InvalidJiraIssues

Invalid JIRA Issues

This page tries to explain why some issues on the Apache JIRA get closed as 'invalid'.

The Apache JIRA server is used for two things

1. discussing and co-ordinating feature development of Apache Hadoop. We welcome people who want to get involved with this.
2. reporting and fixing bugs in the code

What it is not is *a way of people reporting their "I couldn't get Hadoop to work" problems*.

Given that Hadoop is used on thousands of machines by companies like Yahoo!, Facebook and eBay, we are reasonably confident that Hadoop works on:

- well configured servers.
- well configured networks.
- well configured Hadoop installations.

If Hadoop does not work for you, then these are the likely problems -your problems related to local configurations --especially your network

These are not bugs in Hadoop -please do not file bugs on JIRA about them

Bug reports of the form "I can't get Hadoop to work", are going to be closed as invalid, unless there is clear evidence that the problem exists in an Apache release.

Which raises another issue. JIRAs cannot be filed against Big Data Stacks that aren't bundling the Apache releases of Hadoop artifacts. We can't, because we don't all track what those changes are.

Here's a video on how to file good and bag bugs:

- Help! My Hadoop doesn't work!
- Accompanying slides

Please look at the video and understand why your JIRA was closed with a reference to this page. Then follow some of the suggestions below to help debug your cluster.

Don't assume a stack trace means a Hadoop bug

It's easy to see a stack trace with Hadoop classes at the top and think "Oh No! A Hadoop bug!" and rush to file a JIRA. Don't rush to do this, it won't address your problem in a timely manner. Even if it is a bug, you won't get a fix in your hand for months unless you are prepared to build and run your own Hadoop release. And if you are prepared to do that, you need to be able to track down stack traces yourself.

A stack trace with Hadoop classes in it can be caused for lots of reasons, primarily being "something happened while hadoop was trying to do some work". All the stack trace shows is `` `where the code was when the failure occurred` ``. If it's a network problem: you see a Hadoop stack trace. If it's a Kerberos problem: you see a stack trace. If it's a classpath problem trying to load a Hadoop-compatible filesystem —again, a stack trace.

Look at the stack and try to understand it. If there are nested stack traces, it's the one at the bottom which counts. The other ones show the convoluted steps to get there. Be advised: sometimes that bottom stack is being relayed from a server over IPC calls -it's a sign of something happening remotely, rather than in the application doing the reporting.

Read and Understand the Logs

Hadoop, Java Build tools and the operating system all log messages somewhere: to screen, to Hadoop service logs, to the OS logs. Learn to read these, rather than just posting them to the user lists and forums and asking for help.

- The log messages do try to be helpful. Sometimes they are only meaningful to people who understand how Hadoop works -but you are going to have to learn that anyway.
- Some of the log messages include URLs to pages in this wiki. Follow the links -they are there for a reason.
- Search engines are a surprisingly useful way of finding out what an error message machines. Before panicking and sending out messages to all known Hadoop mailing lists as well as filing critical-level JIRA issues, why not copy the string from the logs and paste into the search dialog of your search engine of choice -and see what comes up. The discussions on the user list archives are always worth reading, as are any references on stack overflow.
- Hadoop log levels can be dynamically changed by way of the `hadoop --loglevel` command (Hadoop 2.7+). Any of the core Hadoop services can have their log levels viewed and changed, via the LogLevel servlet -just add the path `/logLevel` to the base URL of the service.

Finding an answer by searching for it on the web is the fastest way to get help -and log messages are ideal for searching on

Embrace Metrics

Hadoop applications, and the JVMs themselves, all publish metric information for management tools to view. These help you see what is happening in a cluster, including which parts of the cluster are not actually running. As well as commercial products, the [https://ambari.apache.org/|Apache Ambari] lets you see what is going on. Oracle's jvisualvm tool lets you see what is happening inside a JVM. This is invaluable for debugging memory consumption and CPU performance problems.

Note also that management tools provide ways to configure, reconfigure and restart your cluster without you having to edit XML files by hand.

Ask on the User Mailing Lists

- The hadoop-user mailing list is the place at Apache where discussions on installation and configuration problems should take pace.
- Before asking questions, learn how to ask smart questions first. It will explain why a message called "HELP!!!" isn't going to get any
- Do not ask on the developer lists. Asking the developers for help on configuring your system is like emailing the linux-kernel mailing list for help getting /etc/fstab right. You will be silently ignored.
- Please don't email people directly. Not only are you unlikely to get any help, it's not a good way to start to build a relationship with people you may need to work with later.

Ask on Vendor Forums and Support Channels

If you are not using out-the-box Apache Hadoop, but instead a commercial Bug Data Stack, their support process should be your starting point

- If you have any installation issues -the vendors are the only place to expect help.
- These vendors provide their own mailing lists and forums. Please try there.
- They also provide (paid) support in some form or other. If Hadoop is critical for your organisation, and you aren't willing or able to learn to be self-sufficient, then you need to consider these.
- Any closed source parts of their stack cannot be addressed via ASF mailing lists or bug trackers.

Read the source, books and online articles

- There are a number of quality Books on Hadoop. These are worthwhile investments.
- There's lots of online articles -though you should seek recent articles that cover the version of Hadoop you are using.
- The source is all there for you to explore.

The source is ideal when you are really trying to understand the logs. Some IDEs (example: IntelliJ IDEA) will take a stack trace and work out the source tree, and you can search for all or part of an error string to find out its origin too. Debugging your own problems is a pragmatic way to learn your way round that source tree -just make sure you have the exact version of the source that you are running, so the stack traces match your source.

Keep your version of Hadoop current

Finally: the development and testing goes on Hadoop 2.7+, with important bug fixes backported to branch 2.6. If you have a problem with an older version of Hadoop: upgrade. If you aren't prepared to upgrade, you can't expect any help at all.

Returning to JIRA, it may seem unfair for the developers not to care about your "critical" issue and close it as invalid, despite the fact they are clearly the experts in Hadoop internals. However they ~~we~~ are busy trying to build the future of Hadoop, *the operating system for data*. Most of the people working on this are being paid to do so, either from companies whose business is built around selling supported Hadoop-based products, or from people who use in production internally. None of these people have the time to help you -because if they did help everyone with a problem, they'd never get anything done.

Those developers who are working full time for downstream redistributors of Hadoop works are being paid through support revenue -and their companies have support teams who will help -as can others on the Distributions and Commercial Support page. Those developer using Hadoop on internal projects probably get to field lots of internal support calls -which keeps them busy enough.

To summarise then: your issue was closed as JIRA is not the place to ask for help.

Do

1. Use a recent release of Hadoop. Older versions will have old bugs.
2. Look at any stack traces and try and understand them.
3. Read the error message and try to understand what it means.
4. Search on the web for the error message -and see what others did when they encountered it.
5. Ask on the Hadoop User lists and any vendor-specific forums or other support options they offer.
6. Join or create a local Hadoop User Group -to find nearby people to learn from and solve problems with.
7. Read how to ask good smart questions before asking bad ones.

Don't.

1. File JIRA issues on problems you have trying to get your own code to compile or run.
2. File JIRA issues on problems you have starting up your cluster, unless someone on the -user lists says "this is really a bug".
3. File JIRA issues on problems you have seen on outdated versions of Hadoop -update and try to replicate first.
4. File JIRA issues on problems you have with Apache Hadoop based products provided by third parties, unless these products are actually using the apache artifacts. Try to replicate on the ASF versions first.
5. Ask questions about using Hadoop on the developer lists. You will be deliberately ignored.
6. Send emails direct to developers. That's like emailing Linus because your Linux laptop doesn't power on.

That's why your JIRA issue was closed. It's not that the developers don't care that you can't get Hadoop to work -it's that they aren't the right people to ask.

Sorry.