Contestion-aware Virtual Machine Scheduling via Runtime Performance Monitoring
Chen Liu & Gildo Torres, Clarkson University, {cliu, torresg}@clarkson.edu

Problem
- Cloud computing pervasive nowadays
- OS scheduler treats VMs as regular processes
- Co-running VMs contend for resources
- Poor and unpredictable performance of co-running VMs

Key Idea
Propose to adaptively assist OS scheduler to monitor and manage co-running VMs:
- Use Hardware Performance Counters to collect runtime information of VMs
- Statistical model to capture execution phases
- Assist OS scheduler based on dynamic resource demand of VMs

Mapping VMs to Cores
Select mapping that maximizes HPC distance among co-located VMs:

Squared normalized weighted Euclidean distance:

\[ d(a, b) = \sum_{i=0}^{n} w_i \times (\text{norm}(a_i) - \text{norm}(b_i))^2 \]

- \(d(a, b)\): distance between VM_a and VM_b
- \(n\): number of HPCs
- \(w_i\): adjusted weight of HPC_i
- \(\text{norm}(a_i)\): normalized value of HPC_i of VM_a

Performance Comparison

Coscheduling Impact on Performance

Runtime Degradation Relative to solo

Co-scheduling Impact on Performance

Runtime Degradation Relative to solo

Technical Approach
- **Platform**: Kernel-based Virtual Machine (KVM)
- **Hardware Events**: L2_M_LINES_OUT_STORES RETIRED, L1D_STORES, CPI_REF
- **Runtime Model**: Using memory access to capture dynamic behavior of VMs

Conclusions
- Default OS scheduler unaware of potential contention on CPU resources among VMs
- Co-scheduling directly affects performance
- Contention-aware hypervisor can assist OS scheduler to mitigate contention for resources among VMs and therefore reduce performance degradation

Intel® HPC Developer Conference 2017