=mesos/2016-04-29

Knowledge and activity breadth and depth in the Mesos project as well as areas where the candidate could be a maintainer:

- Took part in implementing/designing:
    - Maintenance Primitives (endpoints and allocator)
    - ContainerLogger module
- Took part in adding features/fixing bugs:
    - Test infrastructure
    - Containerizers
    - PicoJSON
    - Libprocess lifecycle
    - SSL

Can the candidate be trusted to act the right way outside of known areas where he/she has already demonstrated committer-level expertise and behavior:

While I read and internalize code before I start on any changes, I always seek out more experienced committers for input on any code changes.

High quality source code
A committer vouches that the candidate writes high-quality, "readable" code, adhering to Mesos style and best practices standards, both formal and informal.
Name of the committer: _____

Community participation

- IRC
    - I show up as “mslackbot_: [joseph]” in the #mesos channel.
    - When there is activity on IRC, I will generally try to answer; especially if the question is something I can answer (i.e. relates to work I’ve done).
- Dev and user mailing list & JIRA issues
    - I check these lists everyday.

    - I will always try to answer if the subject is something I've work on; or if I'm interested in the subject.
- Developer Community Sync Meetings

Comments on Community Engagement:

I've become more active in the community as my experience in the Mesos ecosystem has grown. In addition to Mesos, I'm also involved in Marathon, Chronos, Docker, Jenkins (plugin), and DCOS.

Presentations at meetups, conferences, etc.: Bookmark

Reviews catching functionality issues

- Extra memory leak: https://reviews.apache.org/r/47804/#review134675
- Backwards compatibility concerns: https://reviews.apache.org/r/43269/#comment181059
- Documentation additions: https://reviews.apache.org/r/44620/#comment185118
- Test fix completeness: https://reviews.apache.org/r/43850/#comment181986
- A mistake: https://reviews.apache.org/r/43519/#review120598
- FD leak: https://reviews.apache.org/r/41253/#comment171077

Reviews catching style issues

- Spacing and looping: https://reviews.apache.org/r/44258/#review121770
- Windows code: https://reviews.apache.org/r/39852/#comment165735
- Back-and-forth: https://reviews.apache.org/r/41334/

Major source code contribution

- Maintenance Primitives - Operator endpoints for transitioning agents into three "maintenance modes". And related documentation.
- ContainerLogger module interface - Module to handle executor logs. Intended to solve the problem of stdout/stderr filling up disks on machines.
- JSON integer types - Added 64-bit integers to the JSON parsing code so that certain operations would not be converted lossily via doubles.

JIRA tickets created

Created 90+

JIRA tickets completed

Resolved 50+

Tests to ticket ratio:

- The test-not-crashing-as-much refactor means I've touched most of the tests in Mesos :)
- Wrote new tests for new features:
    - https://github.com/apache/mesos/blob/master/src/tests/master_maintenance_tests.cpp
    - https://github.com/apache/mesos/blob/master/src/tests/container_logger_tests.cpp

Write down <#tests written> / <#tickets that need new tests>, e.g. like this: "14 / 10". Also write down a list of all tickets that did not need new tests and why.
NOTE: This is just one suggestion for documenting testing efforts. The ultimate goal here is convincingly corroborating that all contributed source code is adequately covered by tests.

Initiated design documents:

- Maintenance Primitives.
- Optimistic Offers.
- ContainerLogger Module.

Reviewed design documents:

- Operator V1 API
- Oversubscription for Reservations

Helping new contributors:

- Agent resource overcommittment: https://reviews.apache.org/r/41305/#comment176725
- Libprocess/Stout/Mesos code split: https://reviews.apache.org/r/42068/#review114910

Contributions to Documentation:

- http://mesos.apache.org/documentation/latest/sandbox/
- http://mesos.apache.org/documentation/latest/logging/
- http://mesos.apache.org/documentation/latest/maintenance/

Dependability

Issues quickly / immediately solved that arose out of own contributions:

- Incorrect FD management: https://issues.apache.org/jira/browse/MESOS-4535
- Flaky test that I wrote: https://issues.apache.org/jira/browse/MESOS-4615
- Compilation: https://reviews.apache.org/r/43276/
- Awareness about some inverse offer edge cases: https://issues.apache.org/jira/browse/MESOS-4385