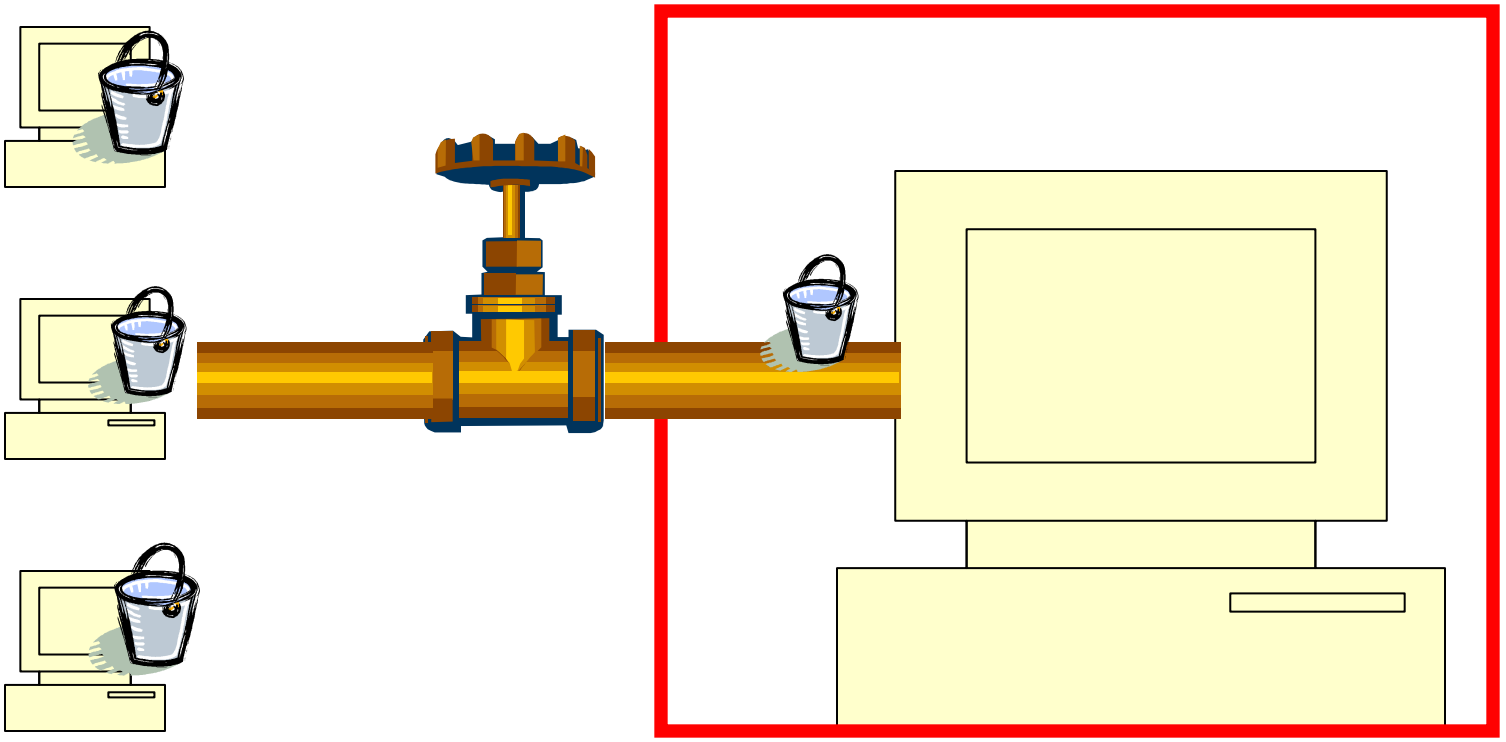
InfiniBand Device Virtualization in Xen

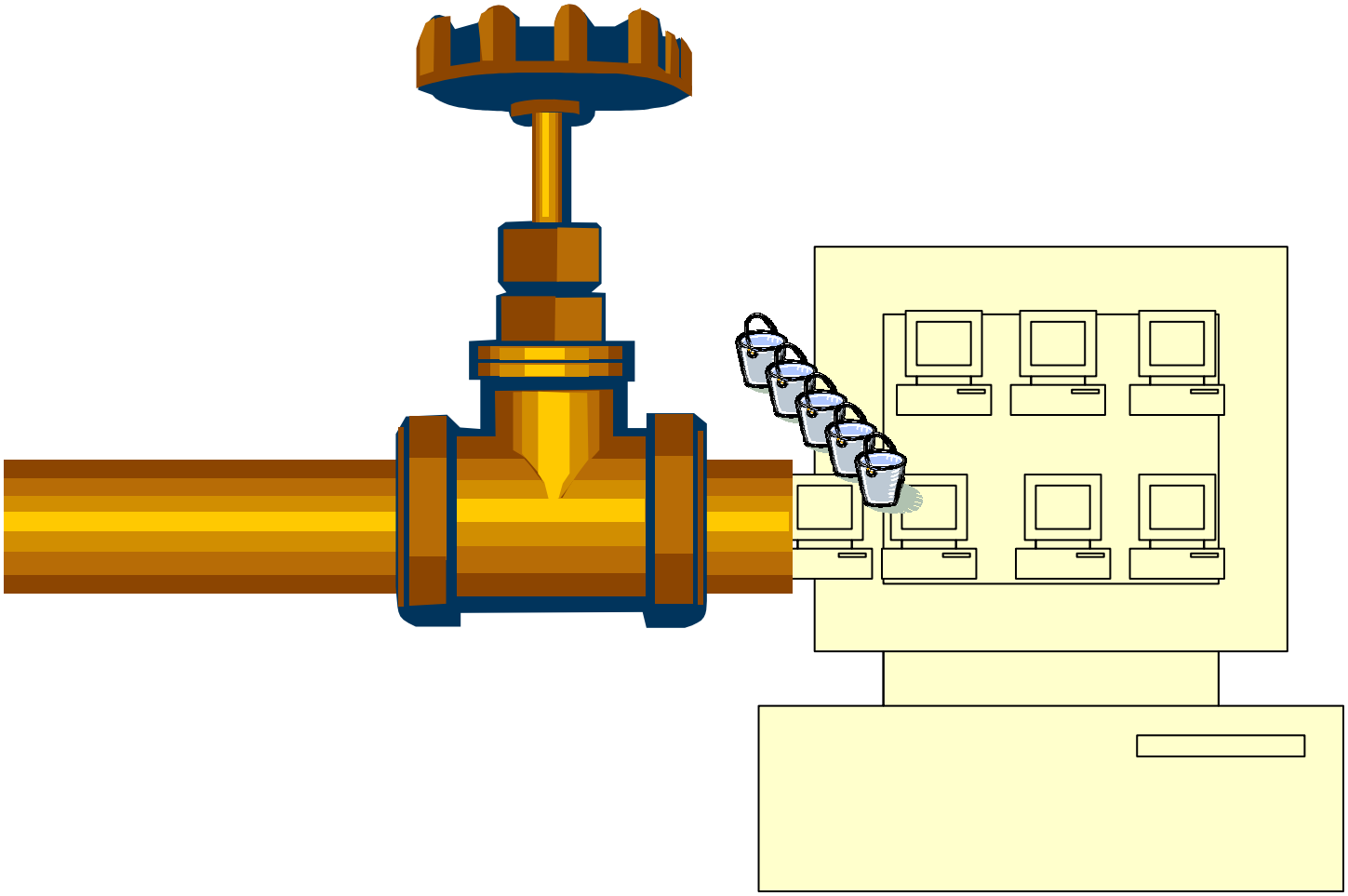
Dror Goldenberg (gdror@mellanox.co.il)

January 19, 2006

The IO Bottleneck



The IO Bottleneck (Virtualization)

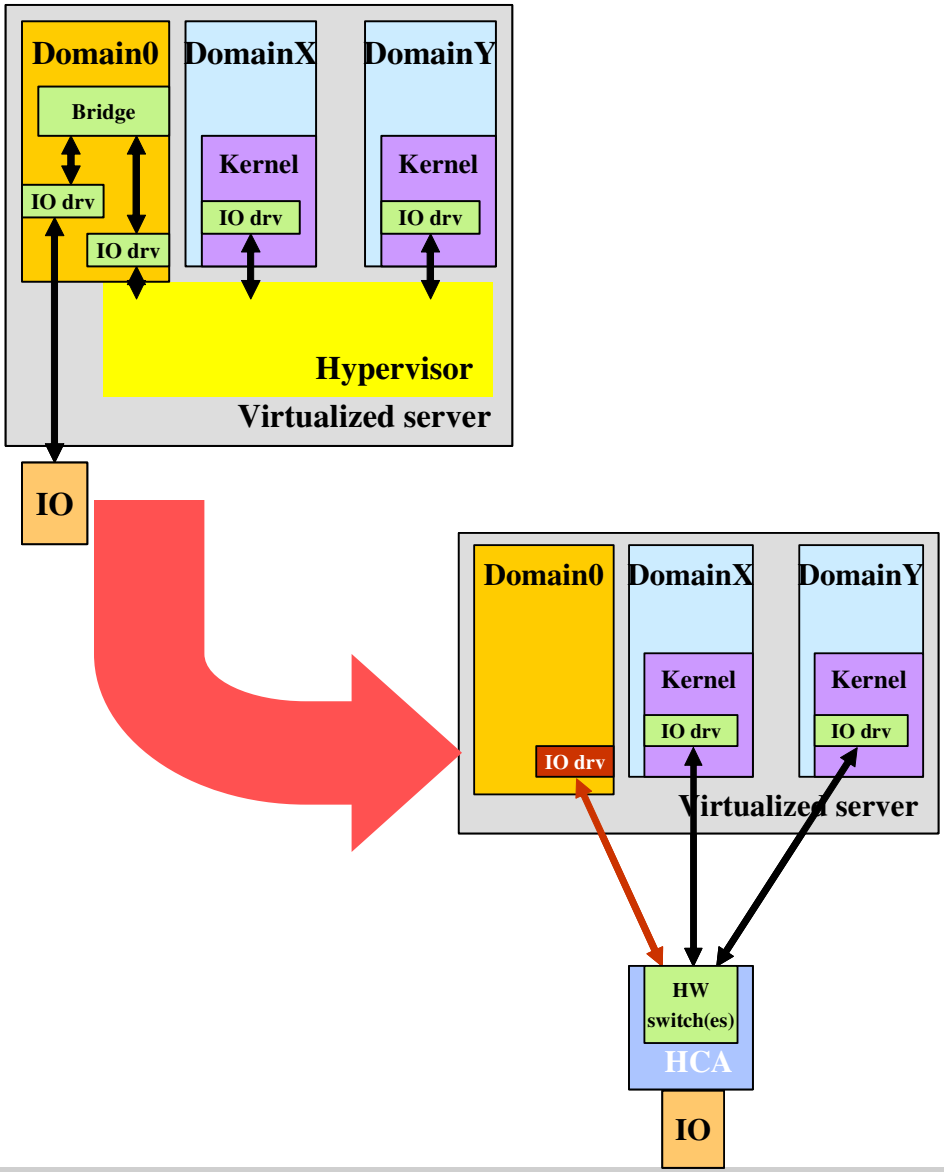


InfiniBand Key Benefits

- **Take InfiniBand Characteristics to the Virtualized Environment**
  - **Unmatchable I/O**
    - Low Latency (<2.5us application to application)
    - High Bandwidth (>2600MB/s bidirectional bandwidth)
  - **CPU Offloads**
    - Transport offload
    - RDMA
    - Kernel (and Hypervisor) bypass
  - **Fabric Consolidation**
    - IPC, Network, Storage, Backup, Management
- **Mature SW Stack**
  - **OpenIB stack is part of the Linux kernel**

InfiniBand Key Benefits (Cont'd)

- Channel Based I/O
  - Cross-channel isolation
  - Cross-channel protection
  - Native device sharing
- Hypervisor and Virtualization Stack Offloads
  - Saves data copies
  - Reduces context switching
  - Virtual switching
- Existing HW Fully Supports Virtualization
  - The most cost-effective path for single-node virtual servers
  - SW-transparent scale-out



Goals

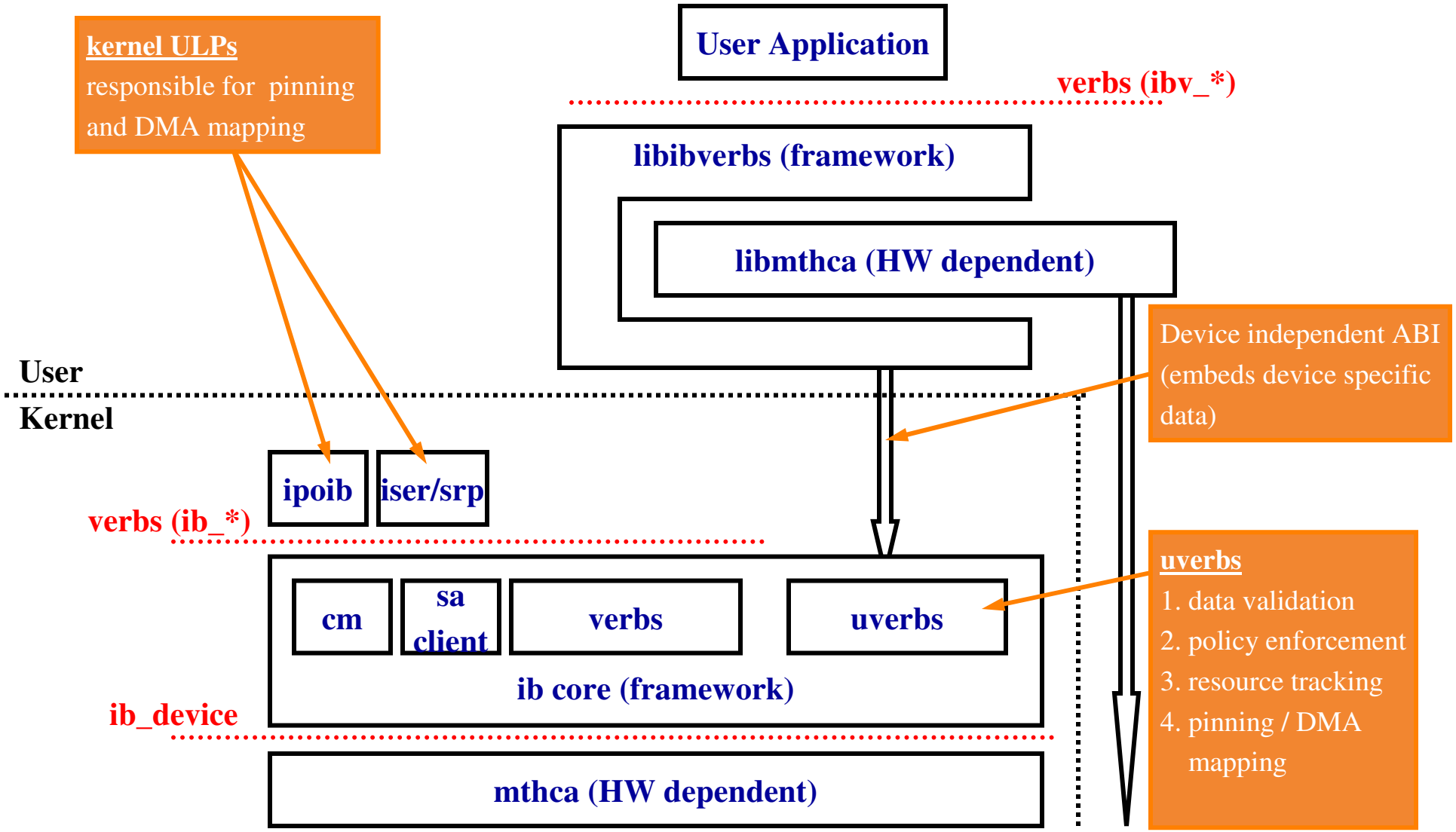
- Leverage high-performance IO in VMM
  - VM-transparent Hypervisor offload
- Direct access to IO from guest VM
  - Full Hypervisor bypass
  - Match native InfiniBand performance on virtual machine
- Enable InfiniBand unaware guests

- Multiple solutions: performance vs. HW awareness tradeoff
  - Partial InfiniBand performance gain – DomU is HW independent
  - Full InfiniBand performance gain – DomU is HW dependent
  - Different customers need different solutions

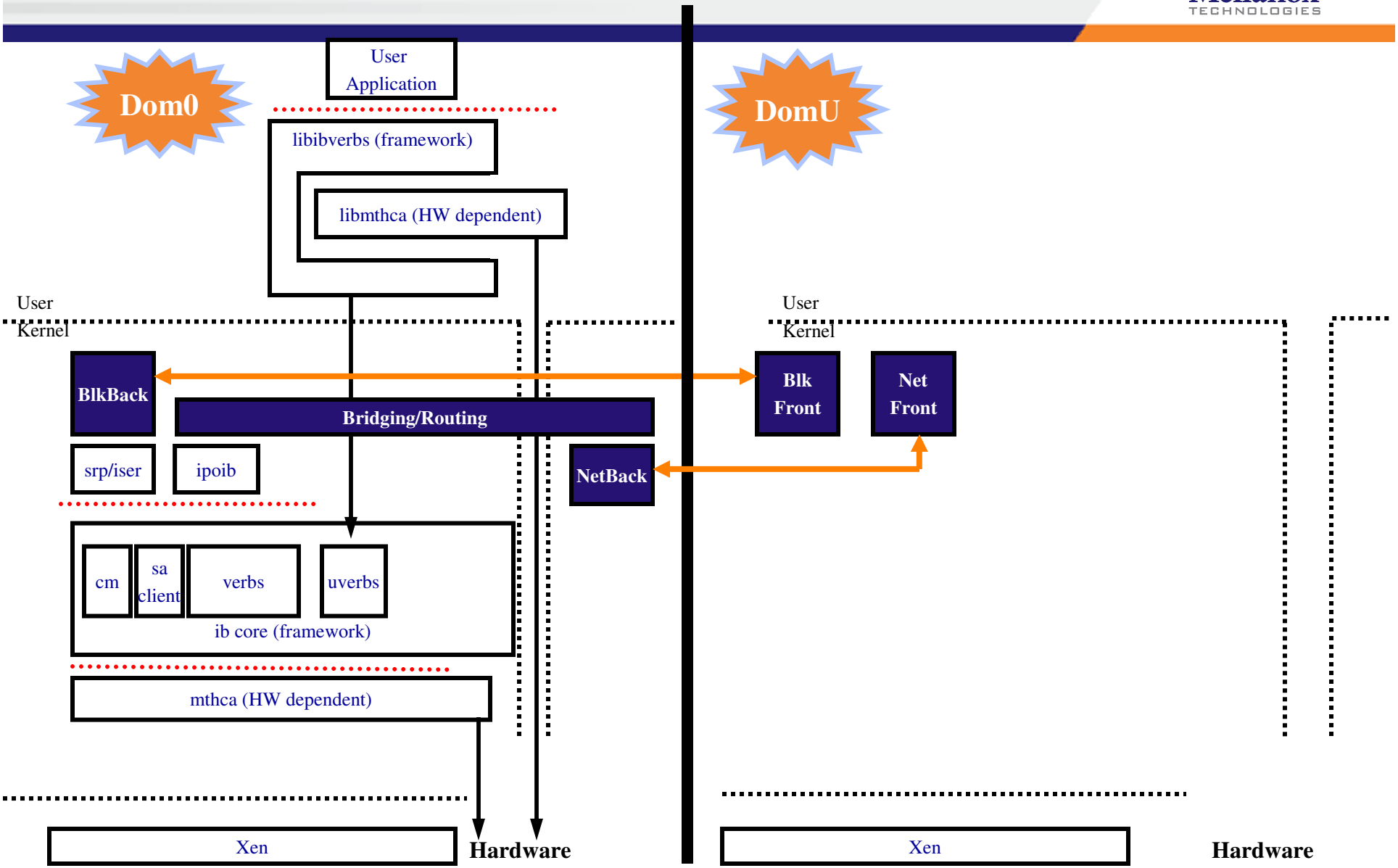
Plan

- Solution I: Enable IB in Xen Environment
  - Networking – IPoIB through standard net front/back
  - Block Storage – SRP/iSER through standard block front/back
- Solution II (1st Part): Native IB Networking and Storage in DomU
  - IB Front/Back infrastructure
  - Networking – IPoIB in Dom U over IB access layer
  - Storage – SRP/iSER in Dom U over IB access layer
- Solution II (2nd Part): Full Native IB Support in DomU
  - SDP
  - File storage (NFS/RDMA)
  - Userland access layer
  - MPI
- Solutions can operate concurrently

OpenIB Linux Stack Architecture



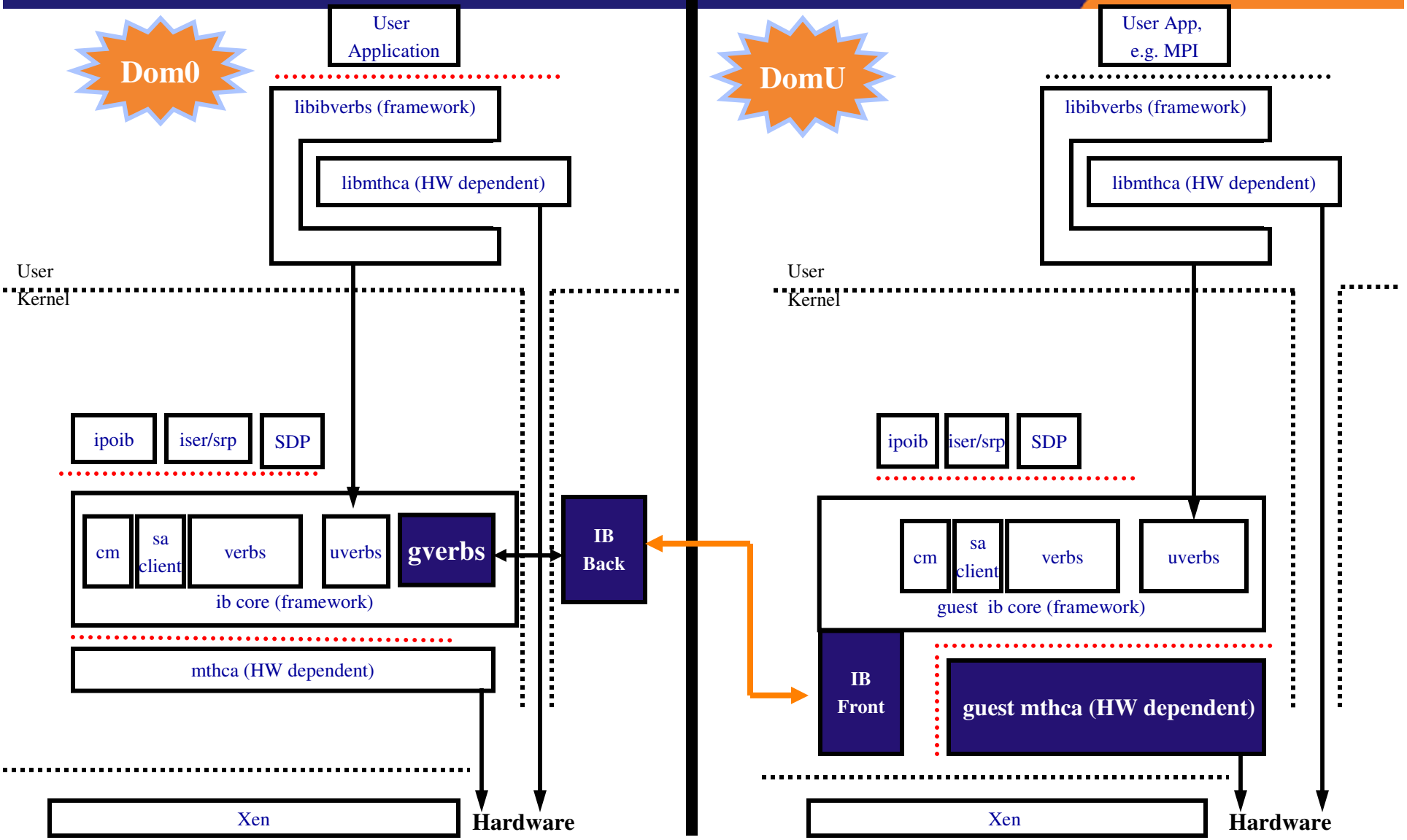
Solution I: Infrastructure



Solution I: Summary

- **Benefits**
  - **Xen machine can connect to InfiniBand infrastructure**
  - **Utilize InfiniBand high performance on network/storage**
    - Can be further improved using Solution II
  - **Hardware independent Dom U**
    - Can enable running legacy OS guest that has no IB support
- **Challenges**
  - **IPoIB MAC address is 20B; alternatives to packet demultiplexing (Dom 0):**
    - use multiple QPs in Dom0 and assign one QP per DomU
    - use 20B MAC address all the way netfront/back/bridge
    - use IP based demultiplexing at Dom 0
  - **Multicast requires explicit registration to groups**
  - **No issues expected with storage**

Solution II: InfiniBand Frontend / Backend Infrastructure



Solution II (1st Part): Summary

- **Benefits**
  - **Bring native IB to the guest kernel**
  - **Improve network/storage performance**
    - Expect close to native performance
    - Reduction in context switch, packet copy, etc.
  - **Enable InfiniBand HW independent Dom U**
- **Challenges**
  - **IPoIB uses physical unregistered addresses**
    - Patch kernel to use pre-registered networking data buffers
    - Register entire DomU physical address
      - Need to exclude PTE/Balloon pages
    - Separate send and receive pinning method
      - Send – use read-only registered memory – can also be used for PTEs
      - Receive – use special allocator that registers buffers for DMA write
    - Use HCA paging on demand
  - **SRP uses physical addresses**
    - Batch registration
    - Use paging on demand
  - **Interrupt forwarding to DomU**

Solution II (2nd Part): Summary

- Benefits
  - Bring native IB to the guest
  - Improve network performance - SDP
  - Enable middleware running in DomU – e.g. MPI