IML Roadmap and Development Community

Joe Grund
IML Team Lead
joe.grund@intel.com
Agenda

• IML Background / Overview
• Roadmap
  • 4.1 Release
  • 4.x Series
• Contributing to IML
Background

- Intel® Manager for Lustre (IML) is an open source suite of tools for deploying, managing, and monitoring Lustre.

- IML simplifies Lustre administration with intuitive interfaces and near real-time feedback.

- Works with new and existing Lustre installations.

- Monitors performance and system health.
Deployment

- Deploy Lustre filesystems from one centralized location
- Near-realtime feedback
- Bring filesystem online from first principles or deploy monitoring for an existing filesystem
- Deploy specialized assets, HSM agent node
- Add more assets over time
Management

• Configure / change state of Lustre and related components
• Uses state-machine to reach end state from different starting points
  • Starting LNet, state machine ensures packages are installed + kernel modules loaded before bringing LNet up
• Handle recovery situations fencing, failover
• Uses Corosync / Pacemaker, with PDUs / IPMI
Monitoring

• Holistic system metrics
• Rich visualizations
• Drill into filesystem, target, server
• Find and monitor top jobs
• Aggregate logs across cluster
• HSM Copytool activity monitoring
• Alerts to cluster issues
• GUI / Email / API
• Searchable command / event / alert log / history
Development History

**Chroma 1.0 – 2012**
- Bare Metal install
- Start / Stop / Add / Remove FS and targets
- GUI, REST, CLI interfaces
- HA over Lustre targets

**IEEL 1.0 – 2013**
- Chroma renamed to IML

**IML 2.1 – 2014**
- Heatmap visualization.
- Job stats collection and display
- Hierarchical Storage Management (HSM) support and display
- Custom profile Support
- Near realtime support

**IML 2.2 – 2015**
- ZFS Monitored mode support
- Enhanced parallel server deployment
- Enhanced command display / drilldown

**IML 3.0 – 2016**
- Pacemaker / Corosync config
- NID config
- Queryable status
- Architectural improvements

**IML 3.1 - 2016**
- ZFS support for managed mode.
- Near realtime jobstats monitoring
- tree-view

**IML 4.0 - 2017**
- First open source release
- Tracks Lustre LTS
- appliance -> services
Roadmap
4.1 Roadmap (Active Development)

• Increase scalability of device detection
• Increase modularity of components
• Start moving towards separate management / monitoring code paths
• Targeting GA Q2 2018
Scope

- **libzfs integration**
- **ZED integration**
  - [https://github.com/intel-hpdd/intel-manager-for-lustre/issues/536](https://github.com/intel-hpdd/intel-manager-for-lustre/issues/536)
- **Reactive Architecture**
  - [https://github.com/intel-hpdd/intel-manager-for-lustre/issues/533](https://github.com/intel-hpdd/intel-manager-for-lustre/issues/533)
- **Full Modularity**
  - [https://github.com/intel-hpdd/intel-manager-for-lustre/issues/534](https://github.com/intel-hpdd/intel-manager-for-lustre/issues/534)
libzfs integration

- IML looking to utilize libzfs in 4.1 release for multiple purposes
- Lower level interface over invoking commands / parsing CLI output
- Fine grained collection of pools / datasets / props / VDEV tree
  - easy to collect more data later
- Useful for monitoring + upcoming management enhancements
ZED Integration

• IML looking to use ZED in 4.1 release for multiple purposes
  • Discovery of pool / dataset / property / VDEV changes
    • This is currently a manual scanning process
    • Will allow for closer to realtime changes to propagate
    • Better scaling
  • Surfacing alerts in the IML GUI / API
    • Alerting on critical events across a cluster
    • Searchable history of all events across a cluster
ZED Integration

node-libzfs -> device-scanner

device-scanner -> Events

Events -> Zed Event

Zed Event -> device-scanner

device-scanner -> Zed Event

Zed Event -> Zed Event
Reactive Architecture

- IML looking to flip device discovery from push to pull
  - Adds further scalability
  - Has lower resource usage
  - More responsive
  - Previous iteration used polling + serial introspection of devices
Reactive Architecture

Manager

Agent

Trigger Poll

Result

Serial query of Blockdevices / ZFS / Mounts

Trigger Poll

Result

Serial query of Blockdevices / ZFS / Mounts

Trigger Poll

Result

Serial query of Blockdevices / ZFS / Mounts
Reactive Architecture

Manager

device-scanner

Events

Device tree

UDev Event

Device tree

ZED Event

Mount Change

Device tree

Device tree

Device tree
Reactive Architecture

UDev run script → Unix domain socket → device-scanner → Unix domain socket → consumer process

Findmnt poll

ZEDLET
Modularity

- IML looking to deliver itself completely via RPM
  - No more tarball
  - Ship everything via Fedora Copr
- Benefits
  - Semver minor updates via yum update
  - Components evolve independently
- Those not wanting continuous deployment can disable upgrades / perform offline install / upgrade.
4.x Series Roadmap

- Proposed enhancements post 4.1
- Items may be scoped / de-scoped by priority and community feedback
- Prioritize before starting next release
  - IML 4.2 prioritization process to start in coming weeks
- Target quarterly IML enhancement releases
- As needed compatibility + bugfix releases
Enhanced Deployment

• IML should make it even easier to setup Lustre
  • Deploy to large scale clusters with minimal operator intervention
    • Describe ideal cluster state
    • Expose variants as scalable UI widgets
    • Deploy installation in parallel with a single click
Timeseries DB

- IML Should make it easy for users to consume + display stats
- Use a popular TSDB to collect stats
  - Enable well known API + integration into 3rd party dashboards
  - Better perf / ability to locate stats separately from main db
Use Database Change Events

• IML should use database change events
  • Reduce overhead
  • Push at manager layer
  • Eliminate long-polling, use change events at DB level
Deprecate Storage Plugin

- IML should support more storage arrays as first class citizens
  - Bring functionality in-house for well known plugins
  - Simplify architecture significantly
  - Tightly integrate storage arrays into all of IML
Hardware Data at rest encryption key management

- IML should allow use of hardware encryption across cluster
  - Allow user to encrypt / decrypt all drives or just subset
  - Ability to store pins in a decentralized store
  - Ability to share encrypted pins between nodes for redundancy + HA
Lustre Snapshot Management

IML should be able to manage Lustre snapshots via GUI

- Schedule snapshots for filesystems at some regular interval
- Ad-hoc snapshot on filesystem(s)
- View / delete previous snapshots
- Rollback to a given snapshot
- Rename a snapshot
- Take snapshot at key points (i.e. Lustre upgrade)
Full ZFS Management

- IML should provide full ZFS management
  - Show the list of all pools and datasets
  - Provide drill-down navigation to elicit more detail on a selected target
  - Show the status of pools and datasets
    - Where imported, mounted, error conditions, configuration
- Management
  - Create zpools / datasets
  - Update volume display to easily setup devices, e.g. RAID-Z
  - JBOD enclosure GUI
I18n Support

• IML should have an easy facility for i18n
  • IML text currently English, but IML is used all over the world
  • Need a way to support non-native English speakers
    • Ideally in a way that allows end-user contribution/translation
      • Modify/contribute *.po files consumed by services

• Help Wanted!
Contributing to IML
How to get started - Contributing

1. Signup for [github.com](https://github.com) account if you don’t have one

2. Navigate to the repo you are interested in under our organization [https://github.com/intel-hpdd/](https://github.com/intel-hpdd/)

3. Fork the repo to your personal Github

4. Clone the desired repository locally
   
   ```
   git clone git@github.com:my-user/GUI.git
   ```
How to get started - Contributing

5. Set upstream to intel-hpdd repo
   
   ```
   git remote add upstream git@github.com:intel-hpdd/GUI.git
   ```

6. Create new working branch
   
   ```
   git checkout -b my-new-branch
   ```

7. Make edits, commit + push
   
   ```
   git commit -s
   git push origin my-new-branch
   ```
How to get started - Contributing

8. Visit your repo in Github GUI, click pull request button

9. Checks will run against CI provider

10. Organization members review, request changes or land
Help Wanted

• Check Github issues for help wanted opportunities
  • https://github.com/issues?
    utf8=%E2%9C%93&q=is%3Aopen+is%3Aissue+archived%3Afalse+user%3Aintel-
hpdd+label%3A%22help+wanted%22

• Easy to implement, team guidance
• Open an issue / submit a PR
  • Use a release train model, pull in work once it’s done
  • Want your feedback on useful enhancements
• Milestones are public
  • https://github.com/intel-hpdd/intel-manager-for-lustre/milestones
Get in Touch

- Many ways to reach out
  - Join us on gitter.im for any questions
    - https://gitter.im/intel-hpdd/intel-manager-for-lustre
  - Request any issues / enhancements on the IML Repo
    - https://github.com/intel-hpdd/intel-manager-for-lustre/issues/new
  - Checkout our docs
    - https://intel-hpdd.github.io/Online-Help/
  - We have extensive docs on contributing
Closing

• IML is a project with a long history and continues advancing
  • IML 4.0 was first open source release in Oct 2017
  • Five maintenance updates to 4.0 release
  • IML 4.1 targeted for Q2 2018
  • IML 4.2, next release; prioritization shortly
• We are actively seeking contributors
  • Open tickets or PRs for enhancements / bugfixes
  • Get involved in IML’s future!