Windsor

Domain 0 disaggregation for XenServer

James Bulpin, Director of Technology, XenServer, Citrix

August 28, 2012

August 27-28, 2012
San Diego, CA, USA

Introduction

- XenServer/XenCloudPlatform (XCP): a distribution of Xen, a domain 0 and everything else needed to create a platform for virtualization
- A platform for server virtualisation
- A platform for virtual desktops (e.g. XenDesktop)
- A platform for the cloud (e.g. OpenStack and CloudStack)
- A platform for virtualised networking (e.g. NetScaler)
- All use cases are tending towards much higher numbers of guest VMs per host
- Current architecture works now but will hit bottlenecks as servers get bigger and more powerful
- Want a flexible, modular solution to scalability

Overview

- XenServer architecture evolution – a brief history
- Limitations of the current architecture
- Windsor: 3rd generation XenServer architecture
  - Using domain 0 disaggregation for scalability and performance

This presentation is about the internal technology of XenServer/XCP. It is not a statement about the future feature set or capabilities of any particular XenServer release.

XenServer Architecture – a brief history

XenServer Architecture – a brief history (1)

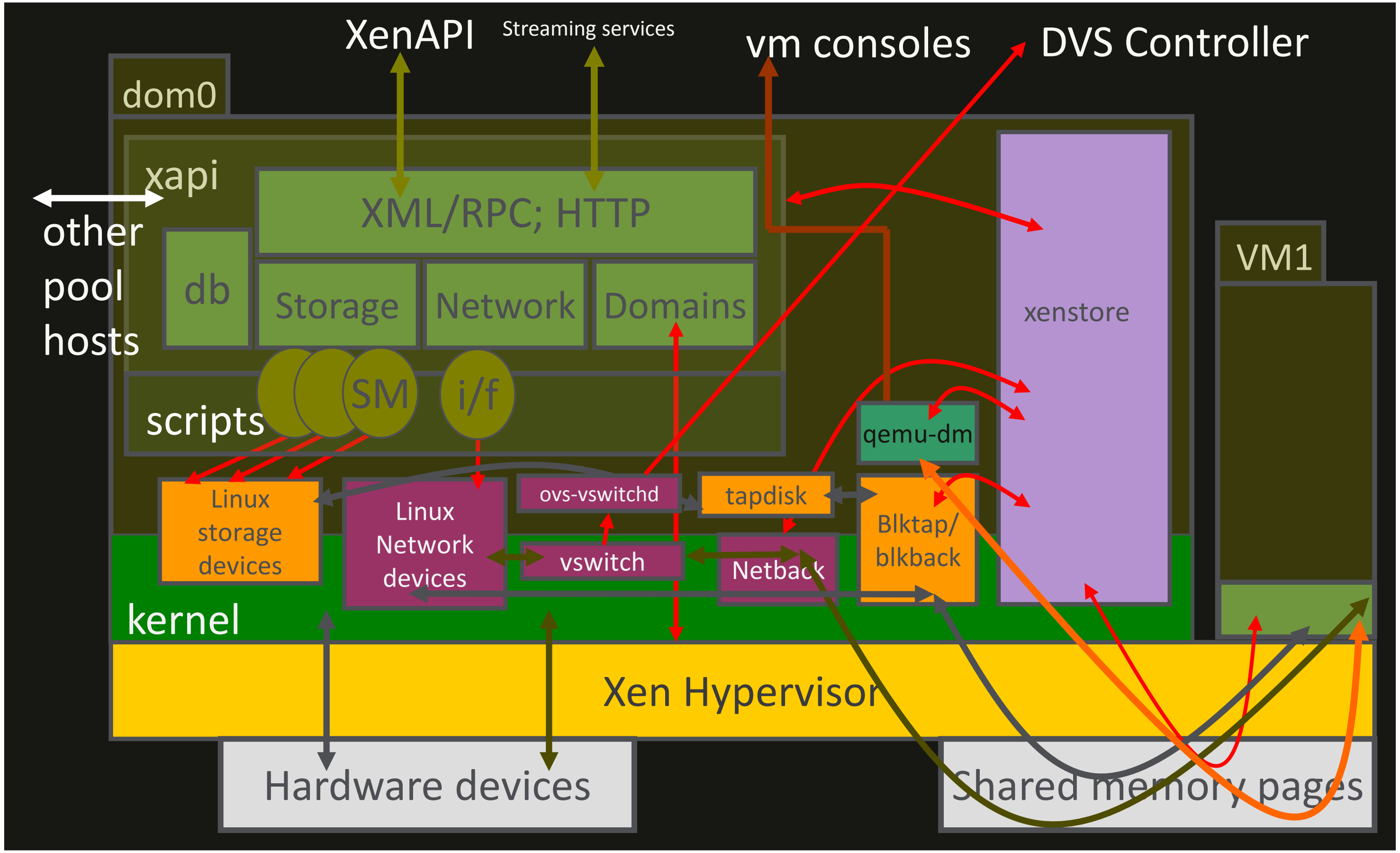
- First generation architecture in *Burbank* – XenEnterprise 3.0.0
- Single host virtualisation: no resource pools, no XenMotion
- Based on open-source xend/xm toolstack
- Basic C host agent driving xend
- XenAdmin Java management application
- Initially used a small, read-only dom0
    - Moved to writable environment in 3.1

XenServer Architecture – a brief history (2)

- Second generation architecture in *Rio* – XenServer 4.0.0
- Current releases, including XenServer 6.1 coming soon, based on this
- All current versions of XCP based on this
- Sophisticated Ocaml xapi toolstack implementing the XenAPI
  - Uses only lowest level part of open-source Xen toolstack (libxc)
- Clustered architecture for resource pools, XenMotion and master mobility
- Domain 0 evolved from 1st generation
  - Fairly full Linux distribution based on CentOS
  - Writable environment using RPMs for package management

Limitations of the current architecture

XenServer 2nd generation host architecture



CiTRIX®

Limitations of the current architecture

- With future larger, powerful servers domain 0 will become a bottleneck
    - Will limit scalability and performance
- Lack of failure containment
- Hard to reason about dom0 security
- Non-optimal use of modern NUMA architectures
- Limited third party extensibility

“Windsor” – XenServer’s new architecture

Goals for the new architecture

- Improved scalability on single XenServer host
- Remove performance bottlenecks
- Cloud scale horizontal scaling of XenServer hosts
- Better isolation for multi-tenancy
- Increased availability and quality of service
- Higher Trusted Computing Base measurement coverage

Design principles and overview

- *Scale-out*, not scale-up – exploit parallelism and locality
- *Disaggregate* Domain-0 functionality to other domains
- Encapsulate and simplify
- Clean APIs between components
- Flexibility
- Design for scalability and security

The approach

- Can we improve performance and scalability with a bigger domain 0? – Yes
- Can we tune domain 0 to better utilise hardware resource? – Yes
- But... this makes domain 0 a bigger and more complex environment
  - Easy to change one thing and have a negative effect elsewhere
  - Multiple individuals and organisations working in the same complex environment
  - Still don’t get containment of failures
- Instead *disaggregate* domain 0 functionality into multiple system domains
  - Can be thought of as “virtualising dom0” – after all we tell users that it’s better to have one workload per VM and use a hypervisor to run multiple VMs per server
  - Helps disentangle complex behaviour
  - Easier to provision suitable resources, allows for clear parallelism
  - Contains failures

Domain 0 disaggregation

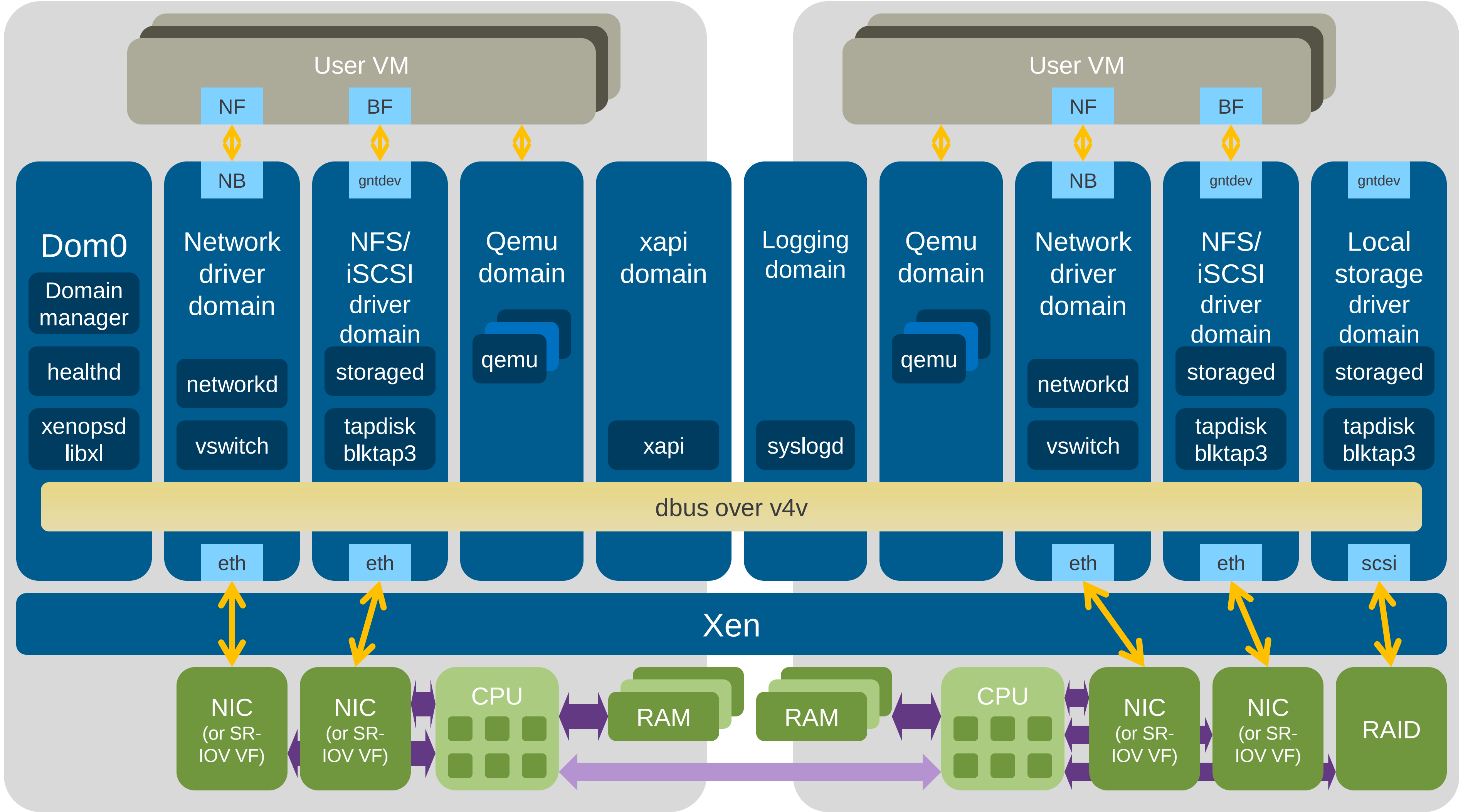
- Moving virtualisation functions out of domain 0 into other domains
  - Driver domains contain backend drivers (netback, blkback) and physical device drivers
    - Use PCI pass-through to give access to physical device to be virtualised
  - System domains to provide shared services such as logging
  - Domains for running qemu(s) – per VM *stub* domains or shared qemu domains
  - Monitoring, management and interface domains
- Has been around for a while, mostly with a security focus
  - Improving Xen Security through Disaggregation (Murray *et. al.*, VEE08) [http://www.cl.cam.ac.uk/~dgm36/publications/2008-murray2008improving.pdf]
  - Breaking up is hard to do: *Xoar* (Colp *et. al.*, SOSP11) [http://www.cs.ubc.ca/~andy/papers/xoar-sosp-final.pdf]
  - Qubes (driver domains) [http://qubes-os.org]
  - Citrix XenClient XT (network driver domains) [http://www.citrix.com/xenclient]

Targets for disaggregation

- Storage driver domains – storage stack (e.g. VHD), ring backends, device drivers
- Network driver domains – network stack (e.g. OVS), ring backends, device drivers
- xenstored domain – busy, central control database for low level functionality
- xapi management domain
- qemu domain (per node, per tenant or per-VM) – emulated BIOS etc.

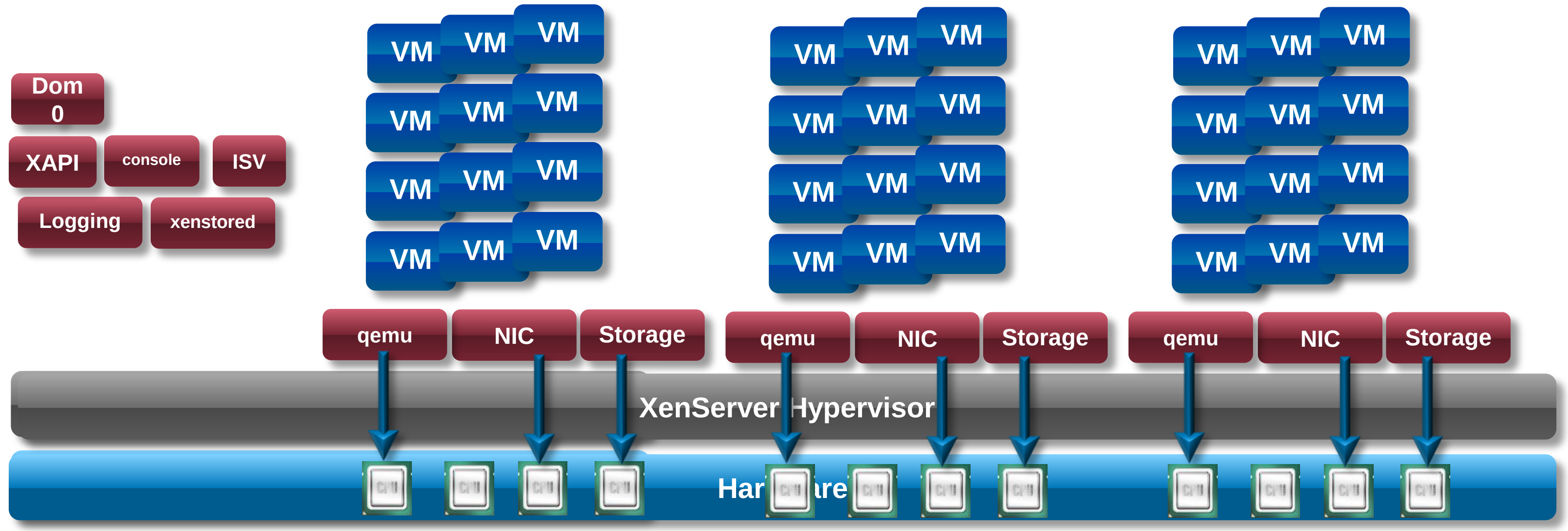
Benefits

- Better VM density
- Better use of scale-out hardware (NUMA, many cores, balanced I/O)
- Improved stability
- Improved availability (fault isolation)
- Opportunity for secure boot etc.
- More extensible, future value-add opportunities

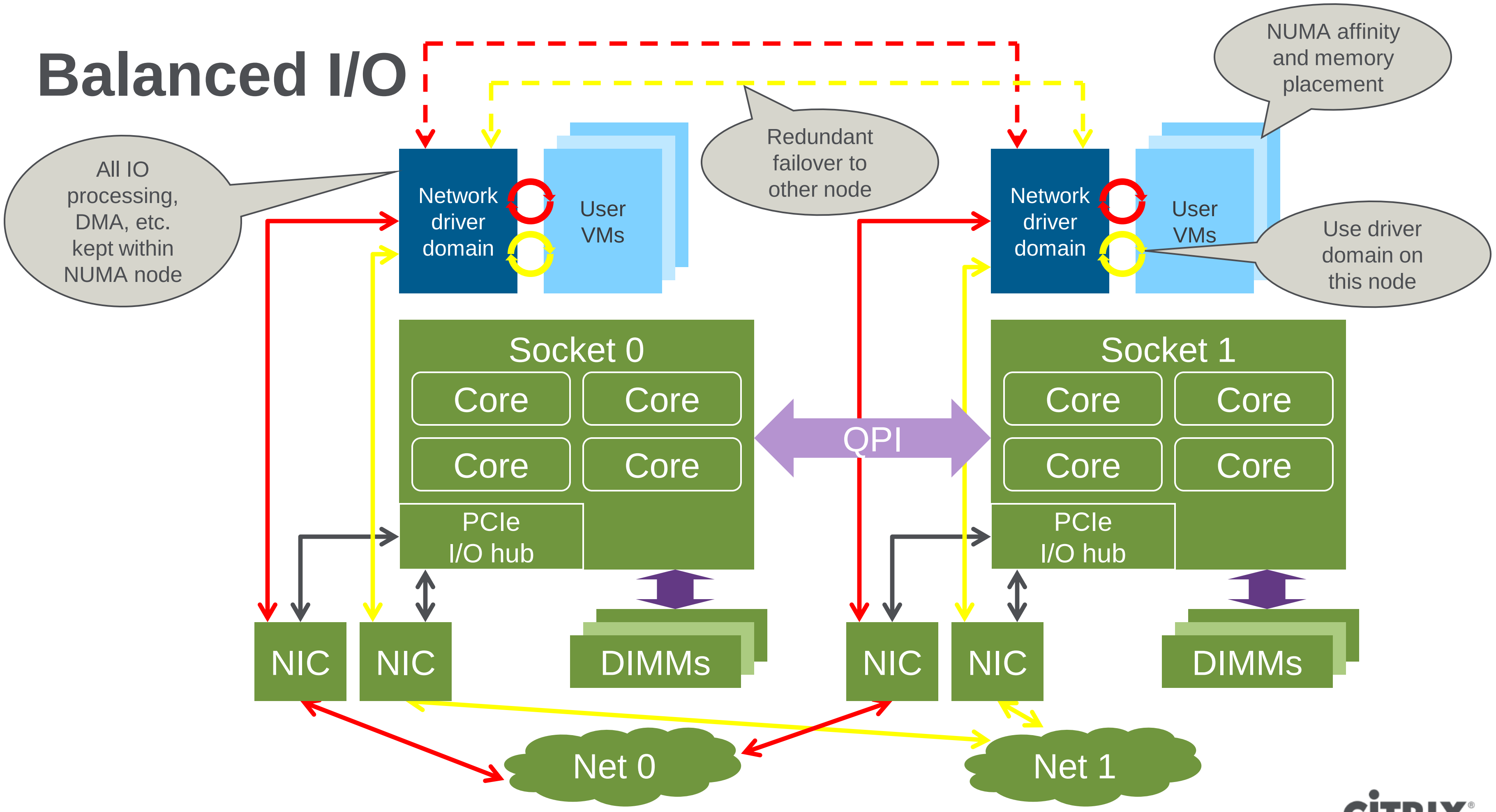

User VM
NF
BF
NB
gntdev
Dom0
Domain manager
healthd
xenopsd libxl
Network driver domain
networkd
vswitch
NFS/ iSCSI driver domain
storaged
tapdisk blktap3
Qemu domain
qemu
xapi domain
xapi
Logging domain
syslogd
User VM
NF
BF
Qemu domain
qemu
NB
Network driver domain
networkd
vswitch
gntdev
NFS/ iSCSI driver domain
storaged
tapdisk blktap3
gntdev
Local storage driver domain
storaged
tapdisk blktap3
dbus over v4v
eth
eth
eth
eth
scsi
Xen
NIC (or SR-IOV VF)
NIC (or SR-IOV VF)
CPU
RAM
RAM
CPU
NIC (or SR-IOV VF)
NIC (or SR-IOV VF)
RAID

Replication to scale-out on big servers

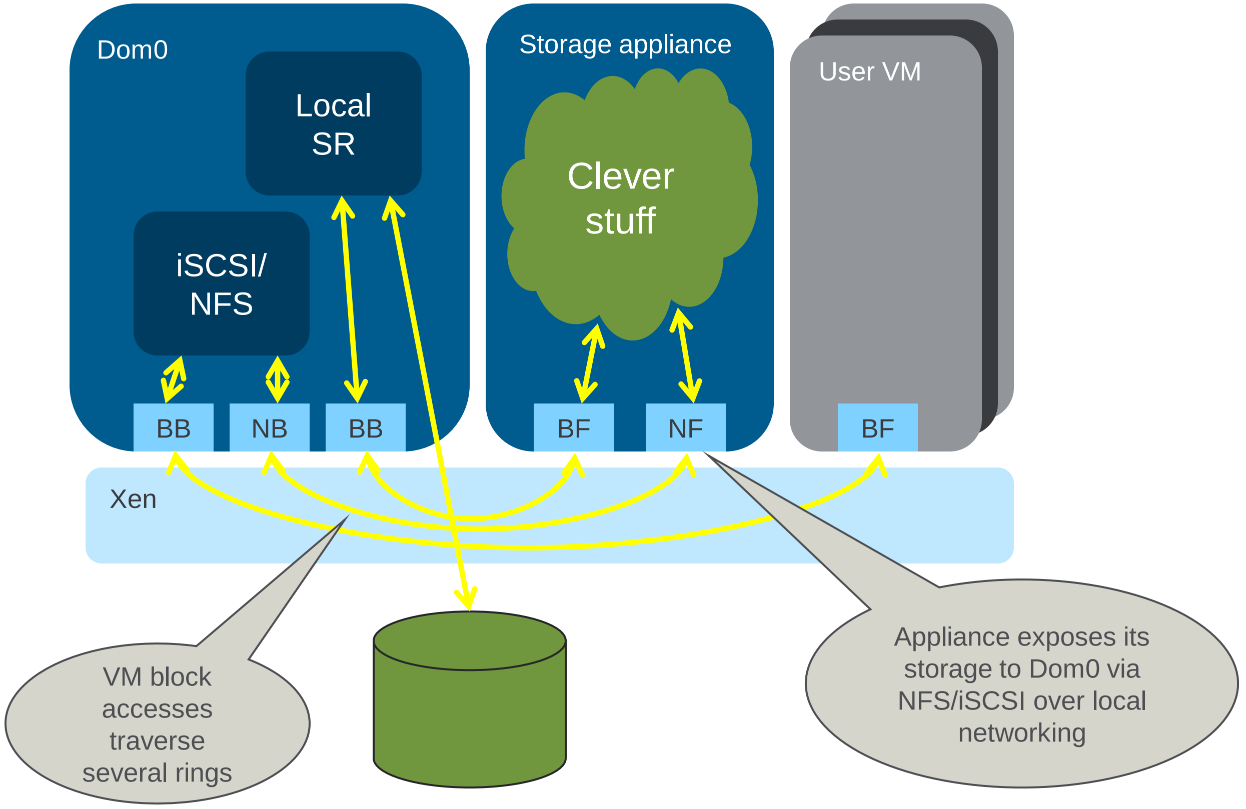
- Scalability
  - More VMs, more system domains
  - Isolation for cloud environments



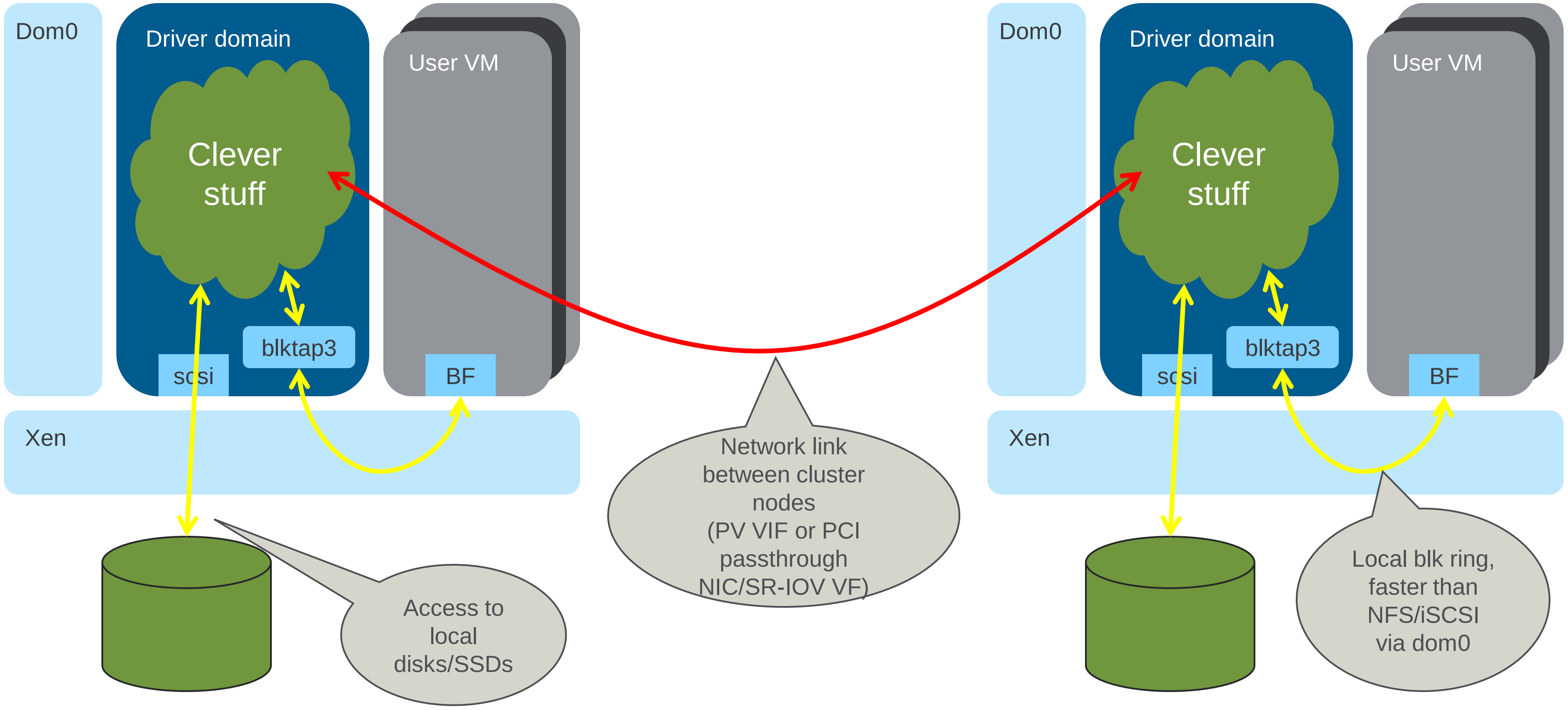
Balanced I/O


All IO processing, DMA, etc. kept within NUMA node
Network driver domain
User VMs
Redundant failover to other node
Network driver domain
User VMs
NUMA affinity and memory placement
Use driver domain on this node
Socket 0
Core
Core
Core
Core
PCIe I/O hub
QPI
Socket 1
Core
Core
Core
Core
PCIe I/O hub
NIC
NIC
DIMMs
NIC
NIC
DIMMs
Net 0
Net 1

Flexibility – value-add storage appliances (today)


Dom0
Local SR
iSCSI/ NFS
BB
NB
BB
Storage appliance
Clever stuff
BF
NF
User VM
BF
Xen
VM block accesses traverse several rings
Appliance exposes its storage to Dom0 via NFS/iSCSI over local networking

Flexibility – value-add storage appliances (Windsor)


Dom0
Driver domain
User VM
Clever stuff
blktap3
scsi
BF
Xen
Access to local disks/SSDs
Network link between cluster nodes (PV VIF or PCI passthrough NIC/SR-IOV VF)
Dom0
Driver domain
User VM
Clever stuff
blktap3
scsi
BF
Xen
Local blk ring, faster than NFS/iSCSI via dom0

Summary

- Next generation XenServer architecture built to scale-out using domain 0 disaggregation techniques
  - Scale with the workload and server size
  - Expect to significantly enhance scalability and aggregate performance
- Well defined APIs between components will allow better extensibility
- Avoids complexity of single, large, multi-function domain 0
  - Easier to reason about
  - Easier to maintain and debug
  - Containment of failures

CiTRIX®

Work better. Live better.