The Open Cluster Framework (OCF) − A Plan for achieving World Domination™ in Clustering ;−)

Alan Robertson − *IBM Linux Technology Center* − <alanr@unix.sh> / <alanr@us.ibm.com>

ABSTRACT

One of the most commonly identified features which is felt to be necessary for Linux™ to be considered "enterprise−ready" is High−Availability. High−Availability (HA) systems provide increased service availability through clustering techniques. Linux is also well−known for its Beowulf High Performance clustering software − which provides the most cost−effective supercomputing available for any platform.

As befits the importance of these capabilities, a number of open source clustering projects have been created. These projects were been created independently largely for complex historical reasons, not generally because of political, philosophical or licensing differences. Because of these kinds of historical reasons, they originally shared little or no code. This minimized the benefits of the open source model, which both encourages and benefits from the sharing of common components. However, many of them share the need for a component for resetting cluster members. A component was created for filling this need with the specific intent of being a common component across open HA systems. This was quite successful, and this component has become standard across most open source HA systems. All of the projects involved have benefitted from this commonality.

In light of this success, the author began to search for more ways to extend these benefits across a broader set of cluster infrastructure components. Towards this end, we have begun an effort to create a standard Open Clustering Framework specification, and have begun to implement a reference implementation.

Background

This document has many holes in it, and will be filled out to include lots of missing details.

The Open Cluster Framework

OCF APIs

The Open Cluster Framework will define a series of APIs − a set of external APIs and a set of internal APIs. It is intended that many clustering systems will eventually conform to the applicable external APIs. Defining these APIs is fundamentally a standards effort whose output includes these APIs. A second part of the OCF effort is to define a reference implementation of the standards. It is expected that the reference implementation will be developed in parallel with the standards themselves.

Clustering implementations which are based on the the OCF reference implementation will also conform to the internal (component) APIs as well. It is perfectly acceptable for a completely closed system not based in any way on the reference implementation to conform to the internal APIs, but it is not expected to be a common occurrence.

OCF Reference Implementation

The OCF reference implementation will be available for download − and will be licensed under an open source license − see the section on OCF IP policies for more details. This effort will be a full−fledged community development effort to develop a set of common clustering utilities which could be used in a demanding commercial setting. It is also the intent of the reference implementation that it could be used in conjunction with commercial components and implementations.

The OCF Process

The OCF working group will be self−governing, operating under as few constraints as are consistent with the needs of the task, legal prudence, and the wishes of the participants.

This section obviously needs a good bit more work, and needs to cover things like what membership means, who can participate, what the method of reaching a decision is, etc.

OCF IP Policies

All OCF documentation will be available free of charge. There will be no charge associated with writing software which conforms to the specification, and no standard will be approved which is known to require the payment of royalties. In other words, the standards we create will be Royalty−Free (RF). However, there may be a trademarked designation for implementations which have been officially certified as conforming to the specification. If such a designation is created, there will likely be a charge for it. The documentation will be provided under a license similar to the Open Publication License or the GNU Free Documentation License. The software of the reference implementation will be provided under the GNU Lesser General Public License (LGPL). Certain selected pieces of software which are classified more as templates, and have not significant algorithmic or testing value may be licensed under the BSD license. It isn't clear if these will be provided as part of the documentation or part of the source to the reference implementation, or some in each. Software or documentation "templates" will be under a slightly less restrictive license than the components of the framework itself.

These practices are expected to be consistent with the recommendations of the Free Standards Group for their standards. We will likely adopt their IP policy for standards as is. We are currently exploring affiliation with the Free Standards Group (FSG).

The OCF working group will refine and ratify these policies.

Ways to Participate in the OCF Standards Process

For legal reasons, it is our expectation that there will be three classes of participation in the OCF standards working group.

- Participants − people who subscribe to the associated standards mailing lists. All of these people will be informed of the IP policies of the OCF process, and their continued participation constitutes agreement with them.
- Individual members − people who actively contribute language and suggestions to the standards process and have formally agreed to be bound by the OCF IP policy as individuals (not their employers).
- Corporate members − people who participate on behalf of their company, and who will be asked to conform to the IP policy on behalf of their company, and who's company has signed the appropriate agreements for them to do so.

Ways to Conform to the OCF Standards

There are several ways which one can write software which conforms to the OCF standards. A few examples are given below:

- Provide a product consisting strictly of (a subset of) software components taken from the reference implementation.
- Provide a product consisting of mainly components from the reference implementation, with a few unique components − which are known to meet the standard APIs.
- Provide a product consisting of a few components from the reference implementation, and a large body of software not from the reference implementation, but with a set of interfaces to the functions provided by this product which conform to the standardized APIs.
- Provide a product consisting entirely of proprietary software with interfaces which conform to the reference APIs, and whose internal architecture bears no resemblance to OCF components.

All of these are legitimate ways to conform to the standard. The first ones are lower−cost to write and maintain, and enjoy the well−known benefits of the open source development model. A company may start out with the in one model, and migrate to the "mostly open" approach over time.

This flexibility will allow a company to continue to meet the needs of their customers, and provide familiar interfaces, while migrating over time to have less and less unique software in their package. This lets them concentrate their development effort on the components which their customers have the most unique needs.

The Future of the Linux-HA project

It is currently planned that the Linux−HA software be migrated to this new framework model, and that it will contribute to this framework project and provide reference components for it. Customers which are currently using the software will be able to use the new versions as they become available. The transformation of the heartbeat software to this new model is already underway.

The Linux-HA To Do list

The Linux−HA To Do list will be updated to include items associated with OCF development activities.

Conclusion

Everything from here on down is old stuff, and not necessarily relevant to this document ;-). Please feel free to ignore it at this time ;-)

Acknowledgments

To Learn More

The Linux−HA web site can be found at **[Rob00]**. *Heartbeat* can be downloaded (in source or RPM format) from the Linux−HA web site download page at: http://linux−ha.org/download/. Information on subscribing to the various Linux−HA mailing lists can be found on the contact page at: http://linux−ha.org/contact/. The Linux FailSafe project is described in detail in **[Vas00]**.

References

[Milz99] Milz, Harald: "The Linux High Availability HOWTO". http://metalab.unc.edu/pub/linux/ALPHA/linux−ha/High−Availability−HOWTO.html

[Phi98] *In Search of Clusters*, by Gregory F. Pfister, 2nd Edition 1998, Prentice Hall PTR.

[Rob00] Robertson, A. L.,: "The High−Availability Linux Project". Http://linux−ha.org/

[Twe00] Tweedie, S. C.,: "Barrier Operations". http://linux−ha.org/PhaseII/WhitePapers/sct/barrier.txt

[Vas00] Vasa, M.,: "The Linux Fail Safe Project". http://oss.sgi.com/projects/failsafe/