# Challenges in solving Large Scale Optimization Problems

**HEMERA – Challenge COPs**


Leaders: B. Derbel and N. Melab
Participants: T-T VU (PhD Hemera), A. Ali (PostDoc Hemera), M. Djamai (PhD Univ. Lille 1)

**DOLPHIN – INRIA Lille Nord Europe**

# General Context

## Combinatorial Optimization

- Applications: Logistics, Supply Chain, Telecommunications, Clouds, Green-IT, etc.
- NP-hard problems
- Resolution methods are computing intensive

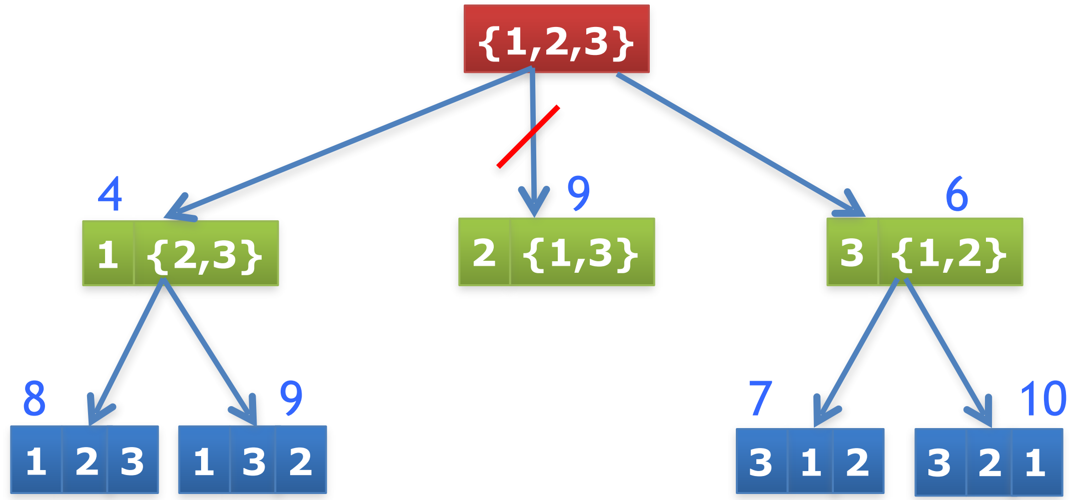
## Computing Resources

- Aggregated resources: clusters, grids, clouds, etc.
- New architectures: multi-cores, many-cores, etc.
- Impressive computing capability (in theory)

## General objective

- Solve Combinatorial Optimization Problems efficiently on large scale computing resources

# Branch-and-Bound



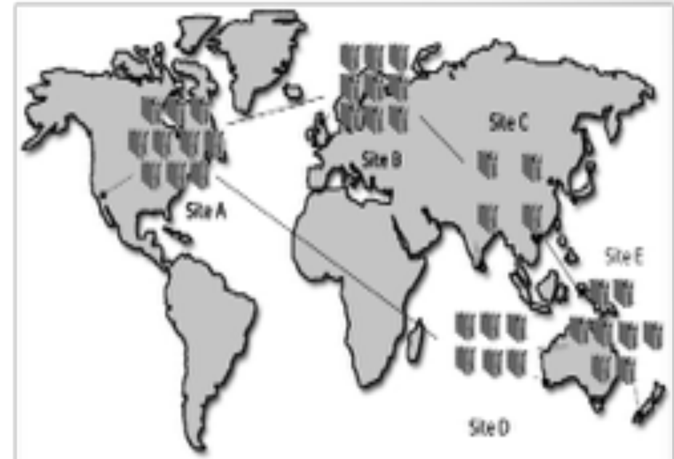

- Branching: divide a problem to several sub-problems
- Bounding: calculate the *estimated* optimal solution
  - lower/upper bound
- Select: tree exploration strategy (DFS, BFS, etc)
- Pruning: eliminate unpromising branches

# Large Scale Heterogeneous Systems

Multi-cores

GPUs

Cluster(s)




Grid + P2P

- Large scale systems
- Heterogeneity
  - Node-level: compute power, programming paradigm, etc
  - Network-level: latency, bandwidth, etc

# Contributions

- Efficient parallel B&B **load balancing**
- Efficient parallel B&B on **node-heterogeneous** systems
- Efficient parallel B&B on **link-heterogeneous** systems

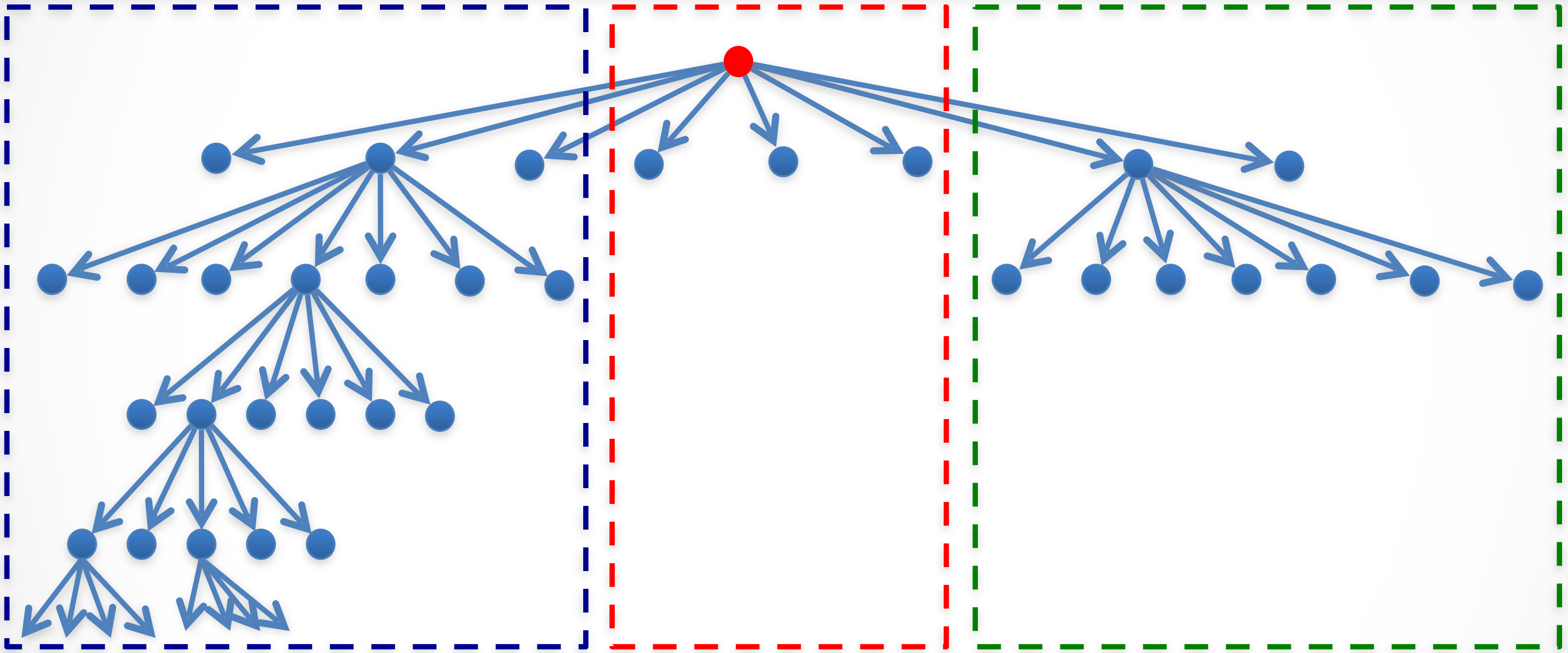
# Contributions

- Efficient parallel B&B load balancing

- Efficient parallel B&B on **node-heterogeneous** systems

- Efficient parallel B&B on **link-heterogeneous** systems

# Irregularity of B&B



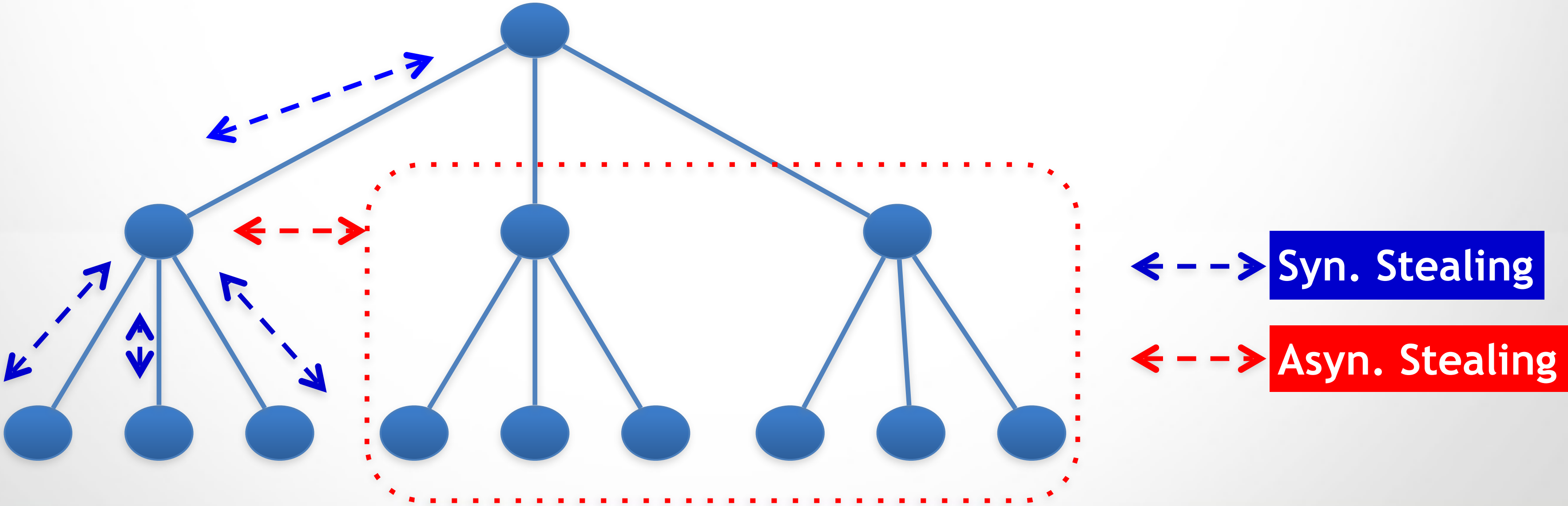
➔ Workload of processing unit varies dynamically

➔ **Work stealing** is a reference approach

# Tree-based B&B Work Stealing



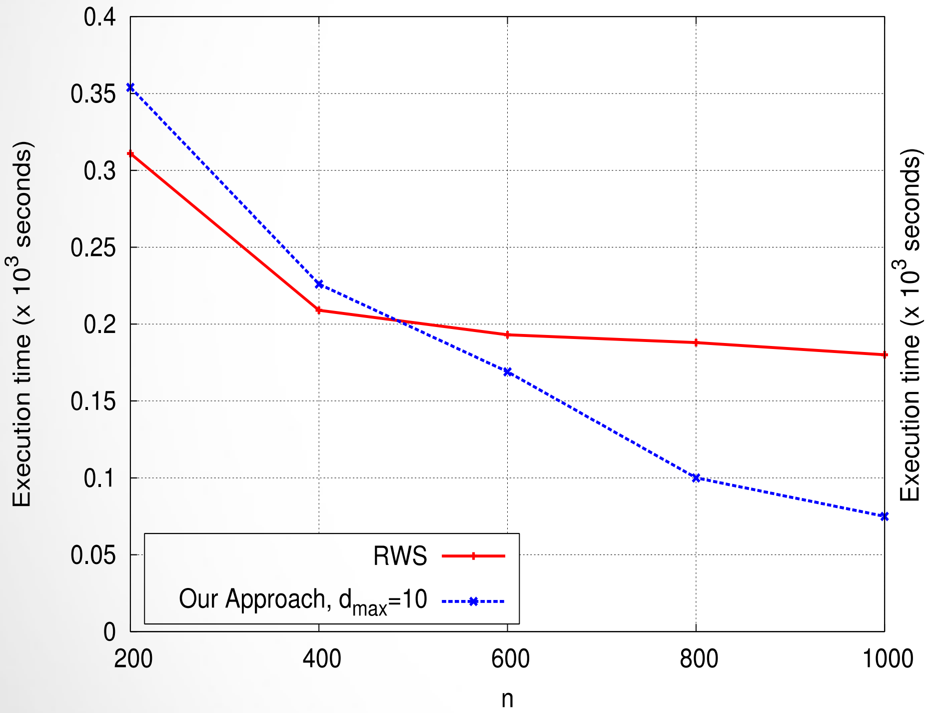
- **Tree-based stealing strategy:** 2 steals in parallel
  - **Synchronous steals** to children or parent
  - **Asynchronous steals** to remote neighbors
    - Attempt to cluster idle peers
  - **Amount of work is adjusted distributively based on subtree sizes**

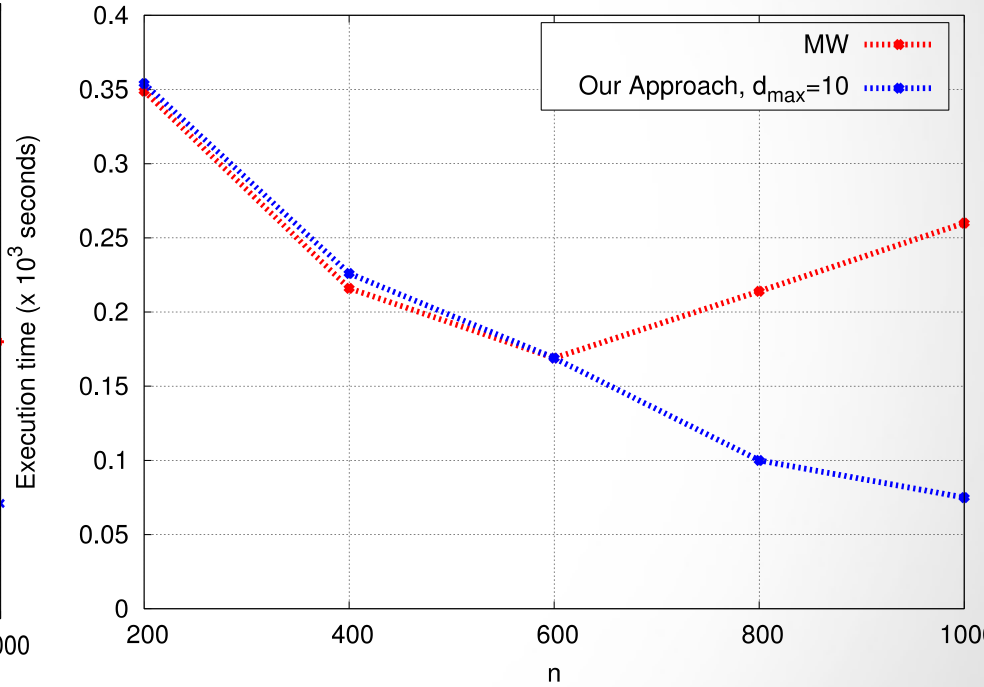


# Experimental Evaluation

- Application Settings
  - **Taillard's Flowshop Instances (Ta20*20)**
    - Permutation FSP: 20 jobs on 20 machines
  - **Generic UTS benchmark**
- Baseline Algorithms
  - H-MW: Hierarchical Adaptive MW (B&B specific) [Bendjoudi et al., FGCS'12, IEEE TC'13]
  - MW: Master-Worker (B&B specific) [Mezmaz et al., IDPDS'07]
  - RWS: (distributed) Random Work Stealing [Dinan et al., SC'09]

# Our Approach vs RWS vs MW
## Large Scale (1000 peers)



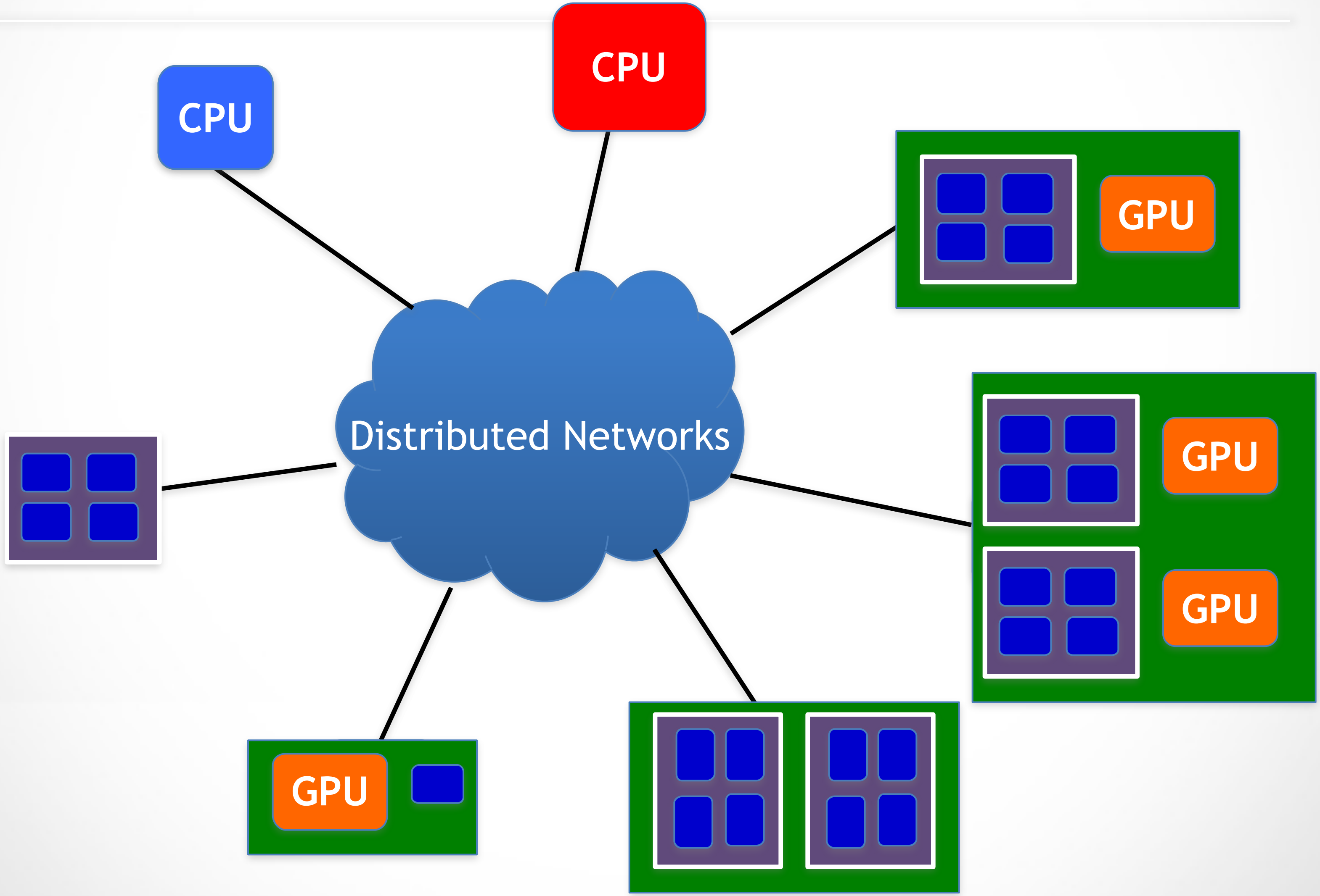

- MW suffers from the bottleneck when scaling the system
- RWS suffers fine-grain parallelism in large scales

# Contributions

- Efficient parallel B&B **load balancing**
- Efficient parallel B&B on **node-heterogeneous** systems
- Efficient parallel B&B on **link-heterogeneous** systems

# Towards heterogeneous B&B





# Towards heterogeneous B&B

- How to profit from current **node-heterogeneous** computing platforms in B&B computations?

- Three main challenges:
  - How to map B&B and hardware parallelism?
  - How to deal with B&B workload irregularity?
  - How to deal with huge differences in compute power?

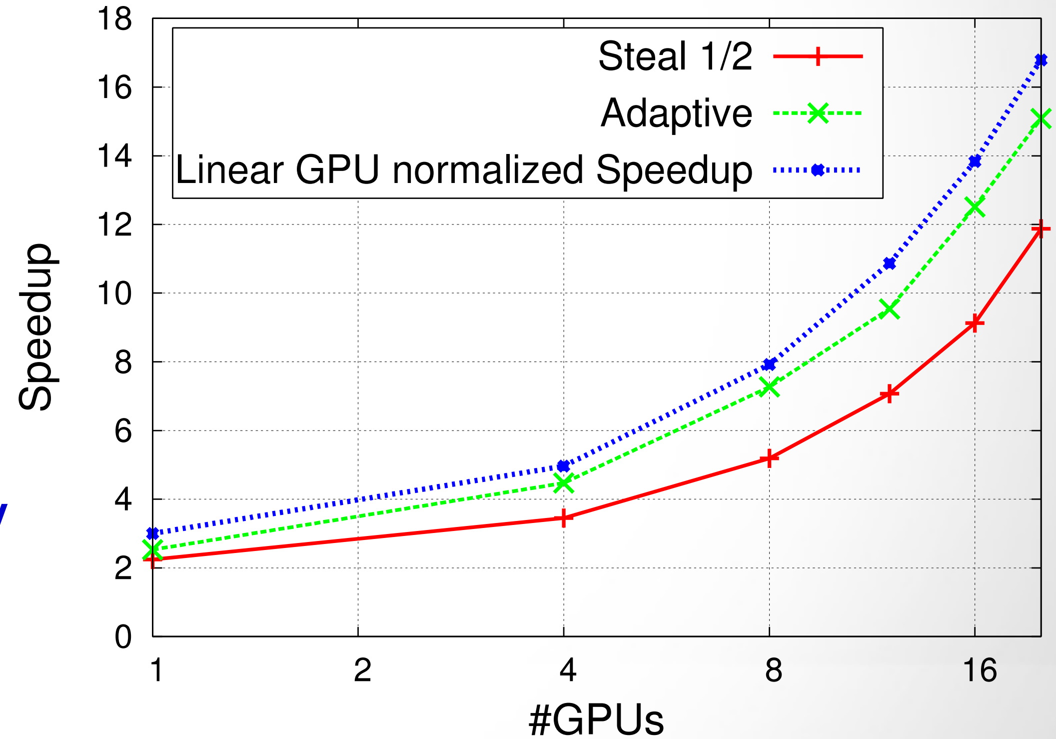
# Heterogeneous parallel B&B

- **The 2MBB approach**
  - Multi-CPU Multi-GPU B&B

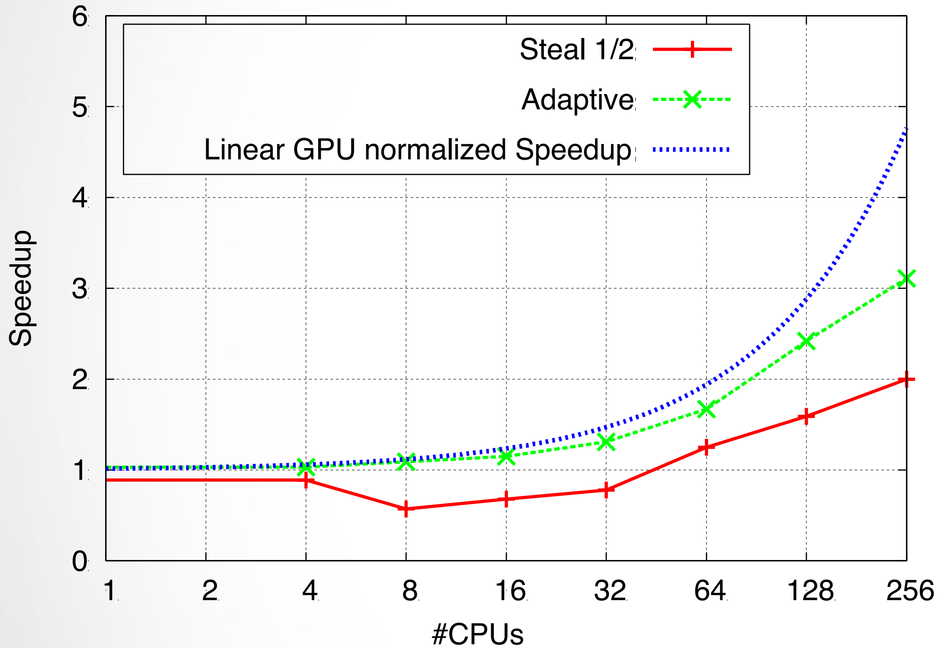
- **The 3MBB approach**
  - Multi-Core Multi-CPU Multi-GPU B&B
    - host-device parallelism in a single CPU-GPU
    - Adaptive workload transfer
    - Hybrid stealing in multi-core systems
    - Lock-free work queues

# 2MBB: Experimental Results

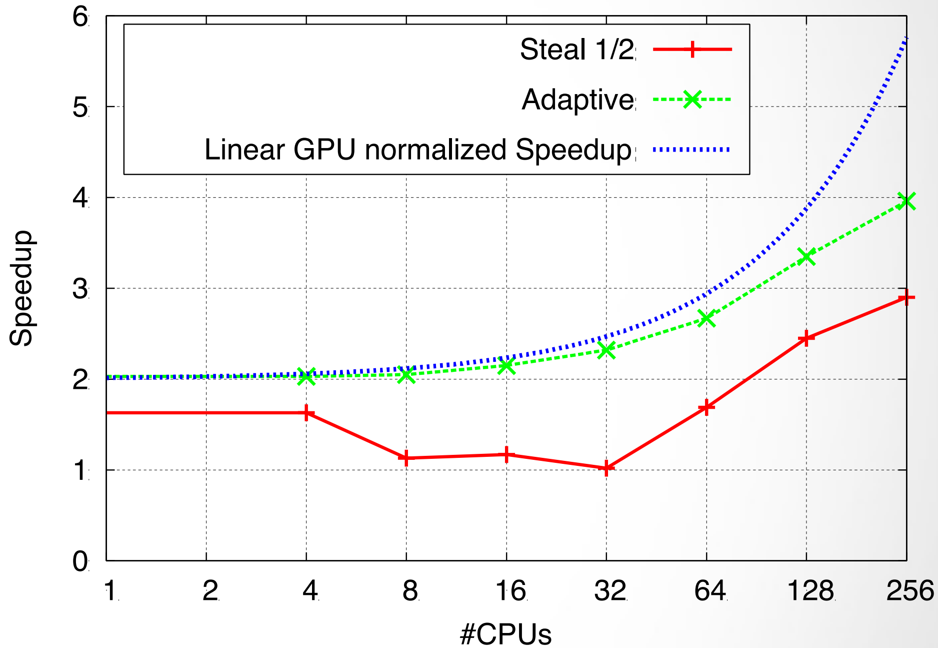
- CPUs: fixed to 128
  - 64 CPUs of 2.27 GHz
  - 64 CPUs of 2.5 GHz
- GPUs: scale up to 20
  - ½ GPUs at full capacity
  - ¼ GPUs at half capacity
  - ¼ GPUs at quarter capacity



# 2MBB: Experimental Results




**1 GPU**




**2 GPU**

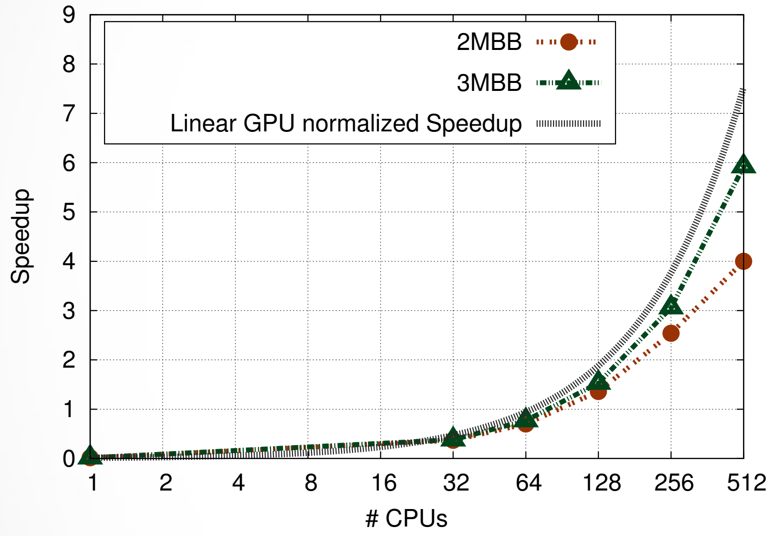
- Moderate scales:
  - We scale close to the linear speedup
  - Baseline suffers from node-heterogeneity
- Largest scales: we are still far from the linear speedup

# 3MBB: Experimental Results

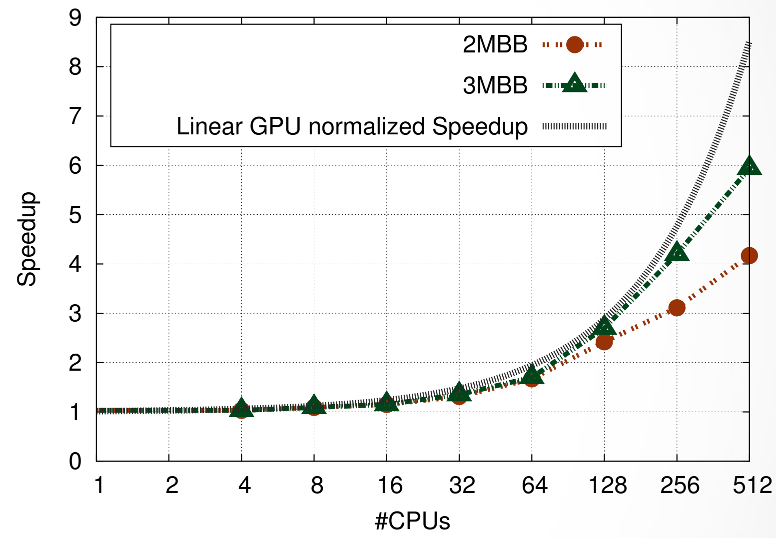


Grid'5000

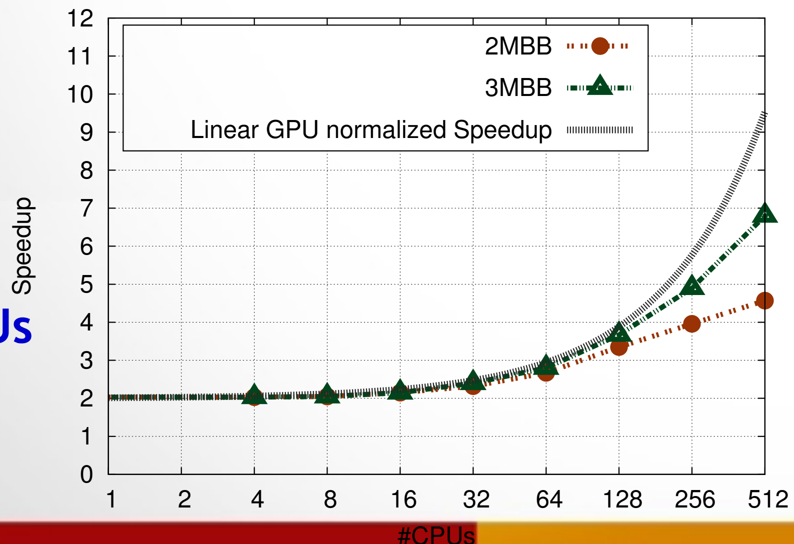
0 GPU



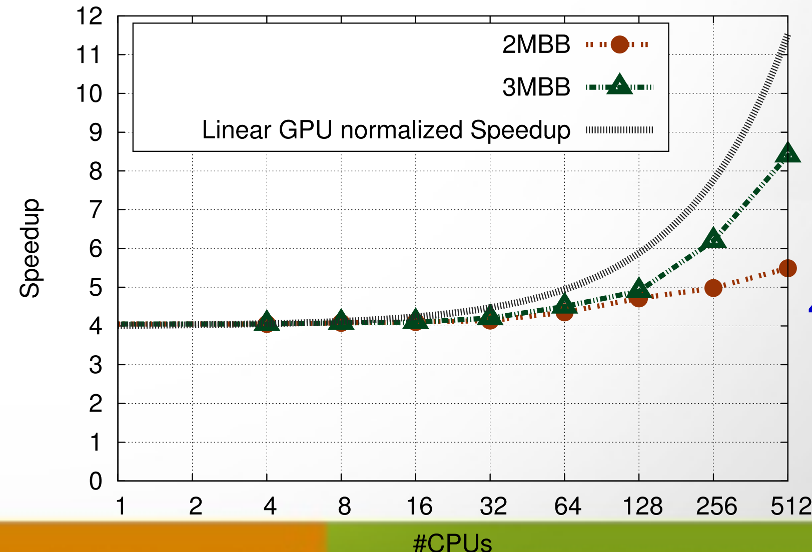

1 GPU




2 GPUs




4 GPUs

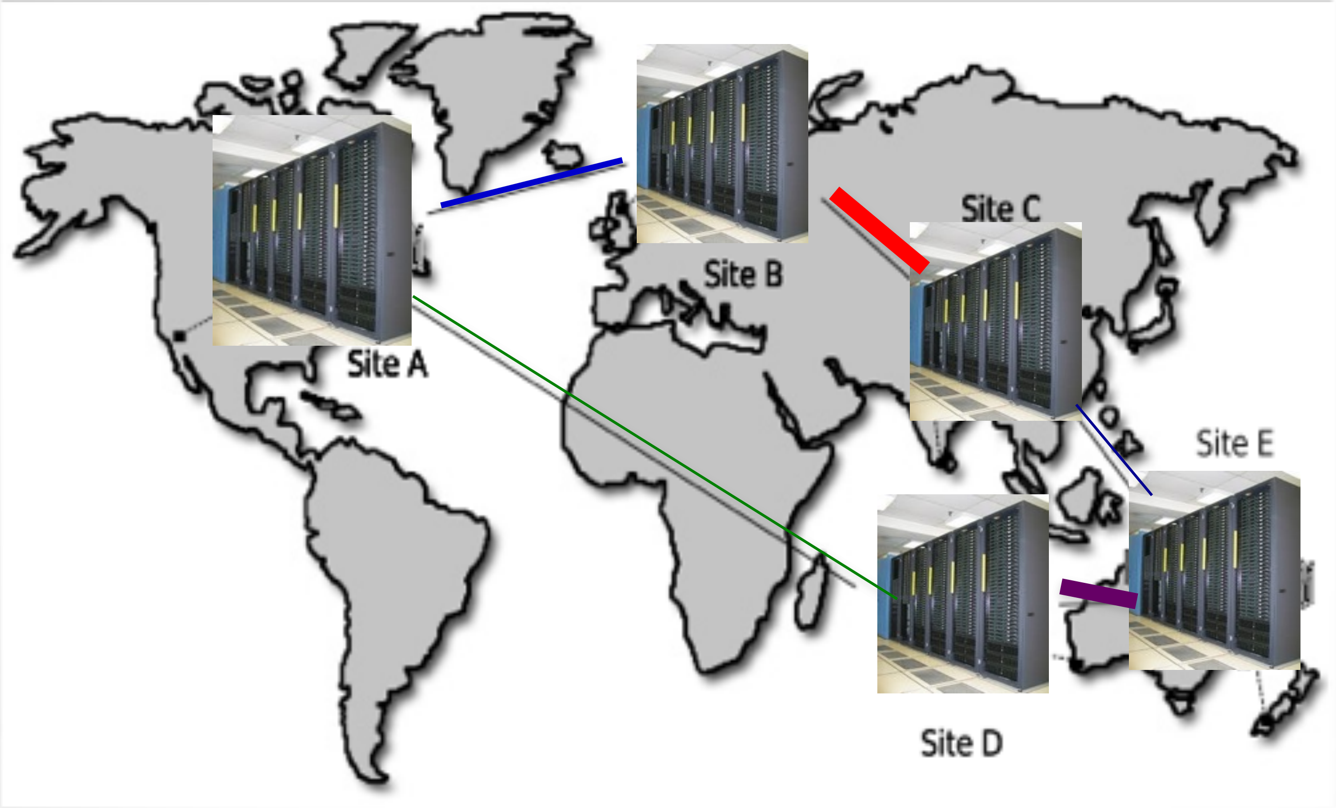


# Contributions



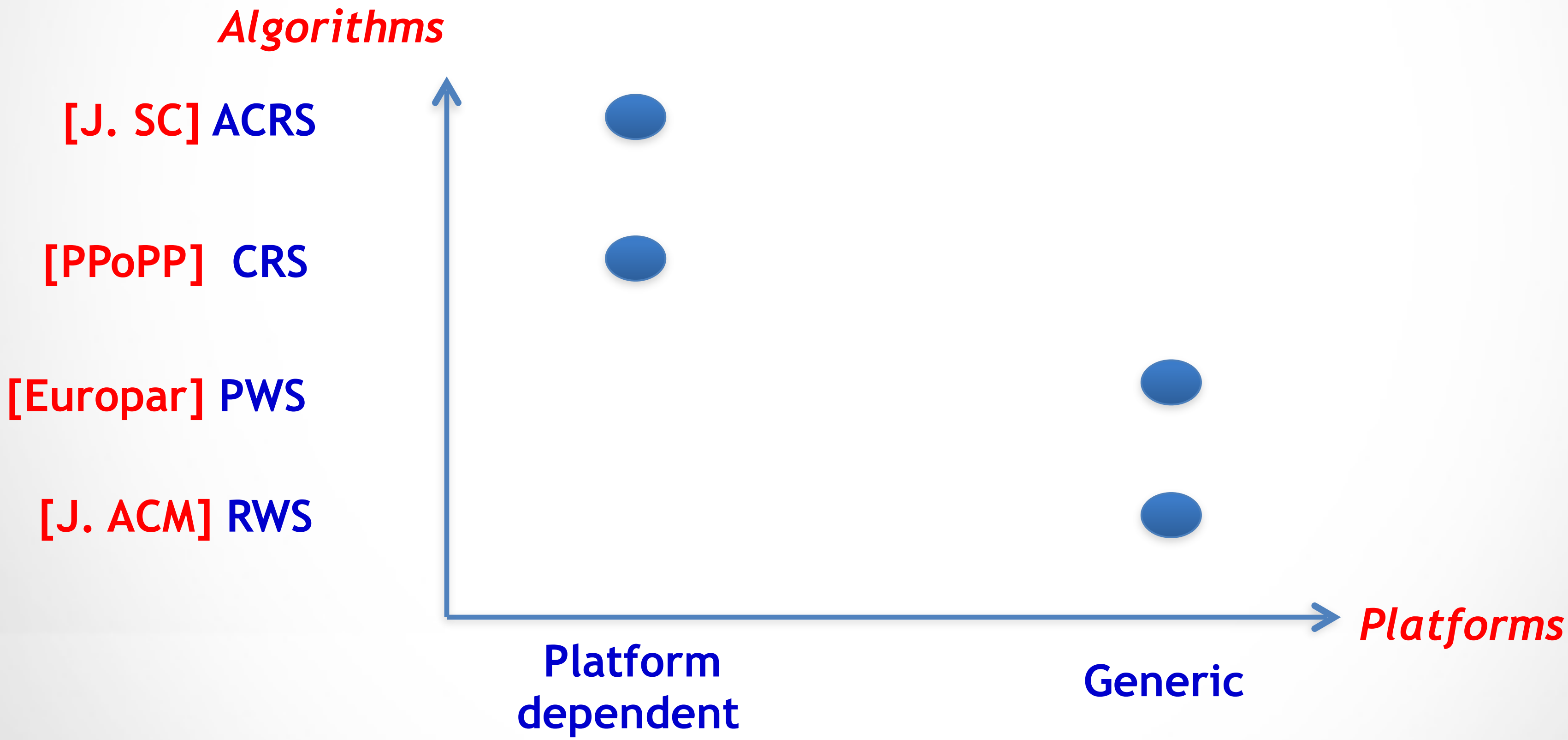
- Efficient parallel B&B **load balancing**
- Efficient parallel B&B on **node-heterogeneous** systems
- Efficient parallel B&B on link-heterogeneous systems

# Heterogeneous links




- Steal requests through WAN links are expensive

# State-of-the-art approaches



# Link Heterogeneous Work Stealing

- **Local Steals:** based on a preference neighbors and a non-uniform adaptive probability

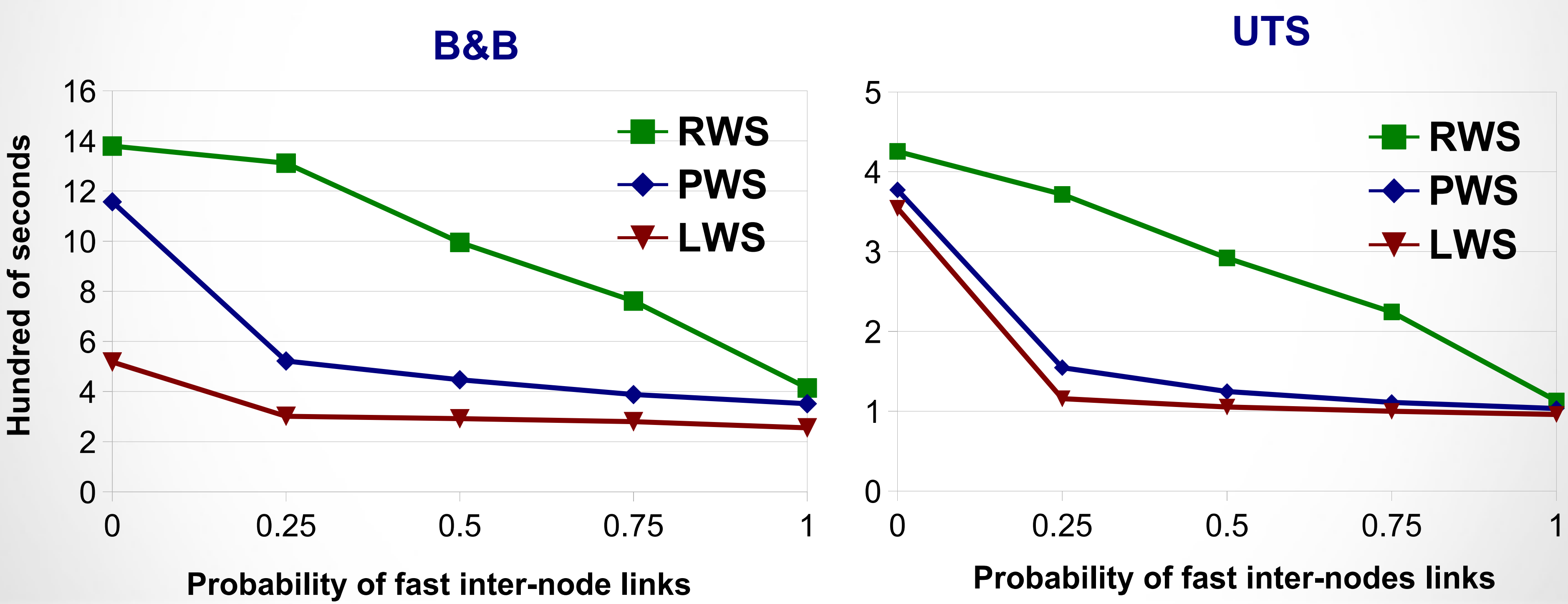
- **Learn local neighbors and remote neighbors at runtime**
  - **K-Means clustering** to return 2 sets of neighbors

- **Remote steals: controlled by a timing window**
  - If the window expires and no work found remote steal is enabled
- **Window size controlled adaptively (additive Increase Multiplicative Decrease)**

# Performance Assessment

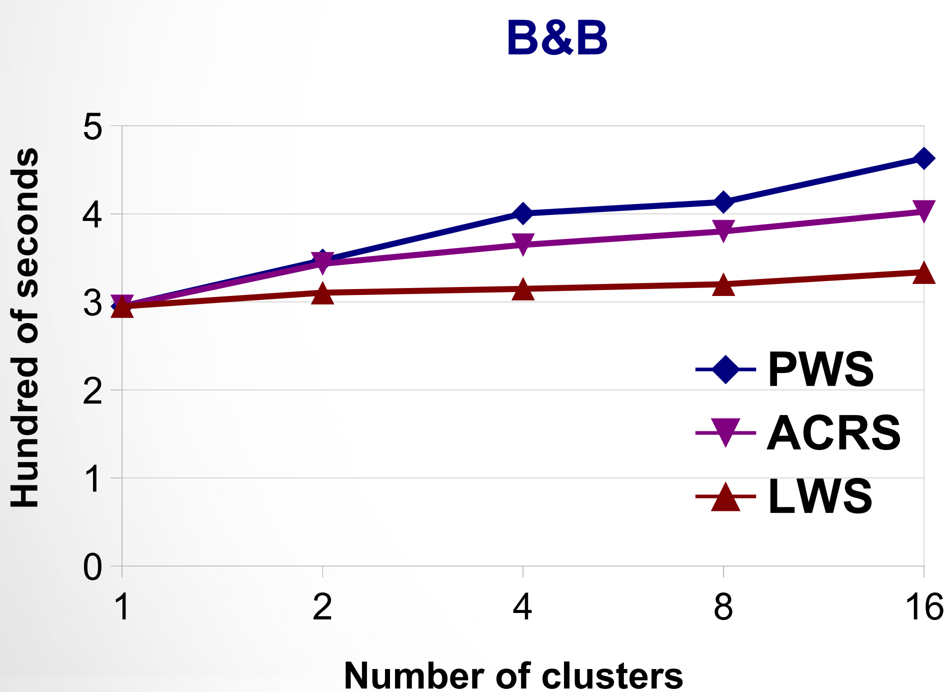
- Experimentation methodology: Emulation
  - Deploy Distem on top of Grid’5000
  - Network configuration is artificially modified by Distem
- Broad range of network configurations
  - Flat: **n-level communication** hierarchy
    - Latency between peers
  - Grid: **two-level communication** hierarchy
    - Latency between clusters
    - Number of clusters

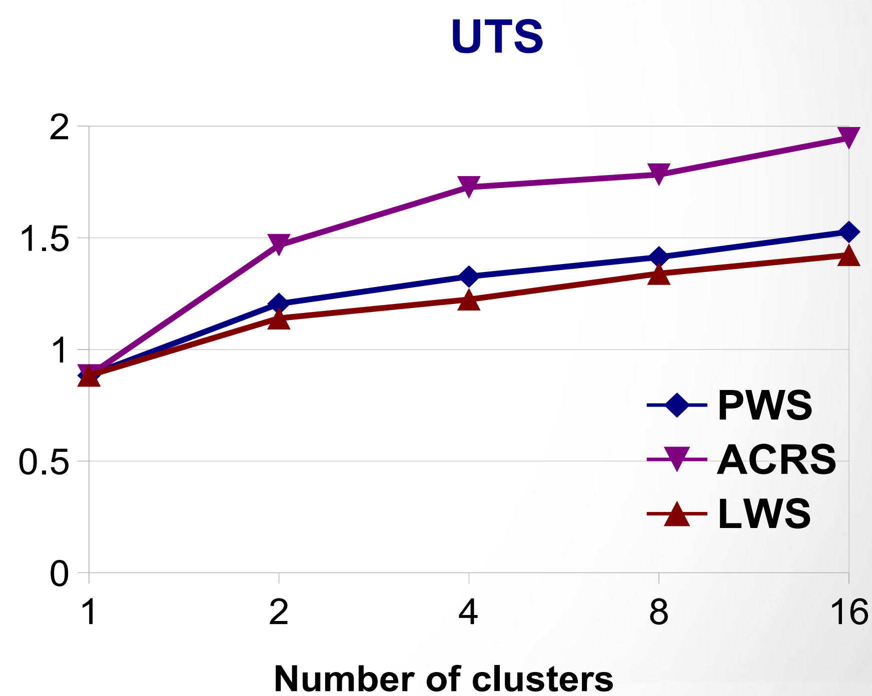
# Flat Configuration




- LWS improves up to 40%

# Grid Configuration




- PWS and ACRS performance depends on application
- LWS is platform and application-independent

# Conclusion

- Design and experimental evaluation of new parallel B&B algorithms for large scale heterogeneous environments

- In the future
  - Investigate **more complex compute systems**
  - Investigate **more complex optimization problems** and other algorithmic paradigms

# Thank You !

## Questions?