iRODS@RENCI

Leesa Brieger, Jason Coposky, Vijay Dantuluri, Kevin Gamiel, Ray Idaszak,
Oleg Kapeljushnik, Nassib Nassar, Jason Reilly, Michael Stealey, Lisa Stillwell, Xiaoshu Wang

irods@renci.org
Renaissance Computing Institute
University of North Carolina, Chapel Hill
{leesa}@renci.org

Abstract

Development and support of iRODS has expanded, boosted by the launch of the new irods@renci team at the Renaissance Computing Institute (RENCI), a research unit of UNC Chapel Hill. RENCI is a center for development and deployment of advanced cyber technologies, and, in close collaboration with the Data Intensive Cyber Environments (DICE) group, has now dedicated resources to supporting and expanding iRODS functionalities. Activities are spinning up and include code hardening and widening the test coverage, providing a collaborative development environment to facilitate testing and community participation, improved Windows support, additional drivers, support for a Java rule engine, iDrop development, PHP and other API support, along with microservice development for specialized data grid deployment and community support.

The irods@renci effort also serves to set us on the path toward a commercial support model, in the style of RedHat/Fedora, with regular releases of the iRODS research code from the DICE group at 3-4 month intervals and less frequent but more highly tested and stable releases of iRODS-Enterprise (iRODS-E) from RENCI at roughly 18-month intervals. The release of the hardened code will allow us to create and customize service level agreements to the demands of a user community that reclaims some form of commercial and sustainable support for this technology. This also puts iRODS on the path toward very long-term sustainability.

Index Keyword Terms—iRODS, data grid, sustainable software development, RedHat Fedora model, community architecture, service level agreement

1. Introduction

The University of North Carolina's investment in data management technology began when it welcomed the first members of the DICE group to Chapel Hill in August 2008, with joint appointments in RENCI, a research unit of UNC Chapel Hill, and at the School of Information and Library Science (SILS). This investment has continued and grown as the DICE data management technology has transitioned from the Storage Resource Broker (SRB) to iRODS. In particular, RENCI, a cyberinfrastructure development center that is equipped to complement and expand DICE support for iRODS, has now spun up a group, irods@renci, with a strongly synergistic relationship to the DICE group.

As RENCI moves into iRODS support and development, the RENCI-DICE combination offers a stronger response to user need. Starting out with straightforward support for existing DICE initiatives, irods@renci begins by offloading weighty support tasks from the DICE group, while bringing iRODS expertise up to speed in RENCI. As familiarity with the DICE methods and the iRODS code base grow, the RENCI development environment is pulled in to facilitate the cross-group development collaboration *and* the integration of community contributions. As irods@renci takes its place alongside the DICE group, collaboration between the two groups strengthens, and ownership of some code components is placed with RENCI. RENCI sustainable software development practices are leveraged to bring hardening and optimization more forcefully into the iRODS code.

Along with its development activities, RENCI is also working to grow and support iRODS user communities among its own shareholders. Supporting those initiatives allows RENCI to move into more

general iRODS user support for other communities with a better understanding of user issues.

Promoting and achieving long-term sustainability of the iRODS technology and providing its users with the assurance that support for the technology is solid and will continue into the future are the primary goals of the RENCI/UNC investment in iRODS. This RENCI-DICE collaboration positions the iRODS community to be able to expect greater levels of support and creates an environment that supports funding models beyond the traditional public funding for research code development. New models based more on service level agreements and quality of service requirements can complement the basic research character of iRODS development, as it has been funded, and move the iRODS technology toward a more sustainable future and its users toward a more solid support base.

2. Software Development at RENCI

2.1. Agile Development

The irods@renci group is organizing its activities using an Agile development approach [1, 2], which is incremental and iterative. Development cycles are short, allowing planning and implementation to be revisited often, building flexibility into the process and providing the ability to adapt to changes in the technical requirements, community requests, staff availability, etc. Short-cycle collaboration across the cross-functional irods@renci and DICE teams is the most natural way to approach the multi-faceted development that goes into a technology like iRODS.

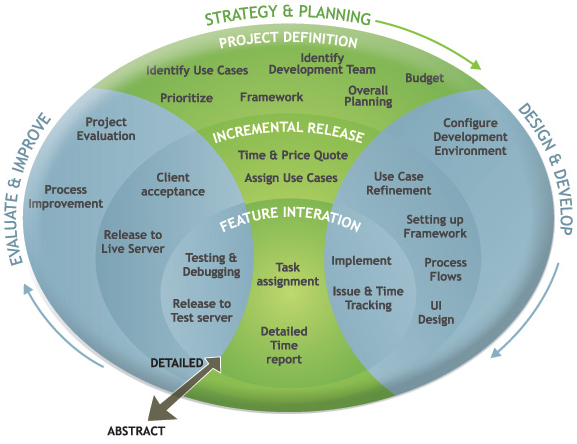

Figure 1. Agile Development [3]

In fact, the iRODS technology does not lend itself to a waterfall model of planning, design, implementation, and maintenance according to initial, fixed specifications that do not change as development progresses. iRODS has evolved according to a *de facto* Agile methodology, driven as it is by the incoming needs of community stakeholders. RENCI plans are to somewhat formalize the DICE group's approach in order to expand and reinforce procedures for code hardening, collecting community input, prioritizing feature requests, etc, all to support long-term sustainability.

2.2. Collaborative Development Environment

The collaborative development environment (CDE) at RENCI provides a centralized virtual environment that allows the trusted administrators and developers of the iRODS project, as well as community contributors, to collaborate on code development. It supports activities such as committing and testing new contributions in a coordinated fashion, tracking bugs and feature requests throughout their life cycle, documentation and reporting, and managing dependencies and artifact support.

RENCI's software development approach is being integrated into the DICE/iRODS support effort in a manner that does not interrupt current development. Git [4] repositories at RENCI now house the Jargon core, iDrop, and Fedora projects, along with the PHP client library. The Git version control system allows decentralized revision tracking, facilitates branching and merging, and is particularly suited to distributed, non-linear development environment. In preparation for future migration into Git, the iRODS trunk SVN repository at UCSD is now automatically mirrored in Git at RENCI. See Figure 2.

GForge [5], a web-based project management and collaboration software suite, forms the basis for RENCI's CDE. It provides a community-based environment for source code management, project hosting, access controls, messaging, reporting services, and trackers that can link code changes to tasks or bugs. RENCI is also using GForge for reporting and tracking tasks in the iRODS support areas. RENCI's GForge area is accessible at https://code.renci.org/gf/; projects are at https://code.renci.org/gf/project/. Anonymous checkouts are selectively enabled; it is recommended that interested users request user accounts and project access.

Hudson [6] provides the continuous integration (CI) environment that RENCI has adopted to automate the continuous builds and tests for iRODS server and clients. Continuous integration allows new or modified code to be integrated with an existing code repository with quality control testing via automated builds for integration error detection. Distributed builds are also supported; jobs can be farmed out to slave build machines. The GForge reporting plug-in couples RENCI's GForge installation to the Hudson installation using SOAP.

Hudson can be set to trigger builds based on new code commits to a repository as well as on a periodic

build schedule. This means integration problems can be better targeted, leading to incremental quality control and to a more cohesive and rapid software development process. This approach allows developers more easily (and confidently) to integrate changes to the project and users more easily to obtain a fresh build.

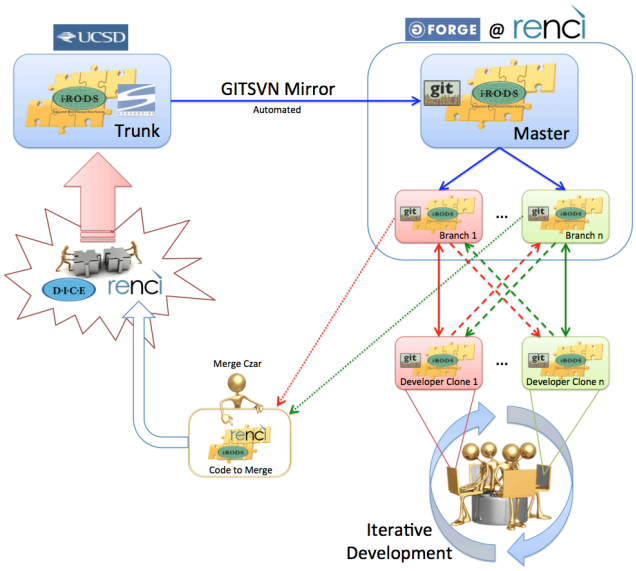

Figure 2. RENCI's Git mirror of the UCSD SVN iRODS repository.

Sonatype Nexus [7] is the Maven repository that RENCI uses for managing software artifacts required for development, deployment, and provisioning of Java code. Jargon is now a Maven multi-project project with reports (Cobertura test coverage, javadocs, etc.) and project information (issue tracking, dependency convergence, etc.): https://ci-dev.renci.org/site/jargon. Dependency management has been added for dependency convergence across sub-projects, and repository mappings point to a new Nexus instance; future releases & snapshots will be found at https://ci-dev.renci.org/nexus. For a look at the Maven-generated Jargon project information page, see https://ci-dev.renci.org/site/jargon/jargon-core/project-info.html.

The GForge/Hudson/Nexus combination provides an infrastructure that supports community-based software development. Moving this development environment infrastructure into the iRODS development process facilitates the interactions between DICE and RENCI, the coordination of the complementary efforts, and the incorporation of contributions from the communities of users.

3. irods@renci Activities

3.1. CI, Testing, and Code Hardening

The RENCI contribution to iRODS will enhance the quality and reliability of the software and facilitate more automated testing of the code. CI techniques are used to align users and developers alike into a structured release cycle such that they know exactly when to expect a new or updated release to their product.

Development cycles are supported with the Hudson extensible CI environment that automates the running of unit tests and allows on-the-fly testing of new commits. IRODS unit tests are now being built into the RENCI Hudson environment. The community code release cycles are already extensively tested by the DICE group using tinderbox and NMI, but code coverage can be greatly widened and testing can be largely automated by relying more heavily on the Hudson environment for the wider code testing. This will reduce time spent in test mode and yet will allow much greater coverage of the code. The enterprise code will undergo exhaustive testing, supported by the CI environment.

Code hardening and refactoring currently go into the community code releases as possible, based on time and funding constraints. However, these constraints are severe, since iRODS funding is largely for development rather than for hardening and optimization. Releases of the iRODS-Enterprise code will require more extensive software engineering practices; RENCI is gearing up to contribute extensively to this hardening of the iRODS code.

The iRODS code will exist at three distinct levels, each with its own degree of hardening: the development level, the standard release level, and the enterprise release level.

3.1.1. Development Level

Development level code is based upon the most recent code releases but without any guarantee of stability or backwards compatibility. This is the day-to-day code on which developers work out the integration for the latest patches, features and fixes. The development level code serves as the testbed for the next version of standard release and enterprise level codes.

3.1.2. Community Release Level

Community code release is based on the previous code release, incorporating the bug fixes and features of the development level that have been subsequently developed and sufficiently tested for release in the research code.

3.1.3. Enterprise Release Level

Enterprise level is hardened, production-ready code that has demonstrated itself to be stable on a wide set of recognized platforms. Code at this release level can be

assured to be backwards compatible for a specific range of previous releases; this is well documented and defined ahead of release. This code will not be incorporated into the daily development cycle, and would only see specific, well documented patches for anomalies as they arise.

3.2. Toward a Unified Cross-Platform Code

One immediate RENCI contribution is to migrate platform-specific APIs and system calls away from server-level code, thereby providing a strategy for facilitated code support on a wide range of platforms. The first step in this cross-platform approach is to compile the code with g++ so that libraries such as Boost C++ [8] can be incorporated. These libraries allow the streamlining of cross-platform implementations and the abstracting out of many platform-specific operations, such as threading, regular expressions, character encoding, signals, forking, etc.

The initial g++ port has been done, and synchronization with the iRODS trunk code and distribution of the converted code is expected to be completed for iRODS release 2.6 or 3.0.

3.3. Windows Support

The first beneficiary of the cross-platform support is the iRODS server for Windows platforms. Due to scarcity of resources, the Windows iRODS non-iCAT-enabled server has not kept pace with the iRODS releases since 2.0; further, there never has been, until now, iRODS support for an iCAT-enabled server on Windows. The first thing RENCI's cross-platform unification of the code will allow is the upgrade of the non-iCAT-enabled server to the current iRODS release. Following closely is the planned release of an iCAT-enabled Windows server.

The Windows iExplorer iRODS client will now also benefit from enhanced support from the irods@renci group, which is developing a .NET implementation of the iRODS client to provide native integration with the .NET framework. The iRODS.NET client will connect to the iRODS server from the .NET platform, performing required iRODS client operations and providing access to administrative and standard user tasks. It will support .NET 3.5 and up.

This native Windows client will enable .NET web and windows application development for interaction with iRODS. It will also enable a variety of other functionalities that are being explored now at RENCI:

- Windows PowerShell commands to simulate the icommands unix client
- a local Windows drive or folder mounted to a (remote) iRODS collections
- LinqToIRODS development to query an iRODS server
- Excel ribbon toolbar to interact with iRODS.

3.4. Database Activities

As part of the RENCI effort to bring up and support iRODS iCAT-enabled servers on Windows platforms, the databases that iRODS currently supports- PostgreSQL, Oracle, and MySQL- are being incorporated into the iRODS build for the Windows platform. Additionally, Microsoft SQL Server support is also being provided in iRODS so that on Windows the iCAT metadata catalogue can be implemented with this database.

Even as irods@renci spins up its iRODS development activities, the group's contributions extend to other database activities taking place in collaboration with the DICE group. Recent DataBase Resource (DBR) development allows iRODS to access and query external databases, managing them as resources. RENCI managed the DBR testing with Postgres, Oracle, and MySQL database instances, both local and remote to an iRODS server. Recent iCAT special query development allows a data grid administrator to implement SQL strings that enable authorized queries of the iCAT DB. RENCI contributed to the effort with usage and applicability examples and testing.

As irods@renci database activities progress, attention will turn toward questions of iCAT data redundancy and failover mechanisms. Additionally, improvements in database performance will be investigated and recommendations made; if need be, these will be implemented as part of planned development activities using standard performance tuning and optimization techniques.

3.5. Java Rule Engine

The rule engine component of iRODS is currently being redesigned and improved within the DICE group, to come out within the next couple of iRODS releases. RENCI is building on that activity with an investigation of a Java-based rule engine that will leverage existing Jargon services to implement a streamlined version of the rule engine. Design and resource requirements are being analyzed to determine the best inter-process communication method between the (next-gen) C-based rule engine and Java, semantic synchronization with the based rule engine, and integration with Jargon.

3.6. iRODS Clients

In addition to the Windows client, RENCI is moving to support other client libraries in use by the user community. The approach is to begin with user support, monitoring the iRODS chat discussion and assisting with troubleshooting and bug fixes, then assume ownership of the client support. The client libraries are then migrated over to the RENCI Git repository in GForge, where unit testing is incorporated into the Hudson continuous integration environment. In the longer term, new feature sets will be explored, based

on community request, and code hardening and standardization among client APIs will be pursued.

RENCI is currently supporting the PHP client that underpins the generic iRODS web client. The first steps are to consolidate the code and complete unit testing on this API, which has not been supported in DICE for a couple of years. PHPUnit will be used to develop a set of unit tests for the core PHP API. The likely approach for future development of the PHP API is that a minimal API library of pure PHP will continue to be supported, with more support going into PHP on JVM. This would allow PHP to call Java objects, so that the elaborate library of Jargon services would be accessible from PHP.

As funding allows and/or user demand requires, RENCI will move into support for other client APIs.

3.7. Special Projects

3.7.1. Shibboleth Authentication

The request for a Shibboleth authentication mechanism into iRODS surfaces regularly from the user community. RENCI will be using its participation in the TUCASI data-Infrastructure Project (TIP) to motivate its work with UNC ITS infrastructure providers to implement Shibboleth authentication for the Triangle-wide TIP federated environment. TUCASI is the Triangle Universities Center for Advanced Studies Inc., funded by the Research Triangle Foundation and located in Research Triangle Park. It funds scholarly collaboration among the three Triangle universities; its funding of the TIP project is dedicated to establishing a federated data environment among these universities.

3.7.2. Climate Modeling with NCDC

The National Climate Data Center (NCDC) is the world's largest active archive of climate data. As climate studies become increasingly important in understanding the future of our planet, NCDC is playing a central role in supplying data and services to the climate community as well as to the public. RENCI is working with NCDC to customize iRODS services in response to the demands of this center. The current collaboration, in preparation of NCDC's engagement in large-scale climate modeling, is a pilot project to prototype data flow and workflow management for data-intensive climate computations.

3.7.3. Sequencing and Genomics

Genomics and bioinformatics are areas in which data generation is outpacing the biologists' capacity to manage the data. RENCI is moving to bring iRODS technology in to support the burgeoning demands of these data-intensive fields. In particular, the first approach to this is to establish a bridge between iRODS and Hadoop by developing a Hadoop file system driver for iRODS. This will allow Hadoop files to be shared, in an easy and controlled way, between data contributors.

Hadoop is an open-source Java based software framework that supports large-scale distributed data processing. At the heart of Hadoop are the Hadoop Distributed File System (HDFS) and the MapReduce computational model. The file system divides large data sets into smaller blocks, spreading them across many machines; the computational model simplifies data processing by decomposing an application into a series of components (mappers and reducers) that enable a form of distributed computing that is not only robust and scalable but also simple and accessible.

Connecting iRODS and HDFS will extend the data management capabilities of Hadoop (through iRODS), while augmenting the data processing capabilities of iRODS (through Hadoop).

3.7.4. Other Projects

In addition, RENCI supports the use of iRODS technology for the Cyberinfrastructure for Billions of Records (CI-BER) project sponsored by NARA and the DHS-sponsored NC Bio Preparedness project.

4. The Open Source Sustainable Model

4.1. Red Hat

Red Hat (RH) is one of the world's premier open source technology companies, sponsoring leading-edge Fedora Linux [9] and providing subscription services to Red Hat Enterprise Linux.

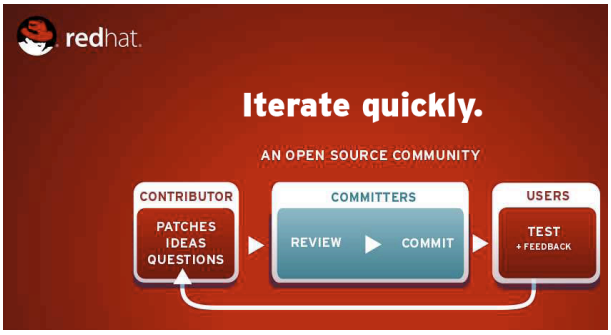

Figure 3. Close ties to contributors and testers in the user community allow fast turnover. [10]

Red Hat's very successful approach to open source is based on their strong relationship with the user-contributor community. The Open Source Way [11], Red Hat's introduction to creating and nurturing communities of contributors, says that, "Contributors are the oxygen" because they are vital to the health and sustainability of the technology. At the same time, the Red Hat mission statement is: *To be the catalyst in communities of customers, contributors, and partners*

creating better technology the open source way. Community is critical to effective open source technology, and open source technology provides robust, sustainable, interoperable software with open standards to the community.

Adaptability is a crucial component in open source sustainability models. As Michael Tiemann, VP of Open Source Affairs at Red Hat and President of the Open Source Affairs Initiative, has pointed out [12], adaptability is now understood as a strategic capability and a key to sustainability (survival of the most adaptable), leading to less abandonment and greater re-use of software.

The Red Hat Fedora model, built on what has been proven to make an open source model work, is basically this:

- the community contributes heavily to Fedora Linux development;
- the developer community and Red Hat select features for integration into the Fedora Linux open source code that is released several times per year;
- community usage contributes to testing and troubleshooting;
- the Fedora code repository is periodically forked and hardened by Red Hat into a tested and certified open source code, Red Hat Enterprise Linux (RHEL), which is released much less frequently than Fedora and for which commercial support subscriptions can be purchased.

Community input keeps the code contributions and evolving directions relevant to the users, RH testing keeps the developments reliable, frequent release of the Fedora community code keeps evolution and debugging active, while the RHEL release is solid enough that support of the code is cost-effective.

4.2. iRODS-Enterprise

It is useful to examine the parallels between Red Hat open source practice and iRODS practice when exploring new sustainability models for the open source iRODS.

The success of iRODS (and of SRB before it) has largely been driven by the connection of the DICE groups to the communities of iRODS users, and the open source nature of iRODS is now a strong element in cultivating communities of practice around the technology. Vibrant communities of contributors have been (and will continue to be) crucial to growing the technology and allowing it to evolve in directions that keep it relevant to its users.

RENCI is building on the Red Hat model to incorporate the best practices of the open source model into iRODS support. The incorporation of some practices, such as agile development for adaptability and flexibility and a more formal community architecture model for collecting user contributions and hardening those into code releases, will allow for solid expansion of support for the technology, improving productivity and collaboration.

To move toward new funding models beyond those of traditional public funding, RENCI is also preparing to emulate Red Hat's Fedora model: the iRODS community code will continue to be developed and released by the DICE group at the current frequency, while RENCI will periodically freeze this code, harden and test it to higher levels of reliability, and release it on a slower schedule, as iRODS-Enterprise, with subscription support services available for commercial clients.

While both code releases will be open source, the differences between the community code and the hardened code will be several and include the following:

- the community code will be for technical enthusiasts using iRODS in non-critical computing environments, while the hardened code will be for users looking for stable, supported, and certified iRODS (business, government, etc);
- the community code will be bleeding-edge technology, released early and often, while the hardened code will be stable, reliable, and broadly supported, easy to deploy and manage;
- the community code will be tested by the developer community, while the hardened code will be rigorously tested by RENCI, DICE, partners and the beta team.

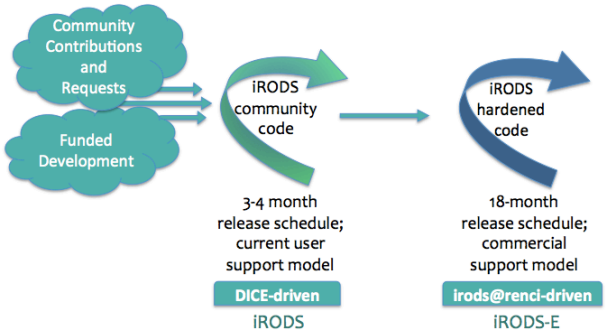

Figure 4. iRODS community code and hardened code releases; inspired by the RH Fedora model.

5. Conclusion

As RENCI ramps up its contributions and widens its support of the iRODS technology, quality of service can be heightened, and commercial service level agreements tailored to specific user needs become more sustainable. This opens up new funding sources, boosting the long-term sustainability of this technology and the services that depend on it.

References

[1] Agile ALM: Impossible or Best of Both? Available at http://www.purecm.com/whitePapers.php (registration necessary)

[2] Agile Scrum and PureCM. Available at http://www.purecm.com/whitePapers.php (registration necessary)

[3] Agile Development. http://www.initto.com/agile-development.html

[4] Git. http://git-scm.com/

[5] GForge. http://gforge.org/gf/

[6] Hudson Continuous Integration. http://hudson-ci.org/

[7] Maven Nexus. http://nexus.sonatype.org/

[8] Boost. http://www.boost.org/

[9] http://fedoraproject.org/

[10] Freedom Isn't Free. Open Source Mechanics. Gunnar Hellekson, Chief Technology Strategist for Red Hat's US Public Sector group. Invited speaker, NSF Workshop on Creating a Scientific Software Innovation Institute (S2I2) for Environmental Observatory Communities, October 5, 2010.

[11] The Open Source Way. http://www.theopensourceway.org/book/The_Open_Source_Way-Introduction.html

[12] "Tiemann on transforming IT the open source way," 3 Jun 2010 by Jonathan Opp (Red Hat). http://opensource.com/business/10/6/tiemann-transforming-it-open-source-way