UNM SCHOOL of ENGINEERING
Department of Computer Science

Quantifying Scheduling Challenges for Exascale System Software

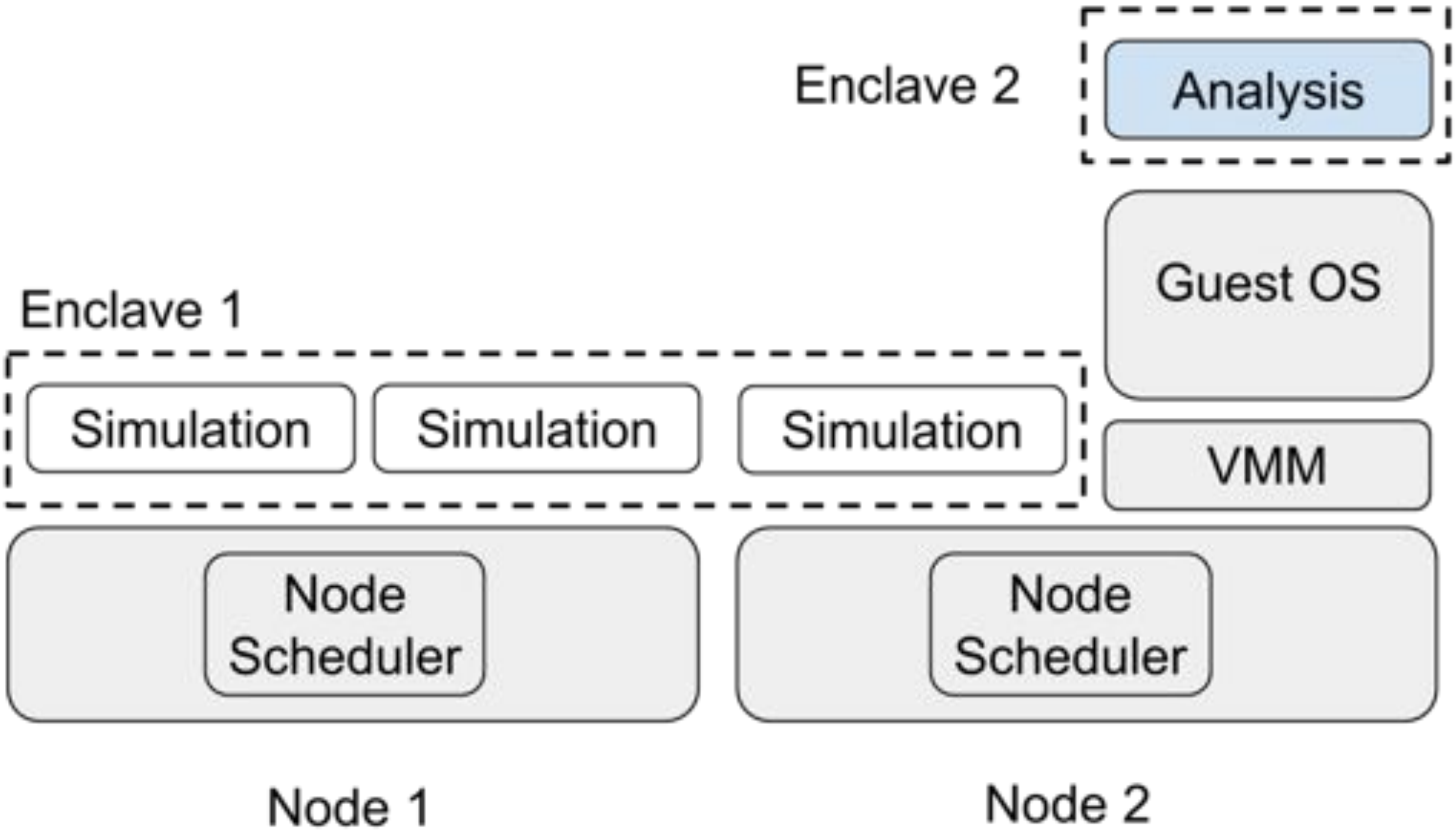
Oscar H. Mondragon, Patrick G. Bridges
University of New Mexico

Terry Jones
Oak Ridge National Lab

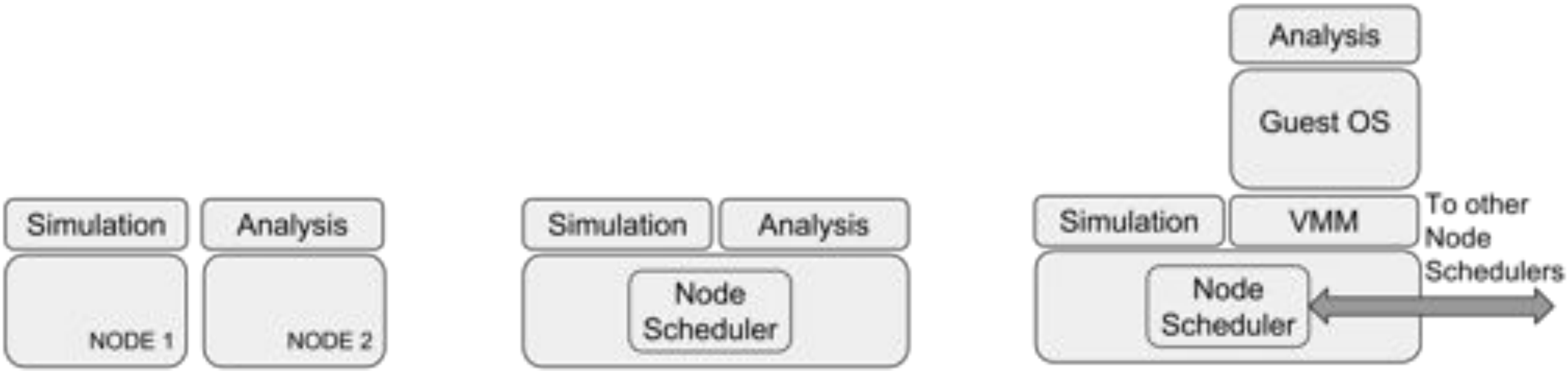
Motivation

- Coupled HPC codes becoming prevalent (e.g., GTC + PreData, LAMMPS + Bonds, CTH + ParaView)
- New scheduling challenges given the number of constraints and performance trade-offs
- Target case: Simulation application with coordination (e.g., gang scheduling) and analytics co-location
- Need to quantify the performance cost of co-location and propose new potential scheduling solutions

Exploratory Analytics Example



Resource Allocation Approaches



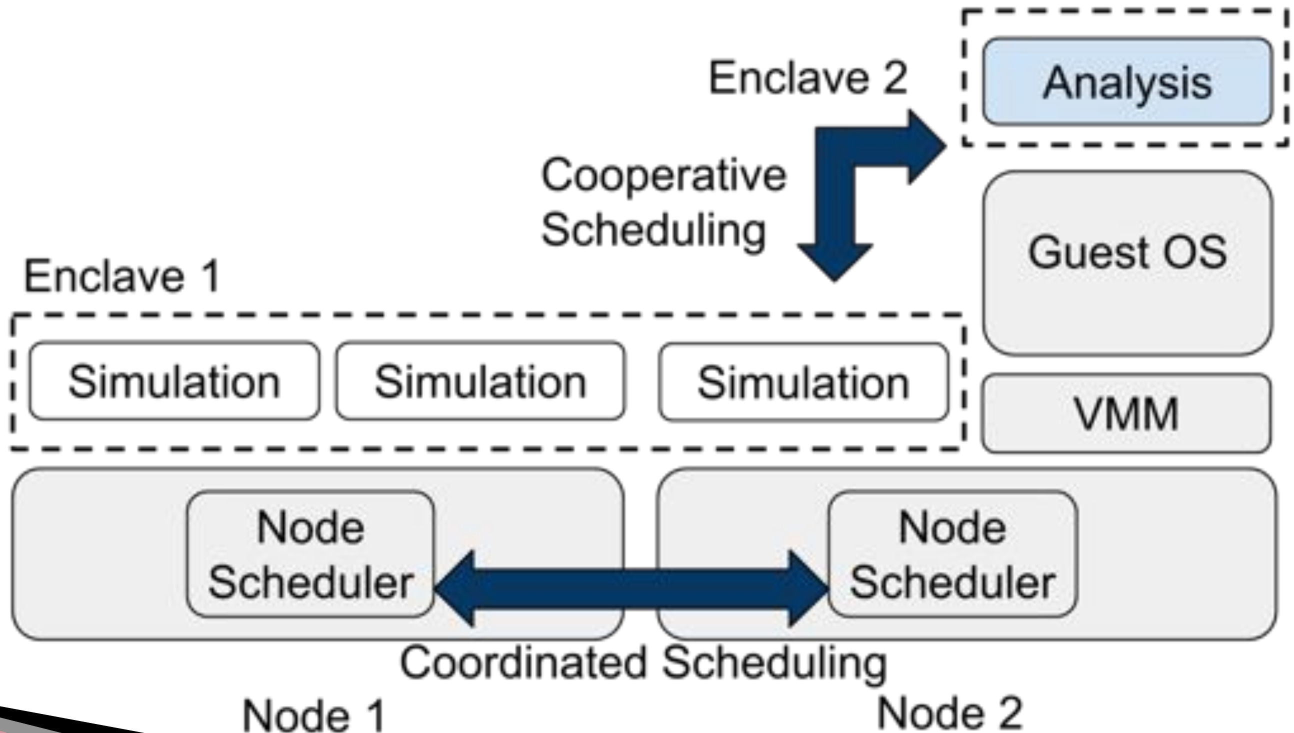
(a) Space-Shared Simulation and Analysis

(b) Time-Shared Simulation and Analysis Running Natively

(c) Time-Shared. Guest OSes with single workloads

Scheduling Challenges

- Node-level Resource Allocation
- Intra/inter node synchronization/coordination
- Co-location of Cooperative Enclaves



Evaluation of Potential Solutions

- Node-level Resource Allocation
  - Explicit Numerical Optimization
  - Our formulation: Constrained Binary Quadratic Programming

- Combined cooperative and coordinated scheduling
  - Build on earliest deadline first (EDF)-based gang scheduling
  - Verify suitability of basic approach to gang scheduling
  - Evaluate additional impact of co-location

Related Work

- Scheduling via Numerical Optimization
  - Convex Optimization: PACORA (Bird, HotPar 2011)
  - Genetic algorithms (Omara, JPDC 2010)
  - Bin-Packing Heuristics (Zapata, 2005)
- Intra/inter node coordinated scheduling
  - Real time scheduler approaches: Vsched (Lin, SC 2005)
  - Clock synchronization techniques (Jones, 2013)
- Co-location of Cooperative Enclaves
  - Interference-aware runtime systems (Jones, SC 2003)
  - User-level interfaces for CPU time sharing of cooperative applications: Goldrush (Zheng, SC 2013)

Node-level Resource Allocation

- Constrained optimization
  - Convex, continuous problems: Inexpensive solution
  - Non-convex or discrete problems: NP-hard
- Goal: Map Palacios virtual cores to physical cores
- Objective: Minimize interference between virtual cores
- Difficult formulation problems
  - Even simple objectives like this are non-convex!
  - Constraints like “one virtual core per physical core” are discrete!
- Result: Non-Convex Binary Quadratic Program
  - Expensive to solve full problem at once
  - Decompose hierarchically to reduce computational complexity

Binary Quadratic Programming (BQP)

- Multilevel Formulation
  - Level 1: VMs to Sockets
  - Level 2: VCs to NUMA domains
  - Level 3: VCs to Physical cores
- Constraints:

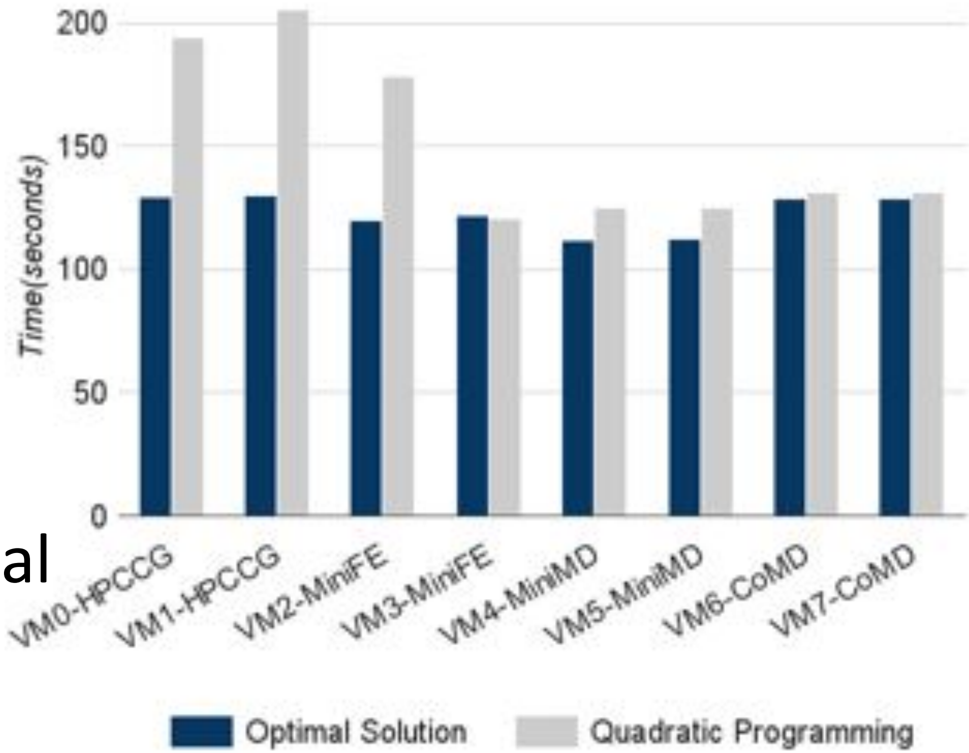
$$\forall i \epsilon V \sum_{j=0}^{N_p} x_{ij} = 1 \qquad \forall j \epsilon P \sum_{i=0}^{N_v} U_{ij} x_{ij} \leq 100$$

- Example: Level 1 Objective Function:

$$min \sum_{u=0}^{Nvm} \sum_{v=0}^{Nvm} \sum_{s=0}^{Nsk} \sum_{t=0}^{Nsk} (I_{VMS}(u,v) S(s,t)) x_{us} x_{vt}$$

BQP often close to optimal schedule

- Goal: Compare our numerical optimization based on a non-convex formulation against optimal solution
- Problem: Map 8 VMs to a 64-core machine with 8 NUMA domains
- Setup
  - Each VM has 8 VCs
  - Each VM runs a 8-procceses miniApp
- Result: near-optimal in 5 of 8 cases, far from optimal in other cases

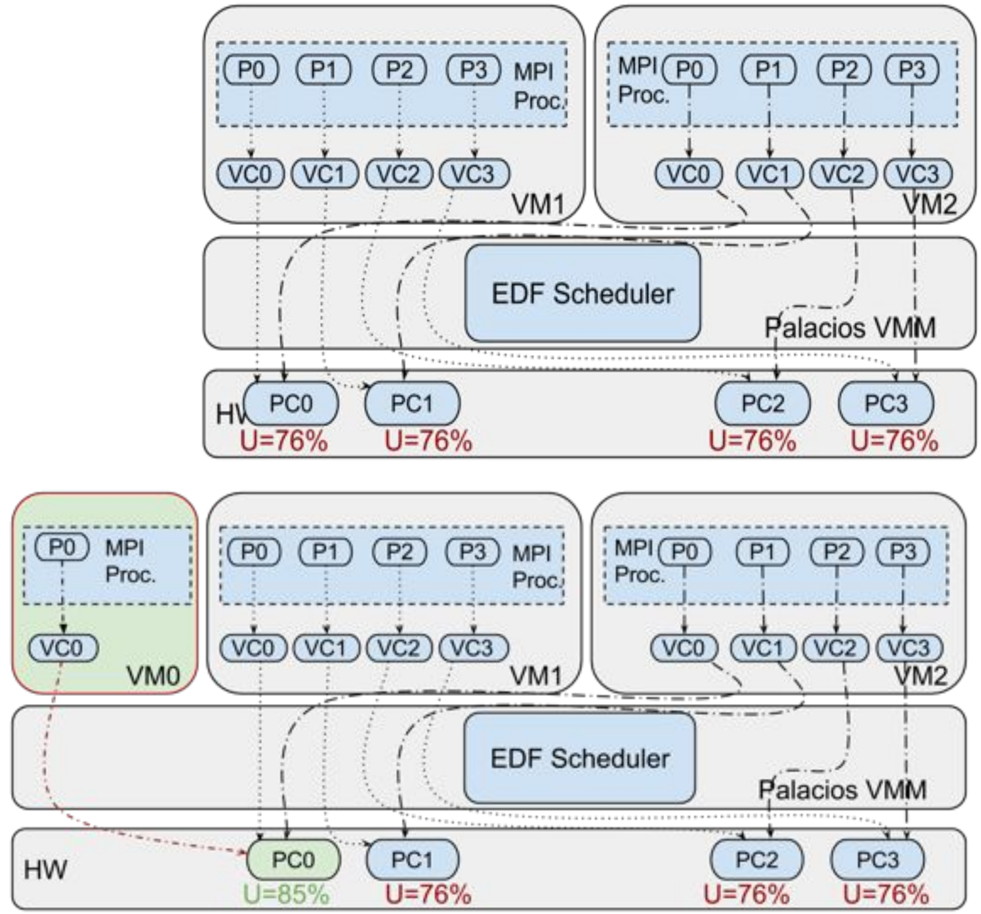


Combined cooperative and coordinated scheduling

- Solution explored: EDF (Earliest Deadline First)-based gang scheduler + co-located cooperative application
- EDF Scheduler added to Palacios VMM
- Experiment 1: verify EDF-based gang-scheduling
- Experiment 2: Gang-scheduled simulation + co-located analytics
  - Create one additional VM on one core
  - Change in utilization could impact quality of gang scheduling

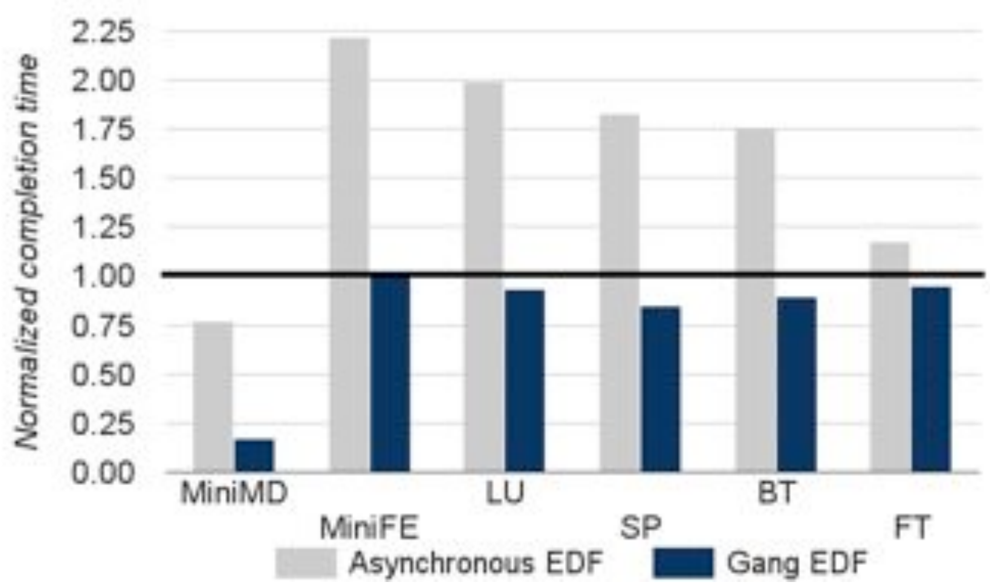
Experimental Setup

- VCs belonging to a VM have same real-time schedule
- Each VM runs a 4-Processes MPI benchmark
- Co-located analytics should use only idle CPU time



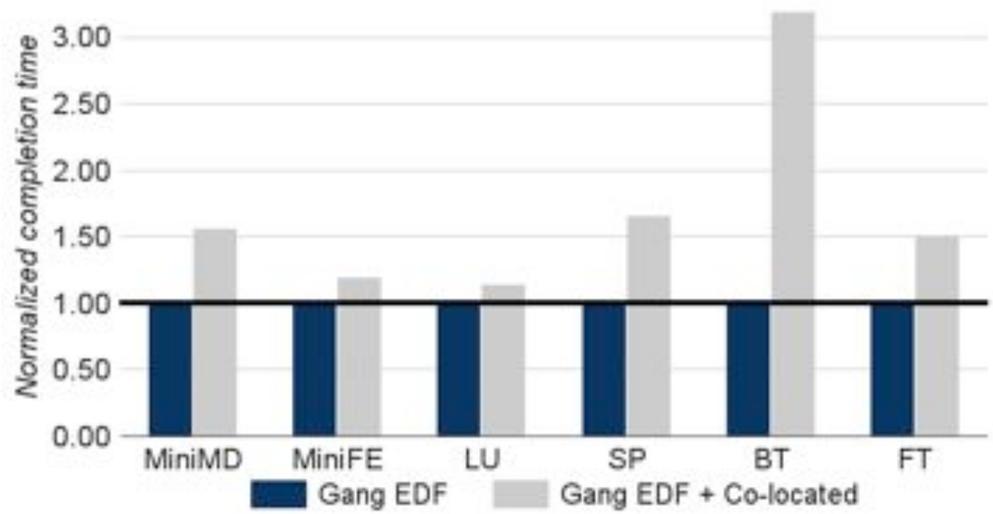
Basic Real-time Gang Scheduling Works

- Control granularity of synchronization with length of deadline
- This also increases scheduling overheads
- Used relatively long deadlines in this case (~130ms)



Co-location counters Gang Scheduling

- Applications lose all gang scheduling benefits
- BT an outlier due to additional cache effects (address via Goldrush-style techniques)
- Need to new techniques to preserve benefits of gang scheduling



Conclusion

- Numerical optimization solutions show some potential to solve the problem of resource allocation however it is not clear if they are sufficient at larger scales
- Current real-time scheduling approaches like EDF scheduling provide gang scheduling capabilities
- Enhancements to this scheduling approaches are needed to avoid performance degradation in the gang when cooperative applications are co-located

Future Work

- Efficient multi-objective optimization approaches that consider cooperative behavior and additional optimization criteria are potentially of high impact
- Enhanced real-time scheduling approaches could provided gang scheduling + BW reclaiming mechanisms
- Lightweight OS and user level interfaces for cooperative and coordinated scheduling
- Coordination/synchronization mechanisms between node-level schedulers

Acknowledgements

This work was supported in part by the 2013 Exascale Operating and Runtime Systems Program from the DOE Office of Science, Advanced Scientific Computing Research, under award number DE-SC0005050, program manager Sonia Sachs, and by the Colciencias-Fulbright Colombia and The Universidad Autonoma de Occidente through the Caldas scholarships program.

UNM SCHOOL of ENGINEERING
Department of Computer Science

Thank you!
Questions?

Contact: omondrag@cs.unm.edu