Performance Comparison of Two Virtual Machine Scenarios Using an HPC Application - A Case study Using Molecular Dynamics Simulations

Anand Tikotekar, Hong Ong, Sadaf Alam, Geoffroy Vallée, Thomas Naughton, Christian Engelmann, and Stephen L. Scott

Computer Science and Mathematics Division
Oak Ridge National Laboratory

3rd Workshop on System-level Virtualization for High Performance Computing (HPCVirt) 2009, Nuremberg, Germany, March 30, 2009

Outline

- Background
- Objectives
- Experimental setup
- Evaluation methodology
- Overall and detailed results
- Conclusion
- Future work

Background

- Virtualization is an increasing field of importance in HPC
  - Availability of low-overhead hypervisors, such as Xen
  - Added efficiency through full utilization of resources
  - Support for customized environments to fit application needs
  - Increased flexibility for high availability and fault tolerance
- While overheads are low, they still exist
  - Virtualization of hardware resources causes performance hit
  - Compute-bound applications experience a lower overhead
  - I/O-bound applications have a higher overhead
- Quantifying these overheads beyond pure wall clock time
  - Can help to understand their root causes
  - Can offer tunable solutions for adaptation to individual application needs

Objectives

- Investigate novel virtual machine (VM) configurations
  - To obtain a benefits vs. performance-loss tradeoff
  - To reduce the performance overhead of virtualization, while maintaining important benefits of virtualization
- Evaluate the difference between two VM configurations that perform the same work with different flexibility
  - 2 VMs per single-core node, 1 process per VM
  - 1 VM per single-core node, 2 processes per VM
- Since overheads are application depended, we focus on a specific scientific application
  - LAMMPS, a classical molecular dynamics code

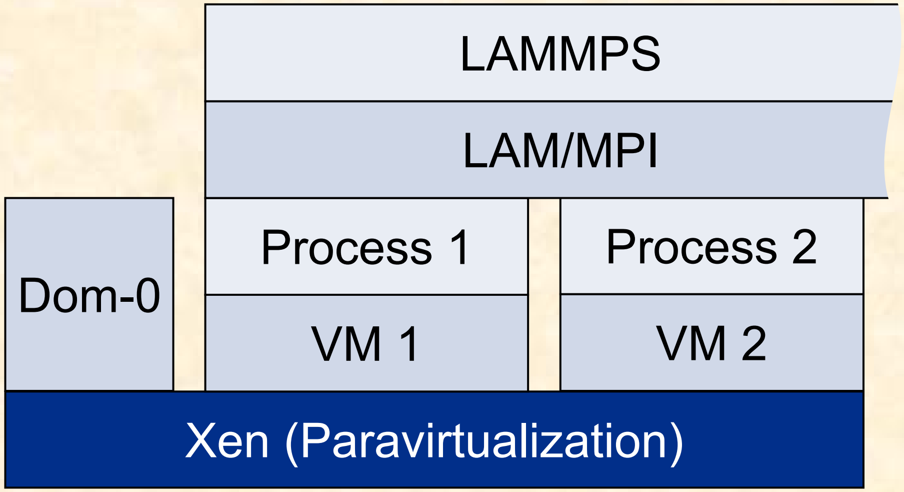
Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)

- Parallel application that studies properties of particles over time
- Interaction through pair-wise forces using Newton’s law
- Widely used from material science to computational biology
- Most algorithms reduce cost from $O(n^2)$ to $O(n)$ through approximations
- Application setup:
  - LAMMPS protein benchmark: Rhodopsin protein in solvated lipid bilayer
  - 1,024,000 atoms, 100 timesteps

Input	Value
Atom style	full
Pair style	lj (Lennard-Jones potential)
Bond style	harmonic
Neighbor modify	Delay 5, every 1 second
Kspace style	pppm

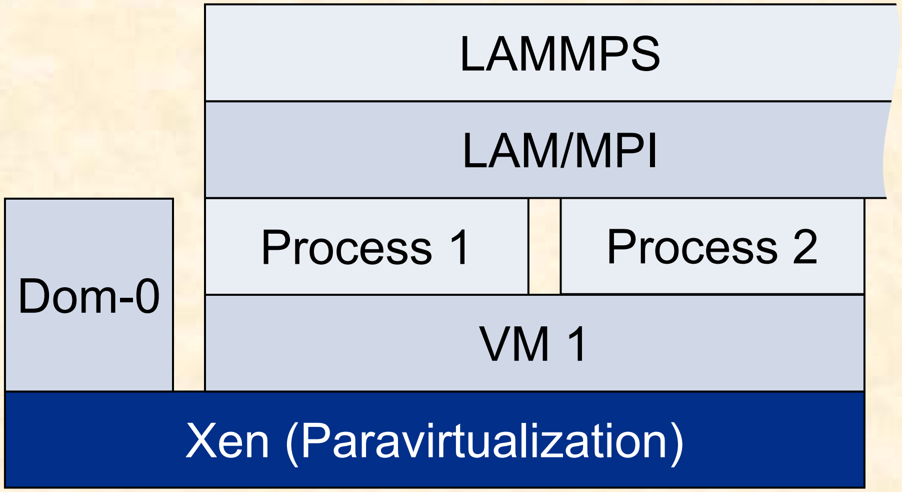
System Setup

Configuration 1



- **VM1 Resources = VM2 Resources**
- **Memory = 256 MB/VM**
- **Application memory = 231 MB/VM**
- **Total number of processes = 16**

Configuration 2



- **Memory = 512 MB/VM**
- **Application memory = 462 MB/VM**
- **Total number of processes = 16**

Physical node: Memory = 768MB, CPU = 2GHz, L2 Cache = 256kB, Local Root FS, NFS

Setup of 1 physical node is shown. This setup is replicated across 8 nodes.

Evaluation Methodology

- Compare VM configurations with the same application run
  - Total wall clock time
  - Detailed CPU, memory, system, and I/O metrics
- Collection of metrics with VMstat process in each VM
  - 1 second sample frequency
  - CPU metrics: User, system, idle, I/O wait, stolen time
  - Memory metrics: Swap, free, inactive, active
  - System metrics: Interrupts and context switches
  - I/O metrics: NFS disk I/O blocks sent and received

Overall Performance Difference

Configuration 1

Description	Wall clock time in sec
2 VMs per node, 1 process per VM	**1686**

- **Averaged over 5 runs**
- **Standard deviation: 3%**
- **2.4% slower**

Application phases	Wall clock time in %
Pair time	**45.33**
Bond time	**1.52**
Kspace time	**32.5**
Neighbor time	**7.05**
Communication time	**7.55**
Other time	**6.05**

Configuration 2

Description	Wall clock time in sec
1 VM per node, 2 processes per VM	**1646**

- **Averaged over 5 runs**
- **Standard deviation: 1.5%**
- **40 seconds faster**

Application phases	Wall clock time in %
Pair time	**46.05**
Bond time	**1.61**
Kspace time	**33.01**
Neighbor time	**7.12**
Communication time	**6.97**
Other time	**5.24**

CPU Metrics: VMstat Average Data

Configuration 1

Metric	VM1 in %	VM2 in %
User time	**27.4**	**26.4**
System time	**1**	**1.2**
I/O wait time	**4.1**	**4.5**
Idle time	**39.2**	**39.5**
Stolen time	**28.3**	**28.4**

Metric	Configuration 1 in %
User time	**53.8**
System time	**2.2**
I/O wait time	**2.6**
Idle time	**41.4**

Configuration 2

Metric	VM1 in %
User time	**63.9**
System time	**9**
I/O wait time	**0.7**
Idle time	**20.4**
Stolen time	**6**

Metric	Configuration 2 in %
User time	**63.9**
System time	**9**
I/O wait time	**0.7**
Idle time	**26.4**

CPU Metrics: XenTop Average Data

Configuration 1

Domain	CPU time in %
Dom-0	**15.4**
VM 1	**31.2**
VM 2	**31.2**
Idle	**22.2**

- **Total used CPU time: 77.8%**
- **More idle time**

Configuration 2

Domain	CPU time in %
Dom-0	**13**
VM 1	**74**
Idle	**13**

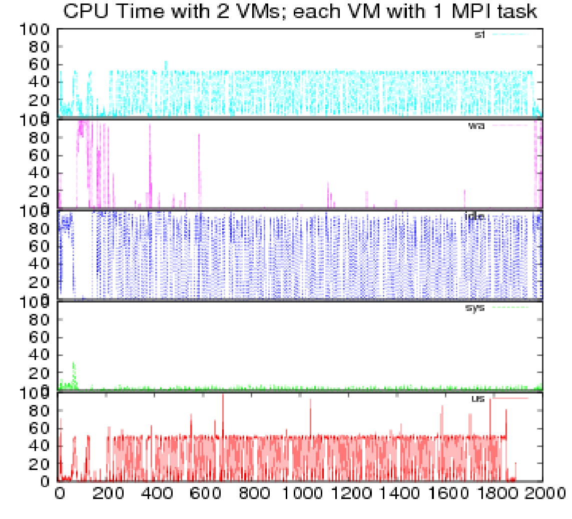
- **Total used CPU time: 87%**
- **Higher CPU utilization**

CPU Metrics: VMstat Sample Data

Configuration 1

Configuration 2

VM 1



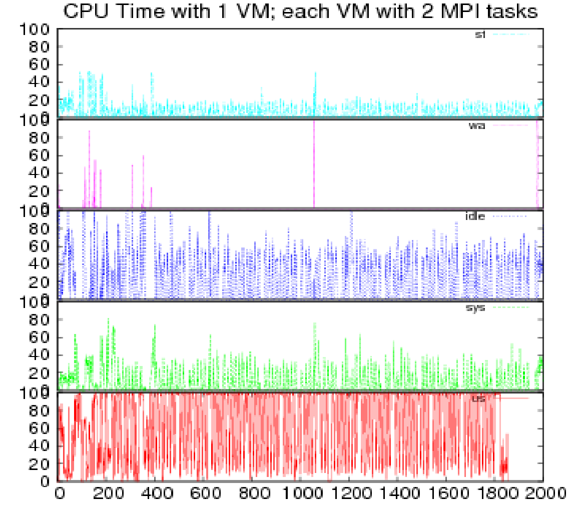
▶ Stolen time in % ◀

▶ I/O wait time in % ◀

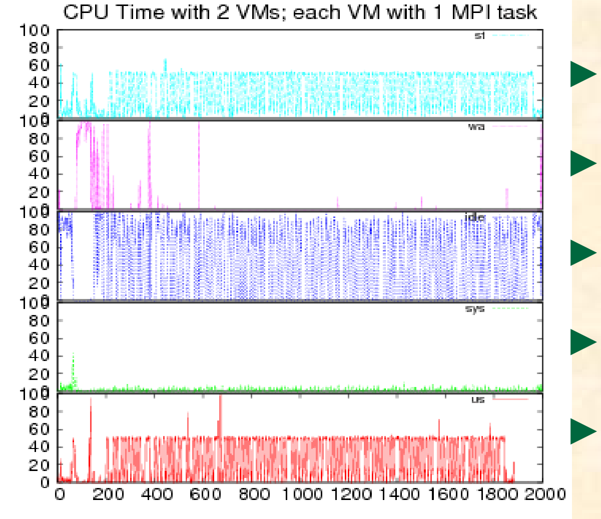
▶ Idle time in % ◀

▶ System time in % ◀

▶ User time in % ◀



VM 2



▶ Stolen time in %

▶ I/O wait time in %

▶ Idle time in %

▶ System time in %

▶ User time in %

- Wall clock difference is 2.4%, but user time difference is 19.2%
- Configuration 1:
  - Xen may not efficiently exploit idle time
  - More NFS pressure with 2VMs
- Configuration 2:
  - More context switches
  - Higher L2/TLB misses?

Memory Metrics: VMstat Average Data

Configuration 1

Metric	Per VM in %
Swap allocated	**20.1**
Free	**3**
Inactive	**18.7**
Active	**57.6**

- **Total available within a VM: 256.0 MB**
- **Used per VM: 285.6 MB**
- **Used by application process: 231.0 MB**

- **More used per VM due to resource management for 2 VMs on the same host**
- **Similar free amounts in both configs**

Configuration 2

Metric	Per VM in %
Swap allocated	**8.1**
Free	**3.9**
Inactive	**26.1**
Active	**61.8**

- **Total available within a VM: 512.0 MB**
- **Used per VM: 509.0 MB**
- **Used by application process: 462.0 MB**

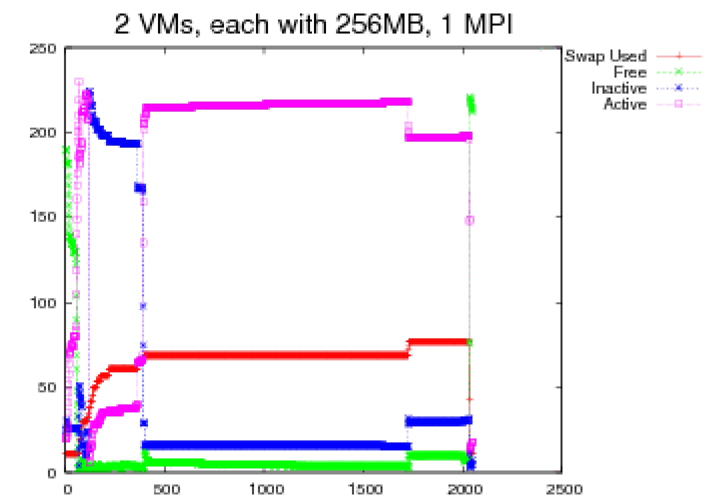
- **Less allocated swap, though not used!**
- **Remember, only 2.4% wall clock difference**

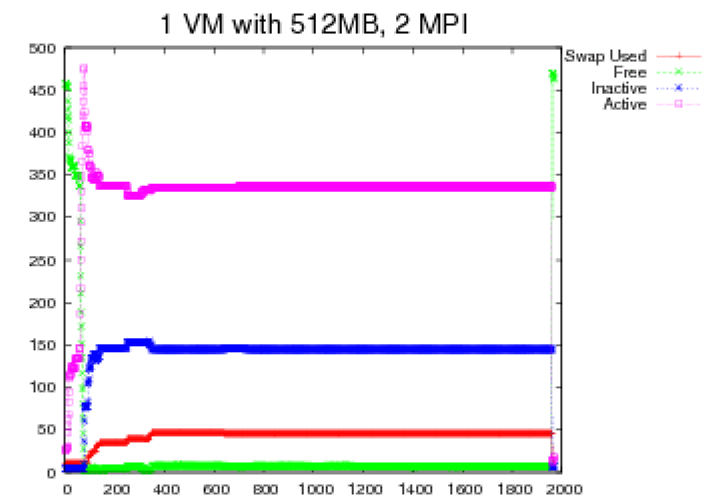
Memory Metrics: VMstat Sample Data

Configuration 1

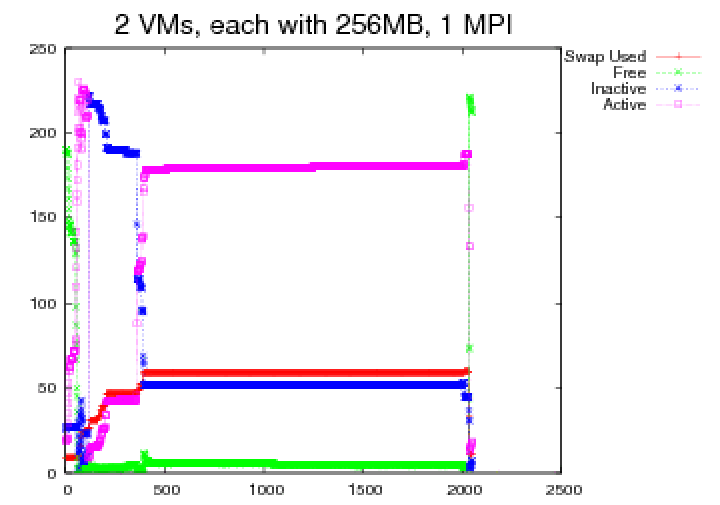
Configuration 2

VM 1





VM 2



I/O Metrics: VMstat Average Data

Configuration 1

Metric	
Disk I/O blocks sent/sec	65.5
Disk I/O blocks received/sec	51.5

Configuration 2

Metric	
Disk I/O blocks sent/sec	20
Disk I/O blocks received/sec	13.2

- Much higher I/O activity
- Remember, only 2.4% wall clock difference

- No aggregation of I/O requests/responses
- Increased I/O activity may be the cause for the increased memory usage per VM
- XenTop: Dom-0 CPU time was 15.4%

- Total number of block sent/received similar for both configurations

- Aggregation of requests/responses possible

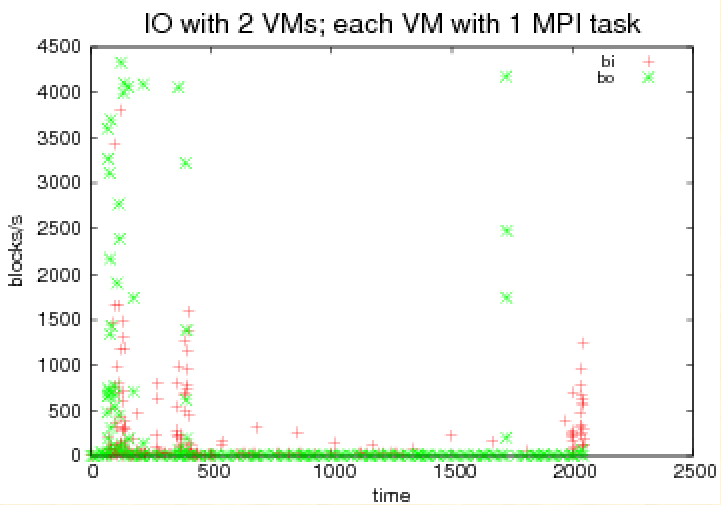
- XenTop: Dom-0 CPU time was 13.4%

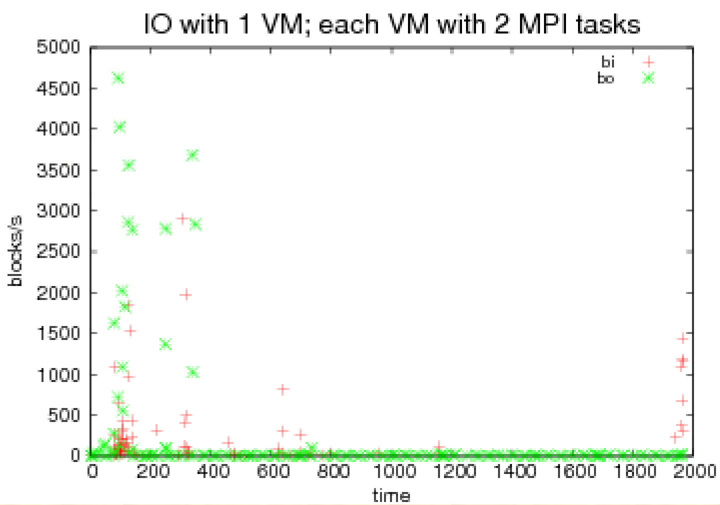
I/O Metrics: VMstat Sample Data

Configuration 1

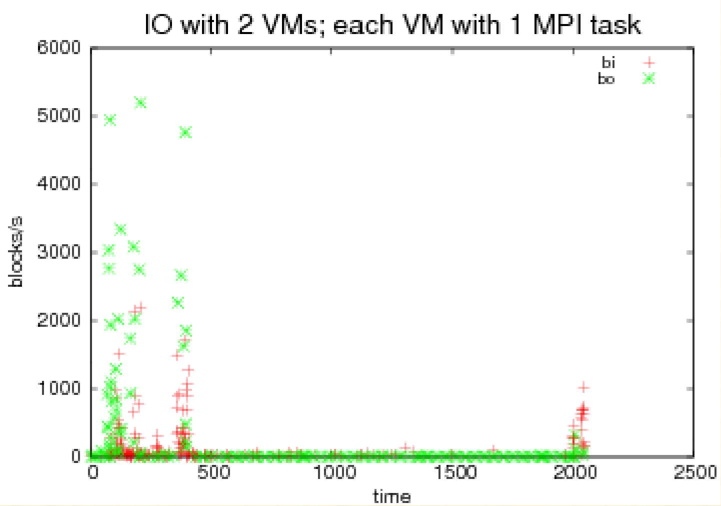
Configuration 2

VM 1





VM 2



System Metrics: VMstat Average Data

Configuration 1

Metric	
Number of interrupts/sec	**1403.8**
Number of context switches/sec	**1332.8**

Configuration 2

Metric	
Number of interrupts/sec	**1357.5**
Number of context switches/sec	**1698.1**

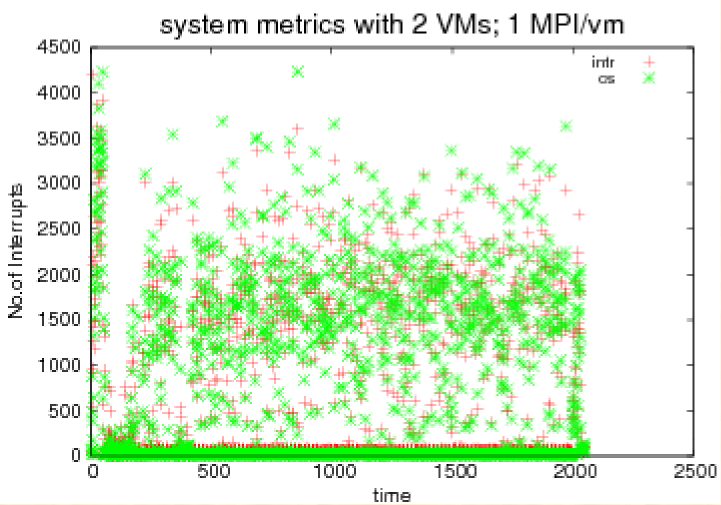
- **22% more context switches**
- **Probably higher L2/TLB misses**
- **May explain higher contribution to user and system utilization**

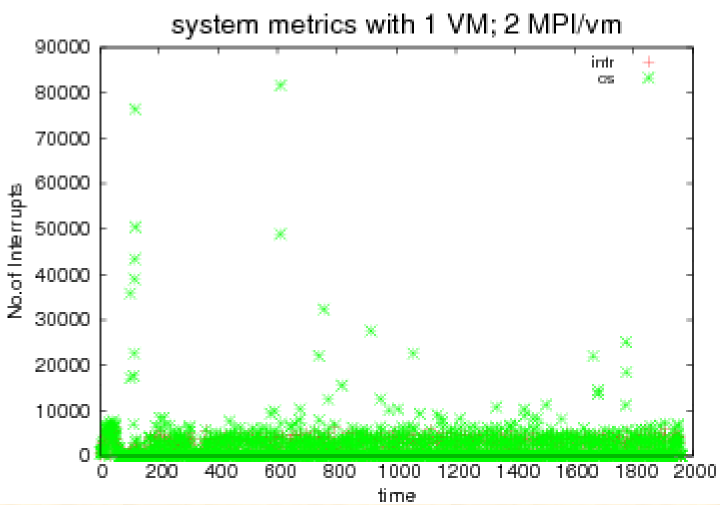
System Metrics: VMstat Sample Data

Configuration 1

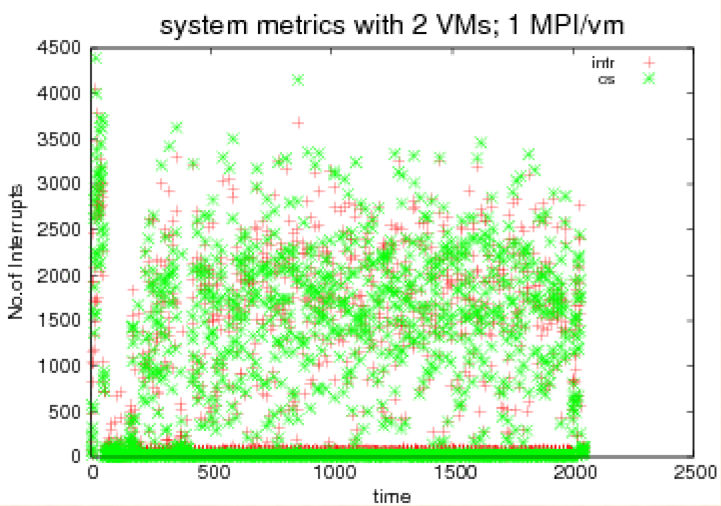
Configuration 2

VM 1





VM 2



Conclusion

- 8 VMs with 2 processes each is slightly more efficient than 16 VMs with 1 process each
- Overall performance difference is only 2.4%
- This study sheds light on how VM configurations impact an HPC application
- The investigation shows how Xen in configuration 1 and Linux in configuration 2 manage resources differently

Future Work

- Compare more VM configurations with and without hardware virtualization support
- Use tools such as Xenoprof along with VMstat and XenTop
- Study more, different HPC applications
- Study and quantify the effects of flexibility offered by various VM configurations

Questions?