Virtualisation and Cloud Services

Colin Butcher
XDelta Limited

www.xdelta.co.uk
+44 117 904 8209

xdelta

Personal background

- Systems architect specialising in mission critical systems
- Engineering background (printing presses, nuclear reactors, power generation)
- Wide range of experience (power generation and distribution, satellite control centres, air traffic monitoring, finance data, healthcare, transport, etc.)
- Started XDelta in 1996

xdelta

XDelta – what we do

- Lead mission-critical systems projects
- Deliver world class services in demanding environments
- Strategic planning, technical leadership and project direction with clarity of vision and an eye for detail
- Systems engineering for availability and performance
- Ensure long term success through skills transfer

xdelta

Agenda

- Concepts and principles
- Physical infrastructure
- Virtualisation
- Cloud services
- Discussion

xdelta

Part 1 - principles

- Abstraction layers
- Performance characteristics
- Parallelism
- Scalability

Abstraction layers

"All problems in computing can be solved by introducing another layer of abstraction."

"Most problems in computing are caused by too many layers of complexity."

We need to strike a balance that is appropriate for the kinds of systems we're building.

xdelta

Using abstraction layers

- What looks like your dedicated resource is just a slice of a much bigger thing over which you have no control:
  - What looks like a CPU isn't a CPU
  - What looks like memory isn't all of memory
  - What looks like a disc isn't a disc
  - What looks like a network isn't the whole network

The “Hall of mirrors”

- You can’t see everything that’s going on
- The view is often distorted
- Hiding things makes it easier to deal with the bits you’re interested in
- Hiding things makes it much harder to understand a problem, especially performance

Parallelism

Understand how your workload could break down into parallel streams of execution:

- Some will be capable of being split into many small elements with little interaction
- Some will require very high levels of interconnectivity and interaction
- Some will require high-throughput single-stream processing

xdelta

Performance characteristics

- Bandwidth – determines throughput
  - It's not just "speed", it's throughput in terms of "units of stuff per second"
- Latency – determines response time
  - Determines how much "stuff" is in transit through the system at any given instant
  - "Stuff in transit" is the data at risk if there is a failure
- Jitter ("div latency" or variation of latency with respect to time) – determines predictability of response
  - Understanding jitter is important for establishing timeout values
  - Latency fluctuations can cause system failures under peak load

xdelta

Part 2 – physical infrastructure

- Data networks
- Storage arrays and SANs
- Systems

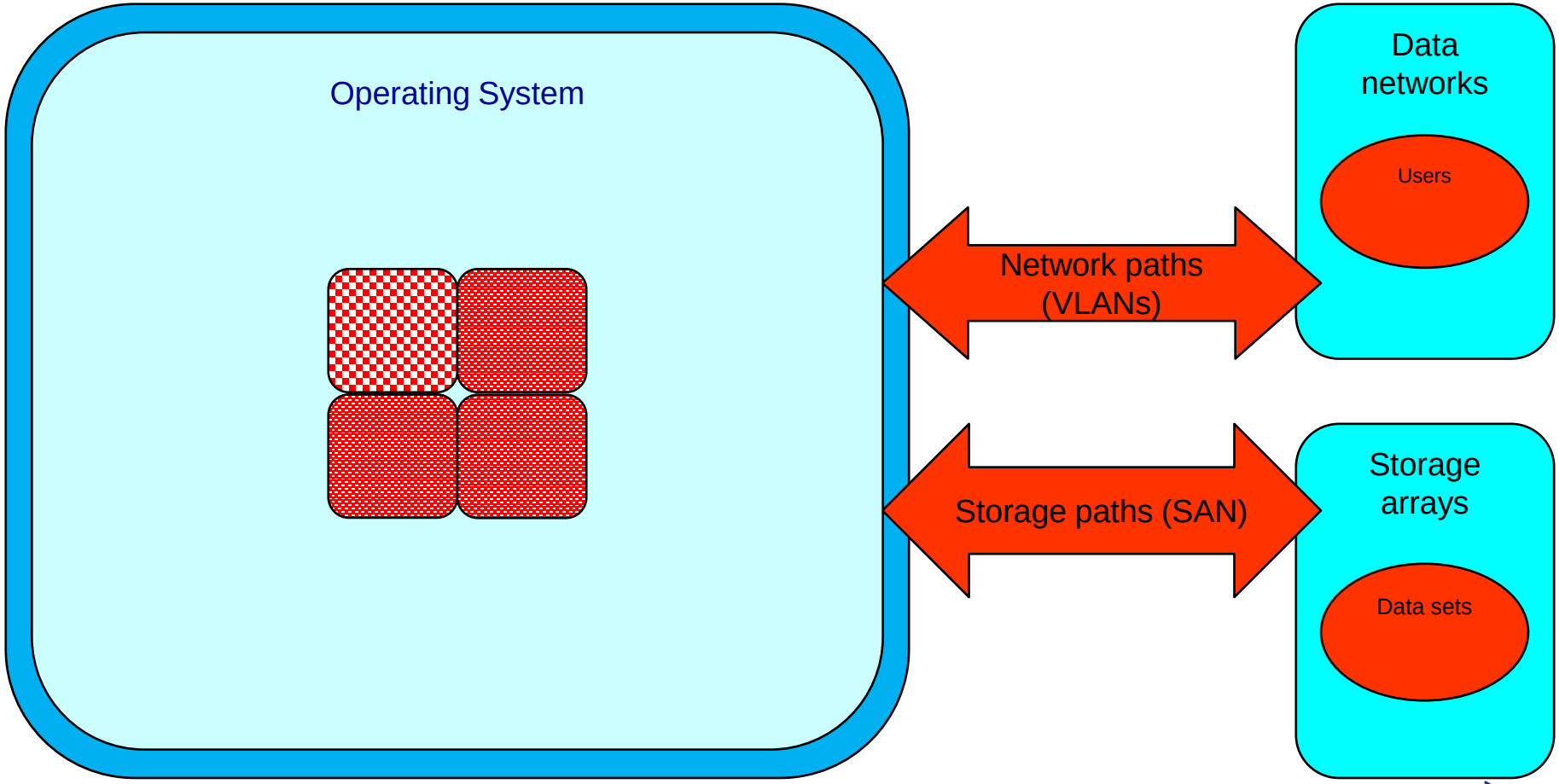
xdelta

Components of a computer system

Systems store data, process data and exchange data with other systems and users:

- CPUs do processing
- Memory holds data and instructions
- Storage subsystems let us store and retrieve data
- Data networks let us communicate

Conventional server computing



xdelta

Data networks

- Core switches
- Edge switches
- Availability
- Traffic management
- Firewalls
- Load balancers

xdelta

Storage networks

- iSCSI – uses data networking
- Fibrechannel – designed for storage networking
- SANs
- Storage arrays

Hardware platforms (1)

Blade technology brings virtualisation of the system infrastructure (chassis components):

- Virtual connections from processing components over backplane channels
- Modular systems provide great flexibility of configuration and interchangeability of components

Hardware platforms (2)

- Big high-end multiprocessor systems are needed for certain workloads
- Stand-alone systems will still be needed in highly secure or mission-critical / safety-critical environments
- Current trend is high core count / large memory machines

Capacity and scalability

- Increasing the capacity of the overall system:
  - “Scale up” or “vertical scaling” refers to increasing capacity by adding more resources to a machine or buying a bigger machine (CPU count, memory, I/O adapters, etc.)
  - “Scale out” or “horizontal scaling” refers to increasing capacity by adding more machines (eg: blades)
  - It depends on how your workloads break down into parallel streams of execution and on what level of availability you need to achieve

Part 3 - Virtualisation

- Data networks
- Storage arrays and SANs
- Systems
- IO flow

xdelta

What do we mean by Virtualisation?

"When I use a word, it means just what I choose it to mean - neither more nor less."

Humpty Dumpty, Through the Looking Glass,
by Lewis Carroll.

Virtual = .NOT. Physical

xdelta

Virtualisation - data networks

- VLANs are used to segment a data network:
  - Implemented by using 802.1Q tagging of packets
  - Systems can behave as if they are switches and send tagged packets for multiple VLANs over the same NIC
  - Switch configurations generally map Layer 3 IP V4 subnets to Layer 2 VLANs and enforce IP routing between VLANs
  - Extended VLANs can span multiple sites
- “Software defined networking” - OpenStack

xdelta

Virtualisation – SAN zoning and VSANs

- Implemented within SAN switches (directors)
- Zoning and VSANs – unlike data networks, nothing connects by default
- Extended SAN fabrics spanning multiple sites
- Fibrechannel and iSCSI behave differently

Virtualisation – storage arrays

- Array controllers cache much of the data, using protected (battery or flash memory) write-back mirrored cache
- Array controller “hides” the behaviour of the physical discs and distributes the IO load across multiple spindles.
- Performance issues:
  - bandwidth to and from the array controller pair
  - contention by systems for access to the storage array
  - controller processing overheads (eg: RAID 0+1 v RAID 6)

xdelta

Virtualisation – virtual tape libraries

- Additional firmware acts as a “front end” to a storage array and presents what appears to be a set of tape drives and media robots to the systems

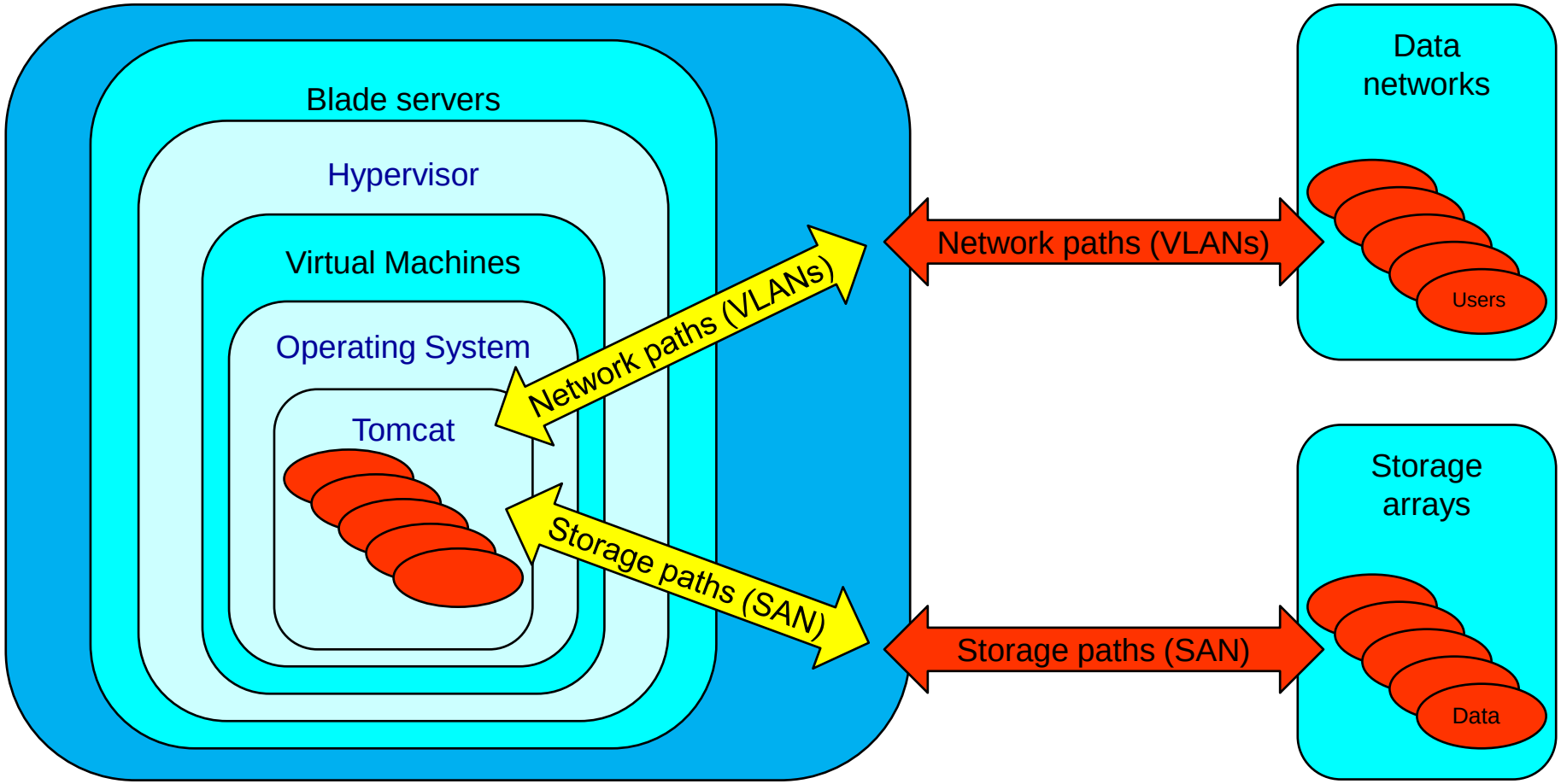
- The storage array is typically populated with high capacity spindles where capacity and throughput are more important that random access performance (ie: FATA not FCAL)

Virtualisation of processing (CPU)

- Improved utilisation of hardware resources
- Improved high availability and disaster tolerance
- Parallelisation of processing
- Multiple run-time environments
- Minimise hardware dependencies

xdelta

Virtual server computing



xdelta

Workloads in a virtual world

- To the “hypervisor”, each and every virtual machine is a workload needing physical hardware resources
- Within a virtual machine, each application (and the operating system overhead) is a workload
- What level of interaction is there between the virtual machines that run your applications ?
- What happens when your host hardware runs out of resources or when virtual machines move to another hardware host ?

xdelta

Virtual machines

- CPU – processing capability
- Memory – stores data and instructions ready for use
- IO – moves data into / out of memory and communicates with other systems
- Console access

Virtual machine creation

- The VM is a multi-threaded application running under the control of the hypervisor
- CPU scheduling in the guest OS maps to threads scheduled by the hypervisor
- Physical memory in the guest OS maps to virtual memory in the hypervisor – lots of physical memory in the hardware platform is a good thing

Virtual machine devices

- Devices presented to the operating system running inside the VM have to ‘map through’ to devices managed by the hypervisor:

- Storage devices typically map to container files, not physical hardware

- Network devices typically map to 802.1Q tagged VLANs using a “virtual switch” in the hypervisor that interconnects the virtual system NICs to each other and to the physical NICs visible to the hypervisor

xdelta

Native instruction set

- CPU virtualisation provides a set of guest machine environments for a native machine running under a ‘hypervisor’ on the base hardware
- The ‘hypervisor’ allocates physical machine resources to the guest machine environments
- The booted operating system running in a guest machine environment co-operates with the hypervisor at device driver level with purpose-written drivers

xdelta

IO flow in a virtual system

- Operating system device support for running under a hypervisor requires purpose written device drivers on the ‘inside’ of the virtual machine to communicate with the way the hypervisor represents a device to the ‘outside’ of the virtual machine

- Device drivers in virtual machines communicate with host system (hypervisor) device representations

- Host system (hypervisor) device representations communicate with physical devices

IO flow – data networking

- Application to OS kernel I/O services
- OS kernel I/O services to OS device driver
- OS device driver to VM side of presented device

- Hypervisor side of presented device to virtual port on "virtual switch" (either tagged or untagged) and tags packets at entry to the virtual switch
- Hypervisor virtual switch passes tagged packets to:
  - Other virtual switch ports (for internal I/O)
  - hypervisor NICs (for external I/O)

IO flow - storage

- Application to OS kernel I/O services
- OS kernel I/O services to OS device driver
- OS device driver to VM side of presented device

- Hypervisor side of presented device to hypervisor device representation
- Hypervisor device representation to container file pseudo-device, which maps block numbers to blocks within the container file
- Container file pseudo-device driver passes I/O to physical storage device

xdelta

IO performance v CPU performance

- Physical I/O operations typically takes a few milliseconds to complete
- We can execute a lot of CPU instruction cycles in a few milliseconds (1 GHz = 1 nanosecond, thus 1 million instruction cycles per millisecond)!
- However, all that extra code to present a device from the hypervisor to the virtual machine is system overhead, not useful work

Improving IO performance

- Para-virtualisation allows a virtual system to directly and exclusively use an external (outside the VM) physical device, bypassing the hypervisor as far as possible

- Multi-core processors with large on-chip caches can help by offloading system overhead code to other cores

- VM instance to VM instance I/O (eg: network traffic between two virtual servers through the virtual switch) on the same hardware host machine will generate a lot of system overhead

Virtual machines - summary

- The end to end I/O path is extremely complex, especially in a blade environment
- Think about how the various applications interact and thus which VMs to place on which hardware platforms
- Think about how to perform maintenance and support activities with minimal disruption to service
- Think about how to architect and monitor the entire infrastructure

xdelta

Part 4 – cloud services

- The deployment problem
- Infrastructure as a service
- Platform as a service
- Software as a service

The deployment problem

- Putting systems into service takes time and is expensive
- Monitoring and managing everything is complex
- Do we need to know about all the underlying “stuff” when all we want to do is run some applications to do something ?
- Users (“clients”) connect to what they need

Cloud services

- A “management wrapper” around the underlying systems
- Automates and simplifies the provision of resources:
  - Infrastructure as a service
  - Platform as a service
  - Software as a service

Infrastructure as a service

- We get the resources we ask for as long as we need them
- All the hardware related stuff is done for us
- We look after the operating systems, patching, applications, data, etc.

Platform as a service

- We don’t see the physical hardware
- We get the resources we ask for as long as we need them
- All the tedious hardware and operating systems related stuff is done for us
- It’s a step change from running systems to delivering a service
- We look after our data and our applications

xdelta

Software as a service

- All the layers below the applications we use are done for us
- We can scale up or down as needed, provided that our workload is scalable
- We look after our data

xdelta

Accessing cloud services

- Your data are not in the same location are you are
- You are entirely dependent on your network connection
- Bandwidth and latency govern the behaviour you get
- Does it matter where your “stuff” actually runs ?

xdelta

Summary

- The delivery of service to clients is built on a huge pile of underlying layers
- In most cases, “it just works”
- Think about security, reliability, data backup, etc.
- As a delivery model, it works well for some things, badly for others

xdelta

Virtualisation and Cloud Services

Colin Butcher
XDelta Limited

www.xdelta.co.uk
+44 117 904 8209

xdelta