Clustering On Linux™

Clustering On Linux™

The Present, The Future

The Open Clustering Framework

Alan Robertson
IBM Linux Technology Center
alanr@unix.sh

Agenda

- Why Clustering Matters
- Opportunities
- The Present
- What's wrong with this
- What to do?
- Open Cluster Framework
- Conclusions

Why Linux Clustering Matters

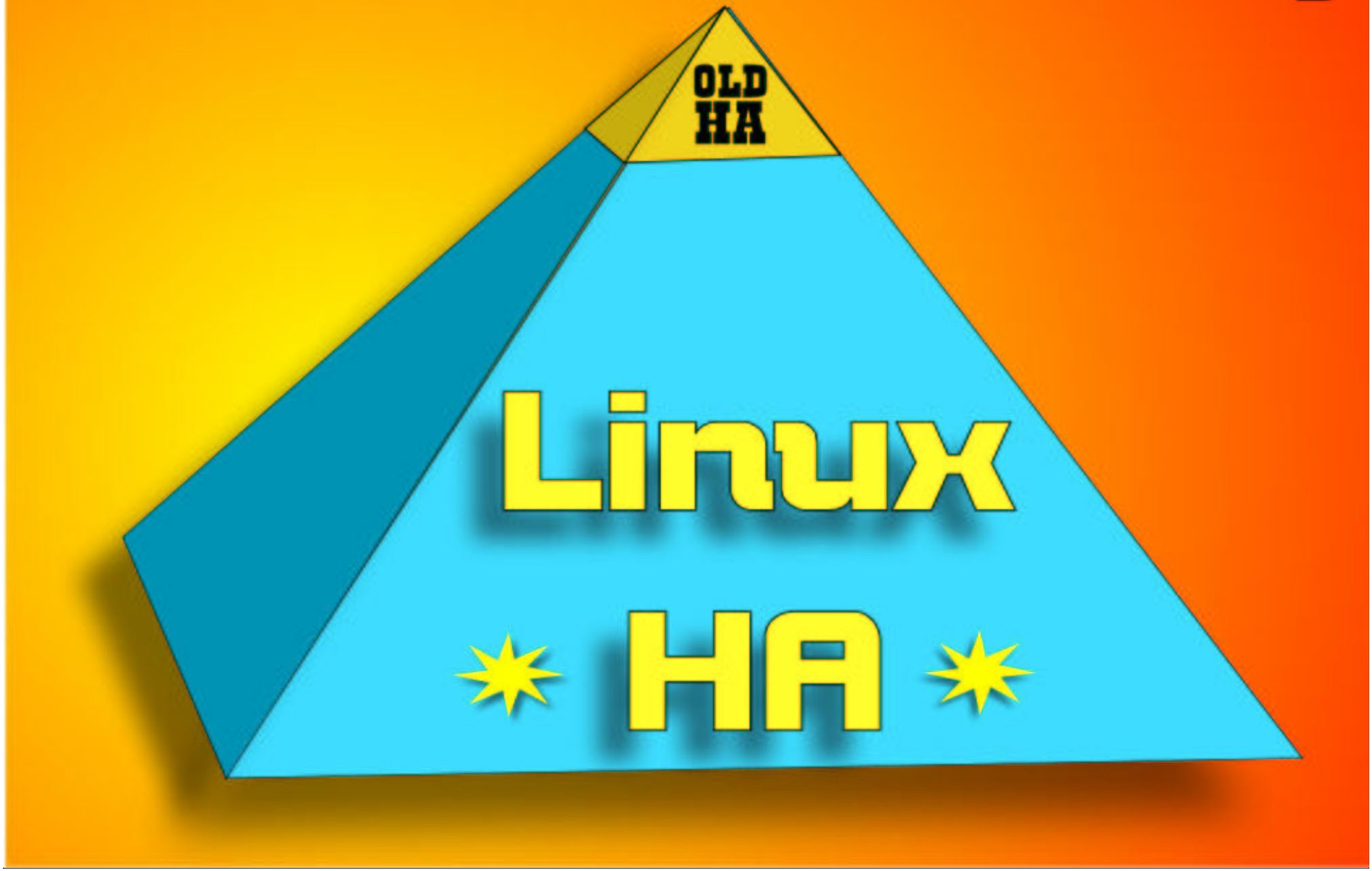
- Linux is the most cost–effective, reliable platform on the planet
- Clusters are the most cost–effective means to:
  - High–Availability (HA)
  - High–Performance (HPC)
- Clustering extends the "reach" of Linux upwards
- Linux Clustering is revolutionary!

High-Performance Linux Clustering

- **All** of the top machines in the "Top 500" supercomputers in the world are clusters
- Of the "Top 500", by far the most cost-effective machines are Linux clusters

Market 2/1/02 4:51 AM

Potential User Community



The Present

- At least 5 OSS HA "products"
- 10–20 Proprietary HA Products
- A VERY large number of HPC "offerings"
- Each has it's own API and expectations
- No "900 pound gorillas" in the Linux market to create/dictate "standards"

What's wrong with that?

- End–users are confused
- Middleware vendors are confused
- OSS developers are confused
- Software is confused

- EVERYONE is confused

OSS Clustering: An Embarassment of Riches

- There are five (and counting) independent general purpose HA systems
- There are many independent HP clustering systems
- These systems can share little or no code – – developer efforts are diluted, all systems are weakened
- There is no particularly good reason for this (except history)

Wasted Effort

- Middleware vendors try and provide compatibility with different clustering products
- OSS developers try and provide the same functions over and over *(and over…)*
- OSS developers try and provide compatibility with different middleware products

Who is hurt?

- OSS projects and customers
  - Effort duplication
  - Developer, tester dilution
- Proprietary vendors – duplicate efforts with each other and OSS projects, and have trouble recruiting middleware partners
- Middleware vendors – effort duplication, and lots of workload from HA vendors
- End users – incomplete, incompatible products, confusion, fear, uncertainty, doubt
- Linux community –weakened clustering offerings

What to do?

- Allow different systems to interoperate
- Eliminate unnecessary duplication of effort
- Encourage useful diversity
- Preserve customer base (for everyone)
- Provide direction to clustering "market"

HOW?

Open Clustering Framework

- Two–pronged approach
  - Define standard cluster APIs
  - Create Component–based Reference implementation
- Both proceed together
- The standards will be a 900–pound penguin…

What if I like locking my customers into my platform?

- Linux has never embraced any proprietary standard
- There *will* be an open standard for clustering
- You can conform to the standard, or be left behind to explain this to your customers
- Your competitors surely will...

Project Structure

API Definition

Select Areas of Interest
Create Subteams
Define APIs
Reach agreement

Reference Implementation

Create Plumbing/Infrastructure
Coordinate with API definition
Define Framework components
Implement components

Standard Clustering APIs

- Neutral (agnostic)
- Royalty–Free
- For OSS or proprietary software
- Creates opportunities for interoperability

API Areas of Interest

- Node services
- Group Services
- Resource Services
- Logging
- DLM
- External Interfaces
  - (GUI, CLI, SNMP, etc.)

Clustering Framework (reference implementation)

- Reference Implementation of APIs
- Component oriented
- Two main divisions:
  - Component implementations
  - Infrastructure / underpinnings / plumbing
- Licensed under the LGPL
- Not mandantory for API conformance

OCF Plumbing

- Provides plugin loading system
- Provides IPC abstraction
- Provides Cluster RPC (built on top of IPC abstraction)
- Provides metadata implementation for configuration information

OCF Components

- Design (external) APIs
- Create Component Definitions
- Design Component–APIs
- Implement Components

- Multiple implementations of components are encouraged
  - Don't fork the framework – fork the components

Open Clustering Framework - cont'd

- First presented at the July 2001 Ottawa Linux Symposium HA working group
- Follow up 2.5 day working group meeting at Linux–Kongress in Enschede, NL
- Currently Endorsed by Linux–HA, FailSafe, Kimberlite, COMPAQ, SuSE, Red Hat, Conectiva, High–Availability.com, etc.
- Will include both HA and HP features
- Linux–HA (heartbeat) is evolving in this direction

Current Status

- Mailing list created for discussions
- Defining groundrules, charter
- BOF held at LWCE –Jan 30,2002
- Discussing affiliation with FSG
- API discussions have begun in earnest in two areas
  - Membership APIs
  - Resource agents

Near-Term Plans

- Focus on a limited number of areas
- Review, refine, implement
- Draft standard APIs for these areas…
- Expand to new areas…
- Repeat until done…

How To Participate?

- Announce your support for the OCF
- Encourage your suppliers, customers to support the OCF
- Join the mailing list
- Commit staff to help create standards
- Join a subteam
- Commit staff to implement, test ref. model
- Evolve your product to conform to the APIs

Conclusions

- Much confusion currently exists in Linux clustering
- The Open Cluster Framework will provide some structure and order
- This has the potential of making Linux the strongest clustering environment available anywhere

References

- http://opencf.org/
- http://linux–ha.org/framework/
- http://linux–ha.org/

Alan Robertson *alanr@unix.sh*
http://linux-ha.org/