HDTProductExperience

HDT Product Experience

This page is intended to collect thoughts about the overall product experience, from a user's point of view, for a mature product. It is hoped this will lead to a shared idealized vision of things like packaging, distribution, scope, and feature set that might guide more practical and concrete development plans. This is intended to be entirely from the user's point of view, without regard to what is realistic, achievable, or proper for us to take on. It's intended to answer the question "What would users really want?"

Many of these ideas might require cooperation, design, and implementation efforts in conjunction with other Hadoop projects. We might expect to provide leadership and some core technology to ensure high commonality.

Installation

- The user should be able to install using a single update site for all Hadoop-related Eclipse tools.
    - Not all components would need to be supplied by us.
    - It might (or might not) be better for the update site to be handled by Hadoop-Core.
    - It should at least be linked from all relevant projects.
- Simple Eclipse install
- Updates made available promptly.
- Compatible with current and recent Eclipse versions.

Configuration

- The user should be able to obtain a cluster's configuration by entering the hostname of the master namenode.
    - All Hadoop services ought to coordinate to publish the necessary configuration information, downloadable via a single URL.
    - The cluster should be able to provide multiple alternative configurations for the user to select from.
- The user should be able to override cluster-provided information.
    - The user should be able to provide multiple overrides of cluster-provided information for different purposes, and select where appropriate.
        - The exact definition of "where appropriate" may get tricky.

Feature Set

This list is intended to be broad features that users might expect to be available across components.

In some cases, these duplicate functionality that is or might be provided via the web UI. It's fine to integrate the web UI into HDT, but we should be looking for ways to link more deeply into to Eclipse. Errors in the log, for example, should take us to Java source for classes mentioned, and failing tasks and jobs should lead us to the appropriate step in the appropriate script editor.

- Data browsing
    - Filesystem browsing, for all supported filesystems
    - Database browsing, for all supported databases
    - Facilities for viewing large data sets – sampling, searching, random access. Should *not* run out of memory trying to look at files in HDFS.
- Job submission, for all types of jobs
- Job tracking, for all types of jobs
    - This would include drilling down into more complex composite jobs, for example Pig, or the multiple jobs that make up a Hive select.
    - Ultimately, this drills down to tasks.
    - Counters
- Log browsing, for logs from all types of jobs and tasks
- Debugging
    - Remote attachment of Eclipse to tasks
    - Remote debugging of scripts (for example, pause after first intermediate join to examine result before passing it to next step, etc.)
    - Counters
- Cluster monitoring (perhaps in one compact display)
    - Service and node status
    - Cluster load monitoring
    - Filesystem available space display
    - Alerts (configurable: job completion, failure, space, node or cluster status changes).
- Project/Source support
    - Project creation that adds required libraries for developing Map/Reduce or YARN (or other future api) applications
    - Templates, quick fixes and other support for application code
    - static code analysis rules (PMD or similar)
    - MRUnit based test code support, and launching

Administration

Ideally, we would provide optional administration facilities. These would almost certainly duplicate the web interfaces (which they should reuse). The advantages of including with Eclipe are

- One-stop shopping

- Tighter integration – for example, if service status display shows a service is down, allowing you to restart the service in place would be handy – or at least taking you to the appropriate admin page.
    - Taking you to the appropriate admin page would allow better integration with 3rd-party tools such as Cloudera Manager.
- Tools for managing filesystem space. Who's using what, where.

Security

- The tools should properly integrate, and be tested against, a locked-down secure install.
    - "Secure" may be overstating things now, but security will become an increasing concern as clusters are used by increasingly heterogeneous user communities.
- Support for multiple user credentials on a particular cluster as part of the user-selectable customized configuration.
- The tools should aid the user in identifying the security status of resources, and tracking down failures due to security restrictions.
- Tools could aid the user in determining whether jobs comply with defined security policies.