Experiences in Using OS-level Virtualization for Block I/O

Dan Huang, University of Central Florida
Jun Wang, University of Central Florida
Gary Liu, Oak Ridge National Lab

Contents

Contents

Motivation

- Nowadays in HPC, job schedulers such as PBS/TORQUE are used to assign physical nodes, exclusively, to users for running jobs.
  - Easy configuration through batch scripts
  - Low resource utilization
  - Hard to meet interactive and ad-hoc analytics' QoS requirements.
- Multiple jobs access to shared distributed or parallel file systems to load or save data.
  - Interference on PFS
  - Negative impact on jobs' QoS

Resource Consolidation in Cloud Computing

- In data centers, cloud computing has been widely deployed for elastic resource provisioning.
  - High isolation with low mutual interference

- Cloud computing employs various virtualization technologies to consolidate physical resources.
  - Hypervisor-based virtualization: VMWare, Xen, KVM
  - OS-level virtualization: Linux container, OpenVZ, Docker

Virtualization in HPC

- HPC uses high-end and dedicated nodes to run scientific computing jobs.
  - Could HPC analysis cluster be virtualized with low overhead?
  - What type of virtualization should be adopted?
- According to the previous studies[1, 2, 3], the overhead of hypervisor-based virtualization is high.
  - Overhead on disk throughput ≈ 36%
  - Overhead on memory throughput ≈ 53%

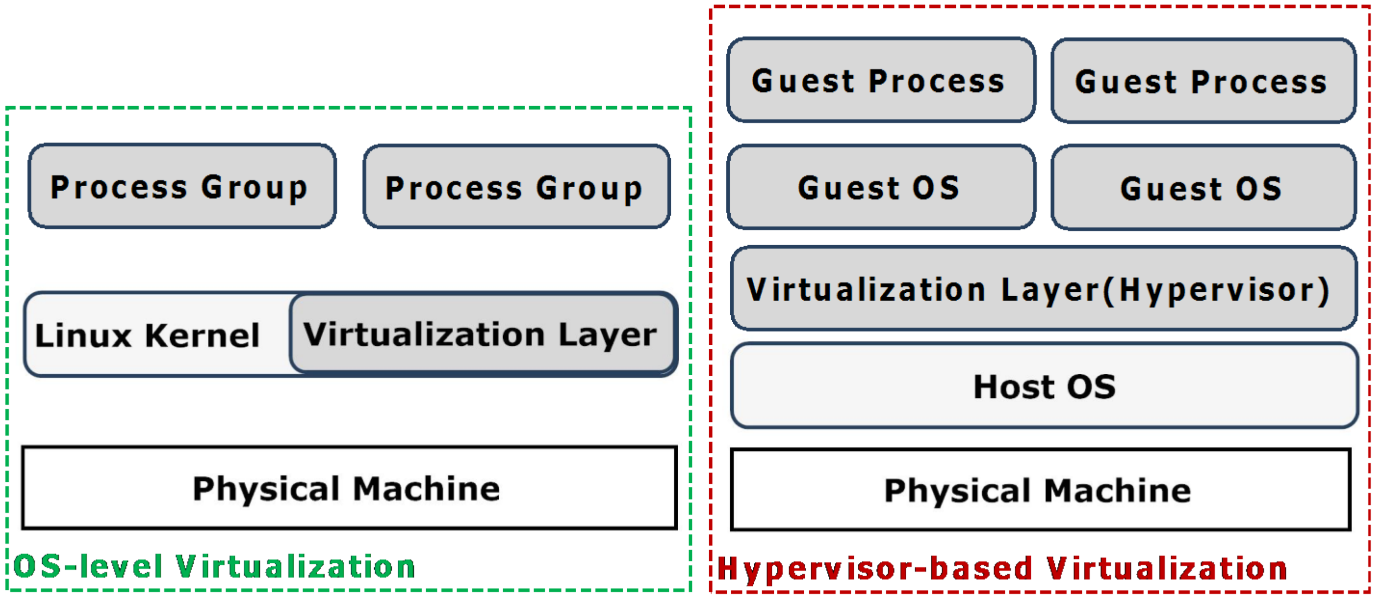
- [1] Nikolaus Huber, Marcel von Quast, Michael Hauck, and Samuel Kounev. Evaluating and modeling virtualization performance overhead for cloud environments. In CLOSER, pages 563-573, 2011.
- [2] Stephen Soltesz, Herbert Potzl, Marc E Fiuczynski, Andy Bavier, and Larry Peterson. Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In ACM SIGOPS Operating Systems Review, volume 41, pages 275-287. ACM, 2007.
- [3] Miguel G Xavier, Marcelo Veiga Neves, Fabio D Rossi, Tiago C Ferreto, Timoteo Lange, and Cesar AF De Rose. Performance evaluation of container-based virtualization for high performance computing environments. In Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on, pages 233-240. IEEE, 2013.

Contents

Hypervisor and OS-level Virtualization

- Virtualization technology takes advantage of the trade-off between isolation and overhead.
- Hypervisor-based virtualization has a hypervisor (or VM monitor) layer under the guest OS and it introduces high performance overhead and is not acceptable to HPC.
- OS-level virtualization (container based) is a lightweight layer in Linux kernel.

Hypervisor and OS-level Virtualization (cont.)



The Internal Components of OS-level Virtualization

- OS-level virtualization shares the same operating system kernel.

- 1) Control Groups (CGroups)
  - CGroups controls the resource usage per process group.

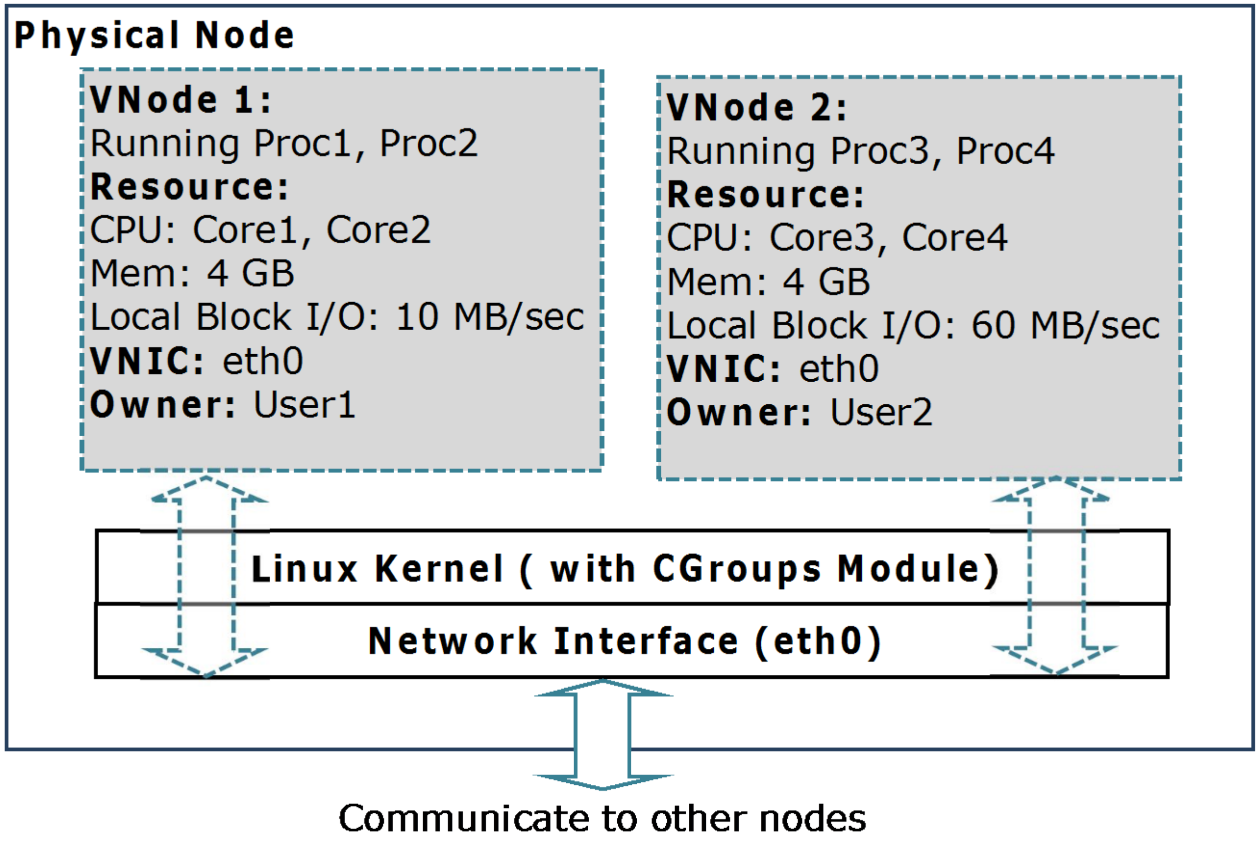
- 2) Linux Namespaces
  - Linux Namespace creates a set of isolated namespaces such as PID and Network Namespaces etc.

Allocating Block I/O via OS-level Virtualization

- There are two methods for allocating block I/O in CGroups module.
- 1) Throttling functionality
  - Set an upper limit to a process group's block I/O
- 2) Weight functionality
  - Assign shares of block I/O to a group of processes

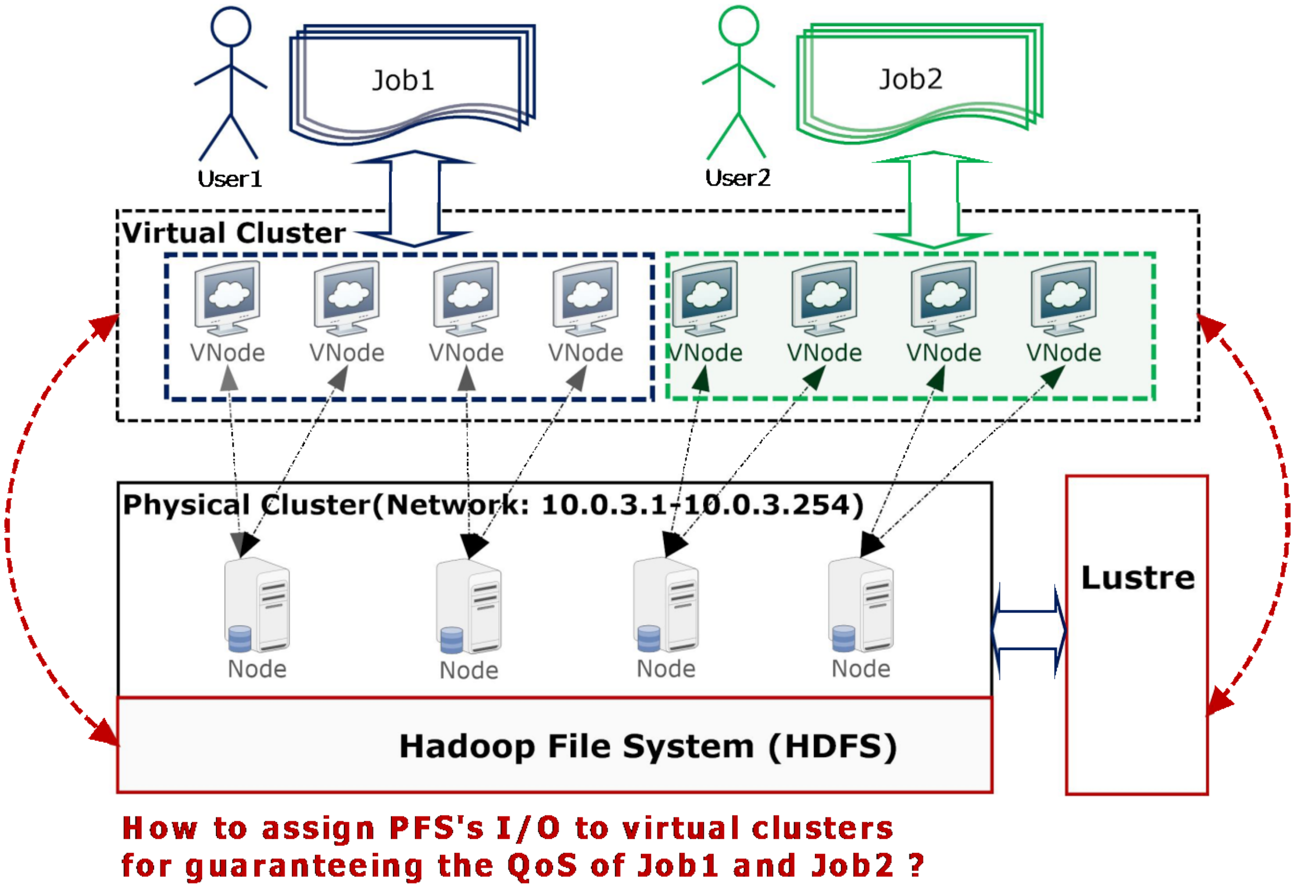
Contents

Create Virtual Node (VNode)

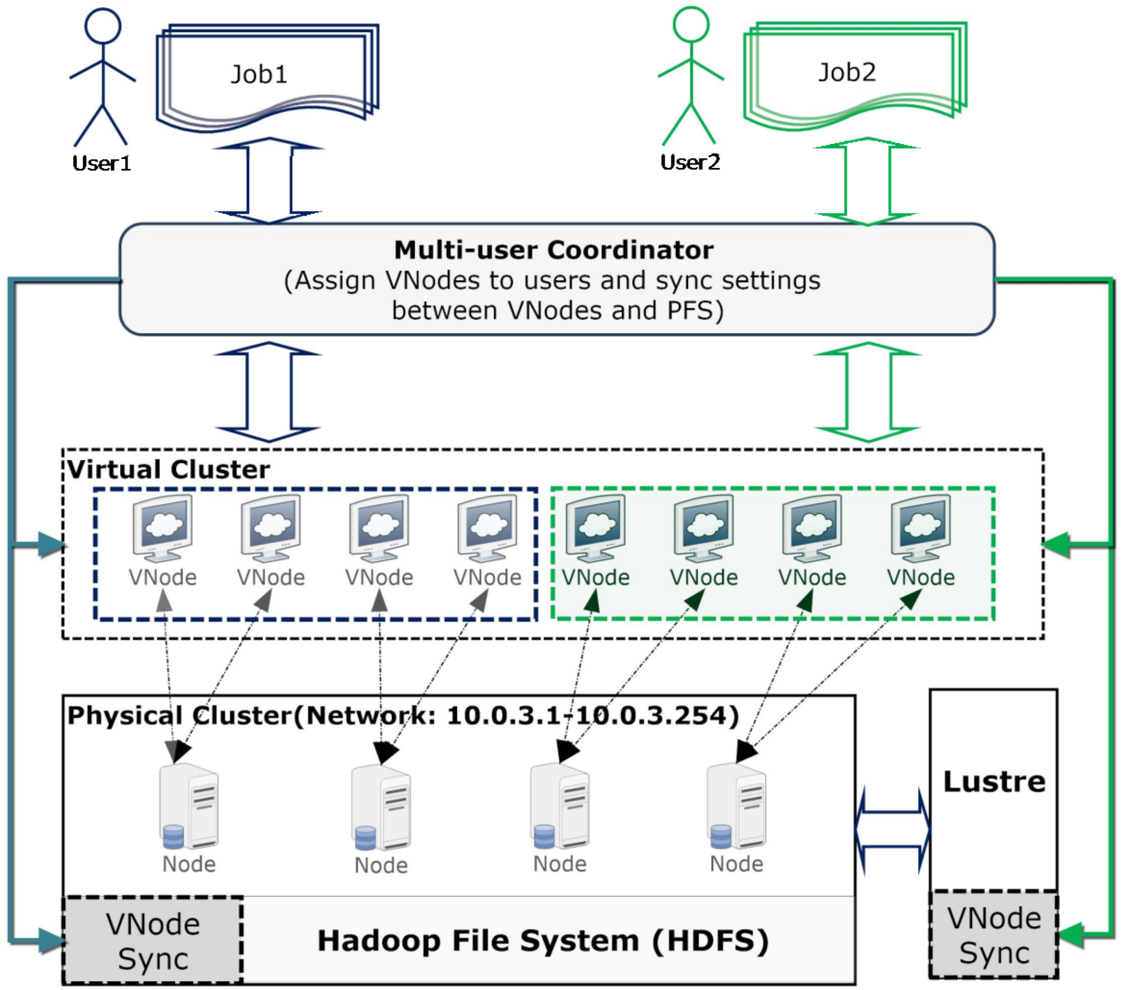


The Gap Between Virtual Node and PFS

Configuration Gap:
The shared I/O resources of a PFS is hard to be controlled by current resource allocation mechanisms, since the I/O configurations on users' VNodes can not take effect on a remote PFS.


User1
Job1
User2
Job2
Virtual Cluster
VNode
VNode
VNode
VNode
VNode
VNode
VNode
VNode
Physical Cluster(Network: 10.0.3.1-10.0.3.254)
Node
Node
Node
Node
Lustre
Hadoop File System (HDFS)
How to assign PFS's I/O to virtual clusters
for guaranteeing the QoS of Job1 and Job2 ?

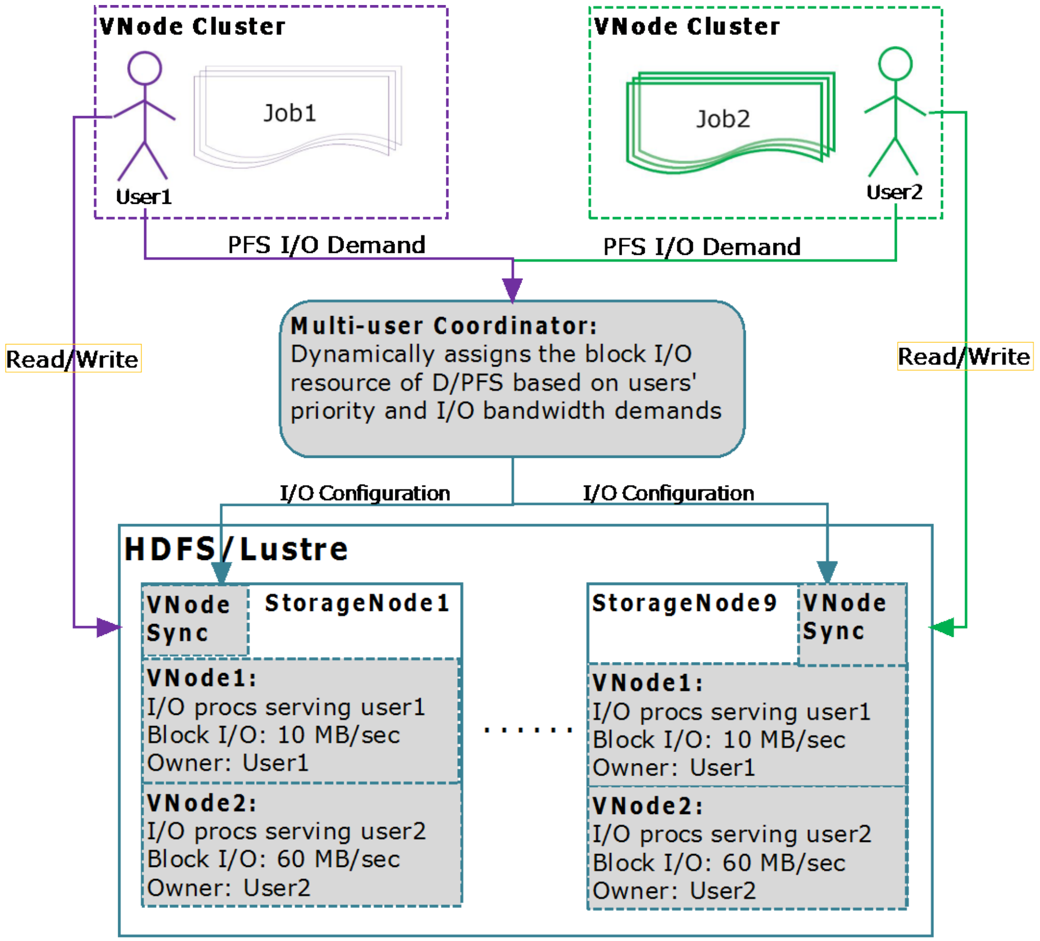
The Design of I/O Throttling Middleware



The Structure of VNode Sync

VNode Sync:

1) Accept I/O configurations
2) Apply I/O configurations into VNodes
3) Intercept users' I/O request handlers
4) Insert handlers into corresponding VNodes



Contents

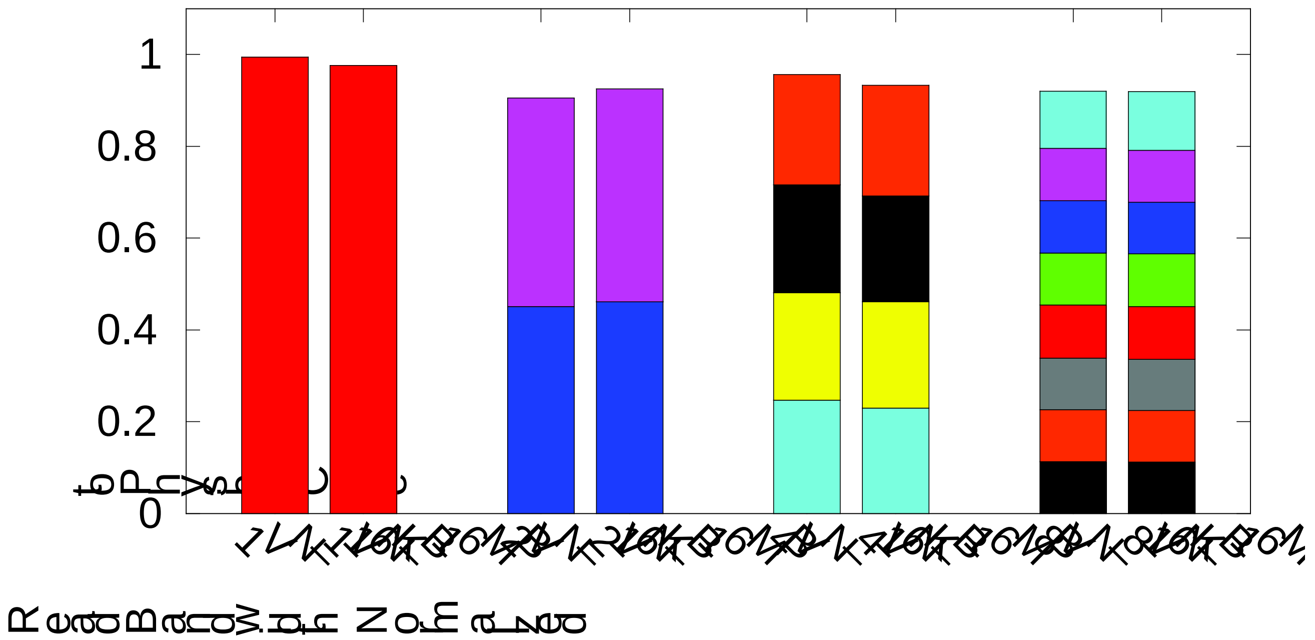
Single Node Testbed

The Configuration of Single Node Testbed	
Make& Model	Dell XPS 8700
CPU	Intel i7 Processor, 64 bit, 18 MB L2, 2.8 GHz, 4 cores
RAM	8×2 GB
Internal Hard Disk	1× Western Digital Black SATA 7200rpm 1 TB
Local File System	EXT3
Operating System	CentOS 6 64-bit, kernel 2.6.32 504.8.1.el6

Distributed Testbed

The Configuration of Marmot Cluster	
Reserve 17 nodes in Marmot	
Make& Model	Dell PowerEdge 1950
CPU	2 Opteron 242, 64 bit, 1 MB L2, 1GHz
RAM	8×2.0 GB RDIMM, PC3200, CL3
Internal Hard Disk	1× Western Digital Black SATA 7200rpm 2 TB
Network Connection	1 × Gigabit Ethernet
Operating System	CentOS 6 64-bit, 2.6.32 504.8.1.el6
Switch Make & Model	152 port Extreme Networks BlackDiamond 6808
HDFS	1 head node and 16 storage nodes
Lustre	1 head node, 8 storage nodes and 8 client nodes

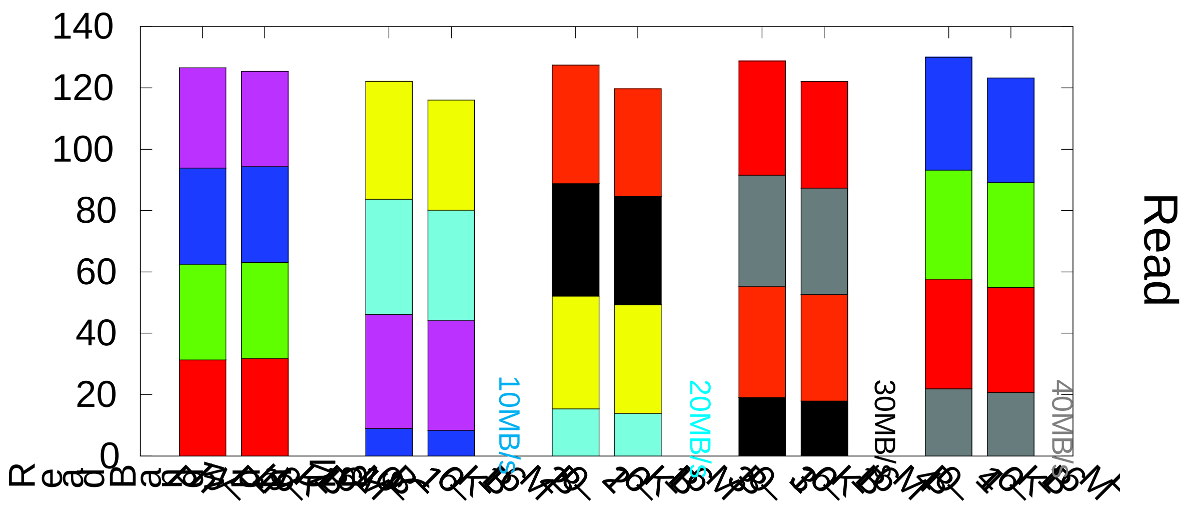
Read Overhead on Single Node



Numble of VNodes and Object Size

The worst read overhead is less than 10%.

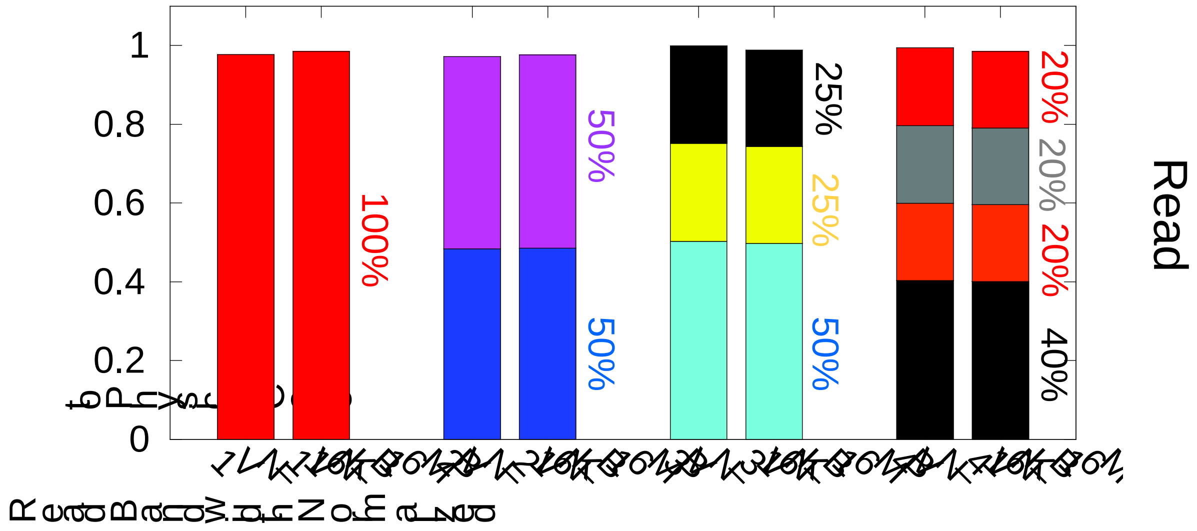
Throttling Read on Single Node



Throttle Rate on Bottom VNode (MB/s) and Object Size

The throttle functionality could guarantee the process's I/O does not exceed the upper limits. But it is largely influenced by other concurrent processes

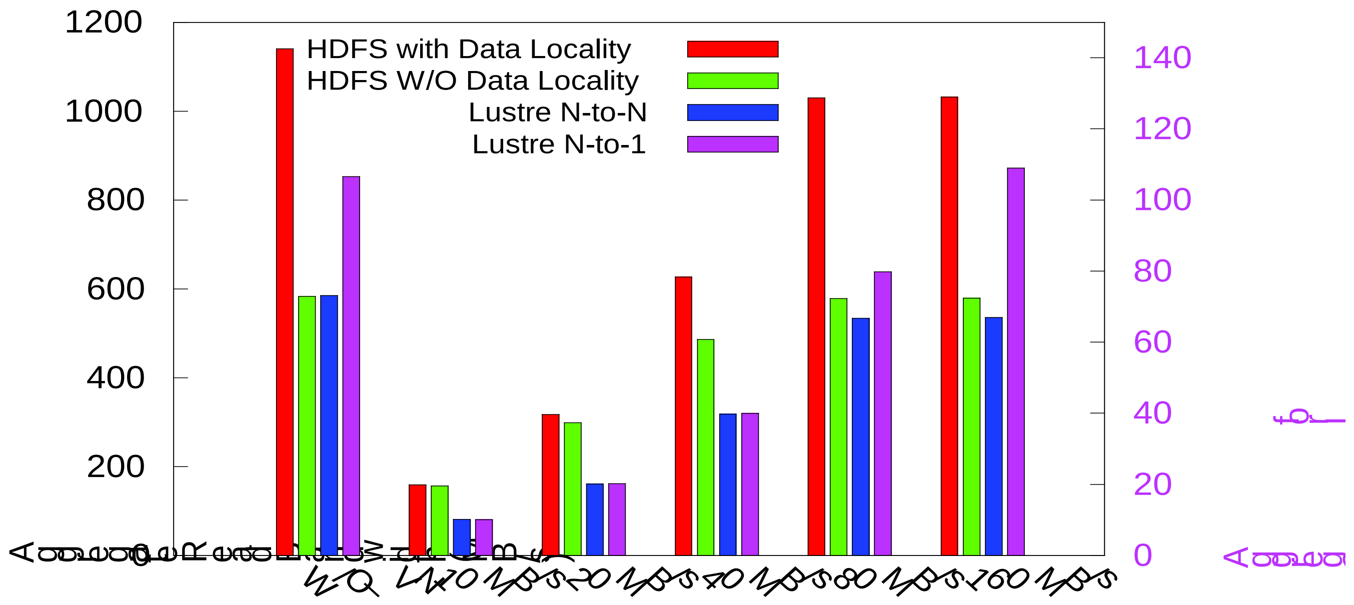
Weight Read on Single Node



Numble of VNodes and Object Size

The result shows that the overhead of the weight function is less that 8%. The weight module does not suffer from interference and can provide effective isolation.

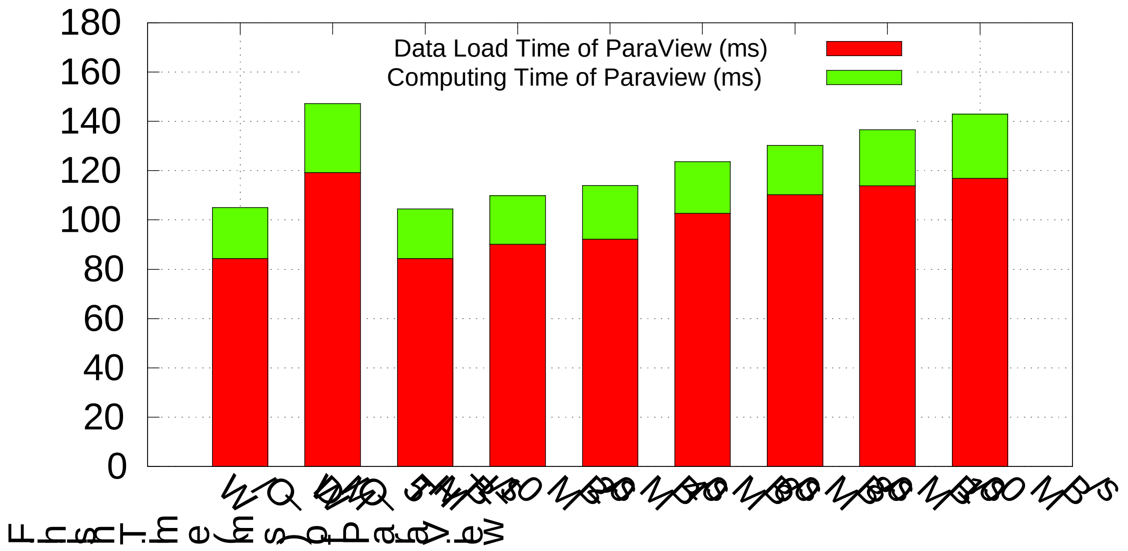
I/O Throttling on PFS



Throttle Rate to DFS Block I/O

I/O throttling middleware can effectively control the aggregate bandwidth of PFSs and introduces negligible overhead

I/O Throttling on Real Application



Throttle Rate to Competing Daemons' I/O

The finish time of ParaView is increasing as the I/O throttle rate of background daemons increasing.

Contents

Related Work

- OS-level virtualization:
    - Authors [1, 2, 3], have evaluated the overhead (CPU, memory and disk) of OS-level virtualization compared with the traditional hypervisor based virtualization.
    - Multilanes [4] builds an isolated I/O stack for eliminating contentions on shared kernel structures and locks, while applying OS-level virtualization to control the I/O of fast block devices (SSD).
- Resource allocation platform via OS-level virtualization:
    - Mesos [5] is a resource allocation platform for multiple users and multiple computing platforms such as Hadoop and MPI. Mesos takes advantage of OS-level virtualization (LXC) to provide cluster resource sharing (only CPU and memory) in a fine-grained manner.

Contents

Conclusion

- In this paper, we investigate the overhead and isolation of OS-level virtualization on block I/O control.
- The block I/O control of OS-level virtualization introduces less than 15% overhead in average.
- The weight functionality introduces at most 8% overhead and shows good performance isolation.
- The throttle functionality introduces low performance overhead but has limited performance on the isolation.
- The I/O throttling middleware can allocate PFS’s I/O to multiple users based on their priorities, with negligible overhead.

Acknowledgement

- The experiments of this work are conducted at the PRObE Marmot cluster.

Reference

- [1] Nikolaus Huber, Marcel von Quast, Michael Hauck, and Samuel Kounev. Evaluating and modeling virtualization performance overhead for cloud environments. In CLOSER, pages 563-573, 2011.
- [2] Stephen Soltesz, Herbert Potzl, Marc E Fiuczynski, Andy Bavier, and Larry Peterson. Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In ACM SIGOPS Operating Systems Review, volume 41, pages 275-287. ACM, 2007.
- [3] Miguel G Xavier, Marcelo Veiga Neves, Fabio D Rossi, Tiago C Ferreto, Timoteo Lange, and Cesar AF De Rose. Performance evaluation of container-based virtualization for high performance computing environments. In Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on, pages 233-240. IEEE, 2013.
- [4] Junbin Kang, Benlong Zhang, Tianyu Wo, Chunming Hu, and Jinpeng Huai. Multilanes: providing virtualized storage for os-level virtualization on many cores. In FAST, pages 317-329, 2014.
- [5] Benjamin Hindman, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica. Mesos: a platform for fine-grained resource sharing in the data center. In Proceedings of the 8th USENIX conference on Networked systems design and implementation, NSDI'11, pages 22-22, Berkeley, CA, USA, 2011. USENIX Association.

Thank you
&
Questions