

MovingToJdk7and8

How to transition Hadoop to JDK7 and JDK8

There has been discussion on common-dev about how we're going to bump the minimum supported JDK version for Hadoop.

A number of proposals have surfaced, but let's start by going over some of the concerns to frame the discussion.

Background

Our compatibility guidelines are documented at <http://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/Compatibility.html>

branch-2 right now still supports JDK6. branch-2 works on JDK7, and there are anecdotal reports that it works on JDK8 too, with a little work.

From a vendor point of view, both Cloudera and Hortonworks released branch-2 based distros this year (2014), so it's likely that they will be doing development on branch-2 for the next year or two. Both seem to ask customers to use JDK7 already for HDP2/CDH5.

Looking at the big picture, it's reasonable to believe that the users of Apache Hadoop would be better served by us if we prioritized operational aspects such as rolling upgrades, wire-compatibility etc. for a couple of years. Since not everyone has moved to hadoop-2 yet, talk of more incompatibility between hadoop-2/hadoop-3 or between hadoop-3/hadoop-4 within the next 12 months would certainly be a big issue for users - especially w.r.t rolling upgrades, wire-compat etc.

Some improvements have also been made to support multiple versions of MR on the same cluster with their own classpaths. See [MAPREDUCE-4421](#) and [MAPREDUCE-1700](#) for further details. Furthermore, we also support allowing user applications to use a JDK different from Hadoop itself - for e.g. HDFS /YARN can run JDK7 while MR applications can run JDK6 if they chose right now.

Proposals

Listed in roughly chronological order. Not attaching names to keep this disinterested.

Proposal A

JDK7 is also EOL in April 2015, so we would like to avoid another JDK6 situation where we're stuck on JDK7 after it's EOL.

- Keep branch-2 the way it is, same JDK support and libraries. Keep rolling branch-2 releases.
- Move trunk to JDK8 now, along with bumping library versions.
- Release a Hadoop 3 in 2015 before JDK7 EOL

Proposal B

- Rename branch-2 to branch-3, so Hadoop 2.6 becomes Hadoop 3.0. Trunk becomes Hadoop 4.
- branch-3 would drop support for JDK6, update libraries like Guava and Jetty, no other compatibility changes with branch-2
- branch-4 would drop support for JDK7, update libraries like Guava and Jetty, no other compatibility changes with branch-3
- Release branch-3 in 2014 as a more immediate fix
- Release branch-4 in 2015 before JDK7 EOL

Proposal C

- Drop support for JDK6 in an intermediate branch-2 release, e.g. hadoop-2.8.
- Drop support for JDK7 in another intermediate branch-2 release, e.g. hadoop-2.15.

Proposal D

- Choose a branch-2 release to designate as the last JDK6 release, e.g. hadoop-2.y
- Set up hadoop-2.y builds with both JDK6 and JDK7
- Drop support for JDK6 in branch-2 and trunk
- Future branch-2 releases require JDK7+ and can use JDK7 APIs
- Discussion of JDK8 is tabled for now

Discussion

Proposal A

- Need to wait until 2015 before we can officially drop support for JDK6
- Need to wait until 2016 before we can officially drop support for JDK7?
- Potentially include many incompatible changes to wire-protocol, drop support for rolling-upgrade etc.

Proposal B

- This results in two new major releases in less than a year, which is potentially painful for users

Proposal C

- Dropping support for a JDK in a minor release maybe deemed incompatible, so this would require further discussion.
- Will continue to support rolling-upgrades, wire protocol compatibility etc.

Proposal D

- Seems to have some consensus on common-dev
- Not carte blanche to drop JDK support in minor releases
- Could do the 2.y release as soon as the upcoming Hadoop 2.5
- Same benefits as Proposal C regarding rolling upgrades, wire compat, etc