Designing virtualised infrastructures

Colin Butcher

"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 have.

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
- Networks let us communicate
- Typically we can trade memory for performance by holding large amounts of data and instructions in memory ready for use

- 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 time) – determines predictability of response
  - Understanding jitter is important for establishing timeout values
  - Latency fluctuations can cause system failures under peak load

- Understand how the applications could break down into parallel streams of execution:
  - Some will be capable of being split into many small elements with little interaction between the parallel streams of execution
  - Some will require very high interconnectivity between the parallel streams of execution
  - Some will require high-throughput single-stream processing
- Understand scalability – do as much as possible once only, do little as possible every time

- Ability to make use of hardware parallelism
- Granularity of data structures
- Synchronisation techniques
- Serialisation techniques
- Scalability techniques
- Compilers
- Application design
- Designing and writing very good code requires very good programmers

What is "virtualisation" in it's current context?

- It's another way of hiding the underlying hardware so that we can use it without having to think about it. We already have virtualisation of networks and storage subsystems. Blade technology brings virtualisation of the system infrastructure (chassis components).

- It's aimed at virtualising the processing element (CPU) where we can share out one or more physical CPUs amongst a number of "guest instances" of operating systems.

What can CPU virtualisation do for us?

- Improved utilisation of hardware resources

It can let us move "workload units" around a set of hardware resources to provide improved utilisation of the available hardware.

It can let us start a virtual system only when we need access to it. That could be an emulated system for intermittent access to historic data or software.

What can CPU virtualisation do for us?

- Improved high availability and disaster tolerance

  If that hardware is at different physical locations then it can also help to provide some level of high availability (within a site) and disaster tolerance (across multiple sites), provided that the data is replicated between the different locations and that the communications paths are available.

What can CPU virtualisation do for us?

- Parallelisation of processing

  It's another way to achieve parallelisation of processing without having to invest time in writing code that works well on a multi-processor system. However, we still need to think about concurrent access to data from multiple "guest instances" and providing access to the multiple "guest instances".

What can CPU virtualisation do for us?

- Multiple run-time environments

  We can create many "guest instances" where each has different operating system and layered product versions, then only run the ones we need at any given point in time.

- What else do we need to consider?

- Effects of latency and jitter
- Monitoring and management tools
- Software deployment techniques
- Planning downtime for maintenance on shared-use hardware with many "guest instances"
- Realistic testing for scale and performance
- Licensing policies
- Security (access control, authentication, encryption)

VLANs are used to segment a network:

- Implemented within core switches
- Connectivity between VLANs
- VLAN tagging of packets (802.1Q)
- VLAN tagging of packets out of NICs
- QoS (Quality of Service) and bandwidth reservation

- Converged ethernet as a common backbone in the data centre

Storage array controllers (eg: HP EVA):

- The array "hides" the behaviour of the discs from you
- The array "levels" the storage to provide best throughput for the access pattern to the entire array (or disk group)
- The array controller caches much of the data
- The only real performance issue is bandwidth to and from the array controller pair and what else is contending for that bandwidth

VSANs are used to segment a SAN fabric:

- Implemented within SAN switches (directors)
- Connectivity between VSANs

- Converged ethernet as a common backbone in the data centre

Hardware platforms

The future is (mostly) blades:

- Virtual connections from processing units over backplane channels – think about how WWIDs and MAC addresses are presented
- Modular systems provide great flexibility of configuration and interchangability of components
- Big high-end multiprocessor systems will still be needed for certain workloads
- Stand-alone systems will still be needed in highly secure or mission-critical / safety-critical environments

Costs and risks

You're trading a relatively high initial investment in the platform / infrastructure for a smaller cost of incremental growth. Going virtual for one or two systems on a new blade platform may not be the most cost effective solution, but it could be a long-term win.

The reliability and ease of use is heavily dependent on device firmware and compatibility between the components, plus the availability of monitoring, management and configuration tools.

Thank you for your participation

Q & A