OpenXT:
Toolstack Modernization

Chris Rogers

AIS, Inc.

June 7, 2016

Briefing Outline

Toolstack Modernization

- **Motivation**
- **Development Strategy**
- **Overview of Current Toolstack**
- **Modernization Tasks:**
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- **Current Development Status**

Briefing Outline

Toolstack Modernization

devastating capability, revolutionary advantage

- **Motivation**
- Development Strategy
- Overview of Current Toolstack
- Modernization Tasks:
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- Current Development Status

Motivation

Toolstack Modernization

devastating capability, revolutionary advantage

- Upstream using libXL
- Newer Xen requires libXL
- Reduce technical debt
- Establish the OpenXT ‘Base Platform’

Briefing Outline

Toolstack Modernization

devastating capability, revolutionary advantage

- Motivation
- **Development Strategy**
- Overview of Current Toolstack
- Modernization Tasks:
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- Current Development Status

Divide and Conquer

Toolstack Modernization

devastating capability, revolutionary advantage

- **Non-trivial task**
- **Four main areas:**
  - Communication between Xenmgr and libXL (Chris Rogers)
  - Linux-based stubdomains on libXL (Martin Osterloh)
  - Dm-agent removal (Eric Chanudet)
  - Updating Support Libraries, blktap (Ross Philipson)
- **Work can be done in parallel, until integration time.**

Briefing Outline

Toolstack Modernization

devastating capability, revolutionary advantage

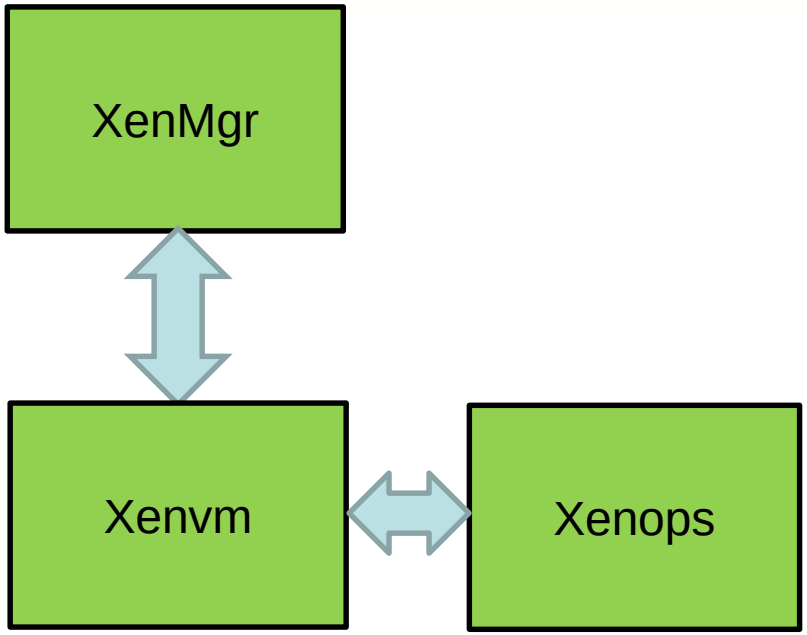
- Motivation
- Development Strategy
- **Overview of Current Toolstack**
- Modernization Tasks:
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- Current Development Status

Current Toolstack

Toolstack Modernization

devastating capability, revolutionary advantage

- **Top Down:**
  - XenMgr
  - Xenvm
  - Xenops
- **Primary mode of communication: RPC (Dbus)**
- **Tightly Coupled**
  - XenMgr functions written to handle Xenvm-specific quirks.
- **Ocaml, Haskell**



Current Toolstack: XenMgr

Toolstack Modernization

devastating capability, revolutionary advantage

- Top-level, domain management piece
- Haskell
- Provides the xec and xec-vm interfaces
- Responsible for domain state
- UI actions -> XenMgr via RPC/Dbus
- Domain config (db, xenstore access)
- Lifecycle ops: start, shutdown, reboot, sleep, etc.

Current Toolstack: Xenvm/Xenops

Toolstack Modernization

- Descendent of xend/xm
- Written to address the limitations of xend/xm for client virtualization
- Ocaml
- RPC server per domain
- Hooks into libxc, libxs
- Target for replacement with libXL

Briefing Outline

Toolstack Modernization

devastating capability, revolutionary advantage

- Motivation
- Development Strategy
- Overview of Current Toolstack
- **Modernization Tasks**:
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- Current Development Status

XenMgr/XL Communication

Toolstack Modernization

devastating capability, revolutionary advantage

- **Majority of protocols are contained within the xenvm.hs module**
  - RPC calls to per-domain xenvm server instance.
- **How to interface with XL?**
  - Implement RPC server in XL?
  - Separate daemon that translates xenvm calls to xl calls?
  - Link libXL library into XenMgr and call directly using FFI?
- **Too much overhead or patching. Simpler solution.**

XenMgr/XL Communication

Toolstack Modernization

- **Replace xenvm.hs with xl.hs**
  - -1, Ick, Haskell.
  - +1, 99% contained here.
  - +1, No need for new agent/daemon
  - +1, Chance to improve/fix some existing protocols
- **Benefits to this approach outweigh the negatives**
- **Interact with XL through the CLI**
  - XenMgr uses ‘system’ function to perform operations: create, shutdown, reboot, etc.
  - XenMgr isn’t responsible for controlling the domain (libXL’s job), mostly it maintains state, handles domain configs, queries information, and read/writes to the xenstore.
  - The CLI is sufficient for these tasks, and doesn’t overcomplicate, such as using the FFI (Calling C from Haskell).

XenMgr/XL Communication

Toolstack Modernization

- **Example of how things are changing:**
- **Xenvm.hs:**

```
acpiState :: Uuid -> Rpc AcpiState
acpiState uuid = catchNoService 5 $ do
  r <- invoke uuid "get_acpi_state" []
  return $ read (unpackStrArg r)
```

- **Xl.hs:**

```
acpiState :: Uuid -> IO AcpiState
acpiState uuid = do
  domid <- getDomainId uuid
  acpi_state <- readProcess "xl" ["acpi-state", show domid] []
  return $ (read acpi_state :: Int)
```

XenMgr/XL Communication

Toolstack Modernization

devastating capability, revolutionary advantage

- **To be clear, xenvm/xenops is being replaced by libXL**
- **From a CLI perspective, ‘xenops list’ is replaced with ‘xl list’**
  - Low impact anyway, most CLI is done through the XenMgr interface.
  - xl CLI is well documented
- **The XenMgr CLI (ie. *xec* and *xec-vm*) is *not* changing.**
  - xec-vm -u <uuid> start
  - xec set enable-dom0-networking true

Briefing Outline

Toolstack Modernization

devastating capability, revolutionary advantage

- Motivation
- Development Strategy
- Overview of Current Toolstack
- **Modernization Tasks**:
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- Current Development Status

Linux Stubdomains

Toolstack Modernization

devastating capability, revolutionary advantage

- **What are Stubdomains and why use them?**
  - Domain specifically to house the device-model (QEMU) for a guest domain.
  - Disaggregation for security
  - Run the device model in a separate, non-privileged domain
  - If the device model is compromised, privilege escalation is limited to the stubdomain.
- **Currently in use on the OpenXT platform**
  - Extensive support in XenMgr, xenvm, dm-agent, and other helpers.
- **Not natively supported on libXL**
  - Upstream support for MiniOS only.

Linux Stubdomains

Toolstack Modernization

- **Work by Anthony Perrard[1] and Eric Shelton[2]**
  - Certain assumptions made then have changed today
  - How to adopt this work for the OpenXT platform?
- **Of the seven existing patches, only four are needed**
  - Don’t need to compile QEMU, kernel, or build a disk image
  - OpenXT uses initramfs
  - Some limitations such as Video Output, Save/Restore, and QMP Connections.
  - Development hurdles: Async Operations
- **Upstreaming Linux stubdom support**
  - Once support in OpenXT is stable, would like to work with xen-devel and the Xen community.

[1] https://blogspot.xenproject.org/2012/12/12/linux-stub-domain/
[2] http://lists.xen.org/archives/html/xen-devel/2015-02/msg00421.html

Briefing Outline

Toolstack Modernization

devastating capability, revolutionary advantage

- Motivation
- Development Strategy
- Overview of Current Toolstack
- **Modernization Tasks**:
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- Current Development Status

DM-Agent Removal

Toolstack Modernization

devastating capability, revolutionary advantage

- **What is dm-agent?**
  - Daemon to handle disaggregated backends and emulation for HVM and PV-on-HVM guests.
  - Communicates through the Xenstore.
  - Toolstack processes advertised capabilities, then writes appropriate Xenstore nodes to setup device-models.
- **Has its own protocol: dmbus**
  - Specifically meant for device-model related RPC across domains
  - Low-overhead protocol, alternative to using v4v.
- **Problem with dm-agent**
  - Introduces yet another communication protocol, but doesn't satisfy disaggregation.
  - Conflicts with upstream toolstacks – specifically, how qemu args are generated.

DM-Agent Removal

Toolstack Modernization

devastating capability, revolutionary advantage

- **dm-agent removal in phases**
  - Reduce its role in dom0-to-dom0 communication
  - Surfman and Inputserver now implement dbus calls
- **Removal from stubdoms will be slightly more challenging**
  - Move dm-agent tasks into libXL
  - Likely continue using Xenstore, but no more dm-agent daemon in stubdoms
- **Major benefits:**
  - Dbus for intra-domain, and V4V for inter-domain
  - Removes dependencies in surfman, inputserver, and stubdoms
  - One less daemon

Briefing Outline

Toolstack Modernization

devastating capability, revolutionary advantage

- Motivation
- Development Strategy
- Overview of Current Toolstack
- **Modernization Tasks**:
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- Current Development Status

Updating Support Libraries

Toolstack Modernization

- **QMP-Helper**
  - Moving to libXL also means providing more support for qmp.
  - qmp server in dom0 is trivial, but not in stubdoms
- **What is QMP-Helper?**
  - chardrv implementation in qemu-char.c, routing traffic over V4V
  - Proxy process in dom0 called *qmp_helper*, provides unix socket for libXL.
  - Talks with the chardrv in the stubdomain, V4V translation happens invisibly to libXL.
- **Why QMP-Helper?**
  - Removed dm-agent, why introduce another daemon?
  - qmp is a requirement now to handle sleep and resume on hvm.
  - qmp-helper lays the foundation for further utilizing qmp in the future.

Briefing Outline

Toolstack Modernization

devastating capability, revolutionary advantage

- Motivation
- Development Strategy
- Overview of Current Toolstack
- Modernization Tasks:
  - XenMgr <-> libXL
  - Linux stubdomains
  - Dm-agent removal
  - Updating Support Libraries
- **Current Development Status**

Current Development

Toolstack Modernization

devastating capability, revolutionary advantage

- **XenMgr -> libXL Communication:**
  - Xl.hs module is mostly written, XenMgr changes to support changes are written.
- **Linux Stubdoms on XL:**
  - Linux stubdom patches are integrated
  - Stubdoms booting
- **DM-Agent Removal:**
  - Surfman and Inputserver dependencies are gone
  - Last step is to remove dm-agent from the stubdomain
- **Support Libraries:**
  - qmp-helper is written, needs to be integrated into libXL (-qmp option for qemu args)

Current Development

Toolstack Modernization

devastating capability, revolutionary advantage

- What does this mean, and what's left to do?
- OpenXT running XL with HVM and PV-on-HVM guests, uivm, ndvm, domain management through XenMgr.
- Stubdoms very close, xsm and selinux policies still need to be written.
- Lots of regression testing.

End

Toolstack Modernization

devastating capability, revolutionary advantage

- **Thanks for listening!**