# Adaptive Runtime Partitioning of AMR Applications on Heterogeneous Clusters

Shweta Sinha & Manish Parashar

Presented By Jyoti Batheja

The Applied Software Systems Laboratory

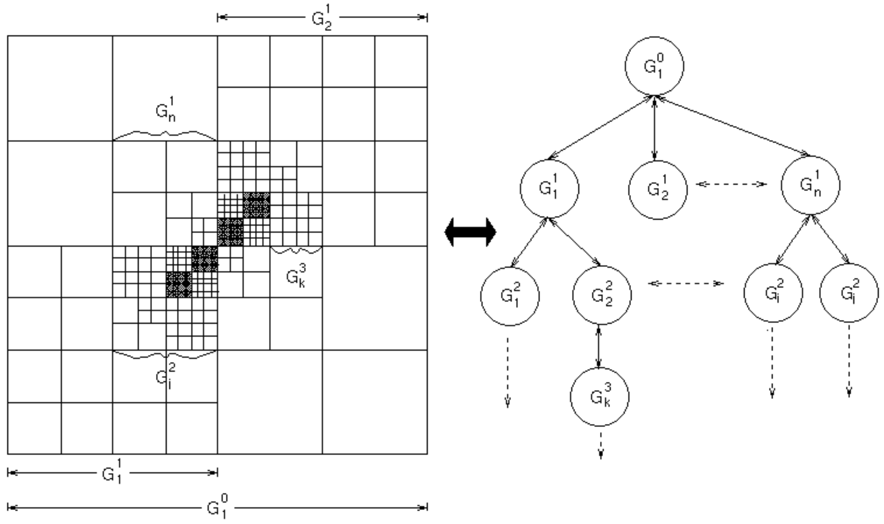
ECE/CAIP, Rutgers University

www.caip.rutgers.edu/TASSL

# Introduction

- Objective
  - *Develop a “system-sensitive” partitioning mechanism for SAMR applications that uses current system state of the networked computing environment to partition adaptive grid hierarchies*
- Approach
  - *Monitor resources of computing nodes*
  - *Compute relative capacities of nodes*
  - *Perform system sensitive partitioning*

# Structured Adaptive Mesh-Refinement



**Adaptive Mesh Refinement**

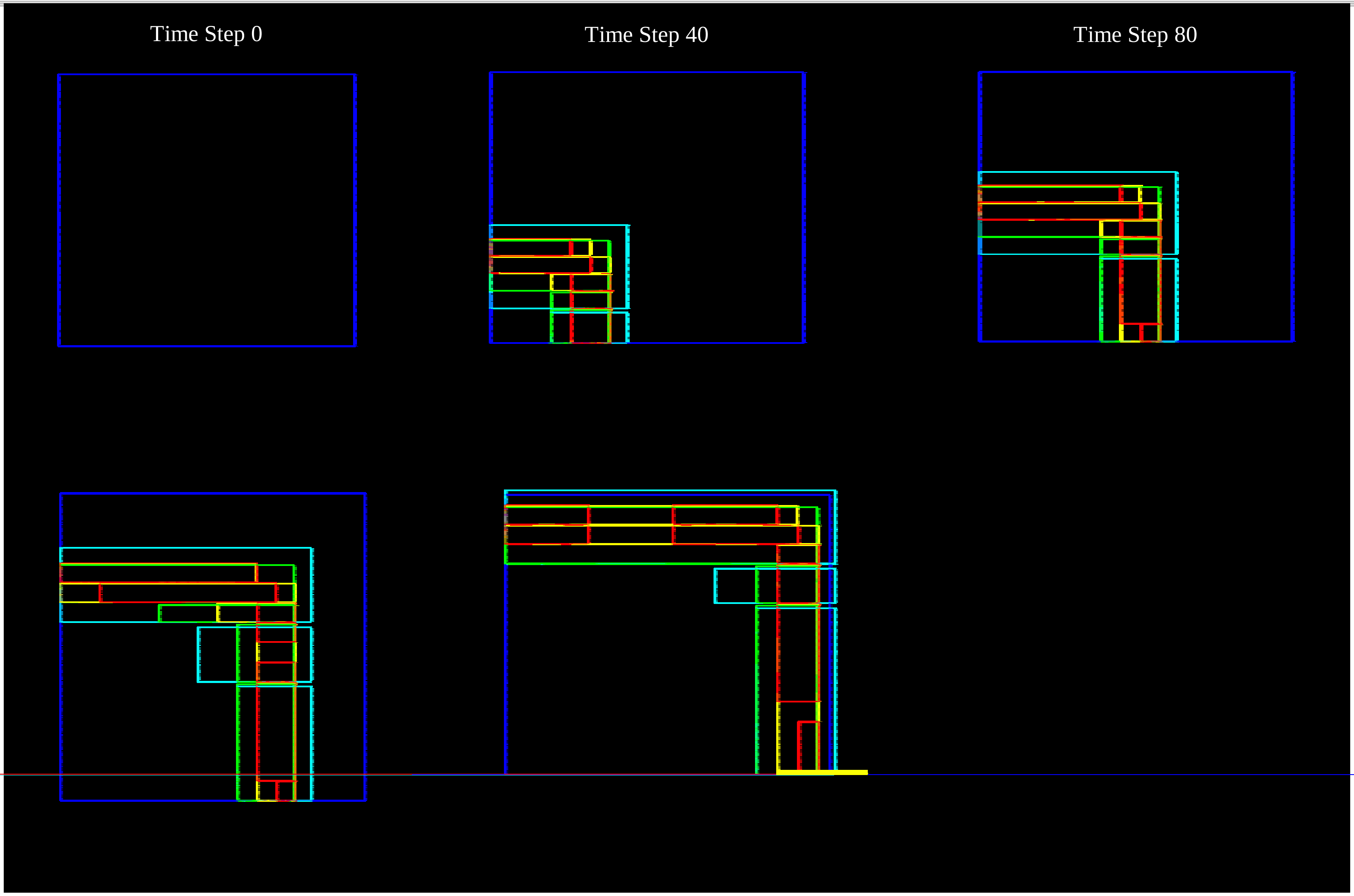
•Start with a base coarse grid with minimum acceptable resolution

• Tag regions in the domain requiring additional resolution and overlay finer grids on the tagged regions of the coarse grid

• Proceed recursively so that regions on the finer grid requiring more resolution are similarly tagged and even finer grids are overlaid on these regions

• Resulting grid structure is a dynamic adaptive grid hierarchy

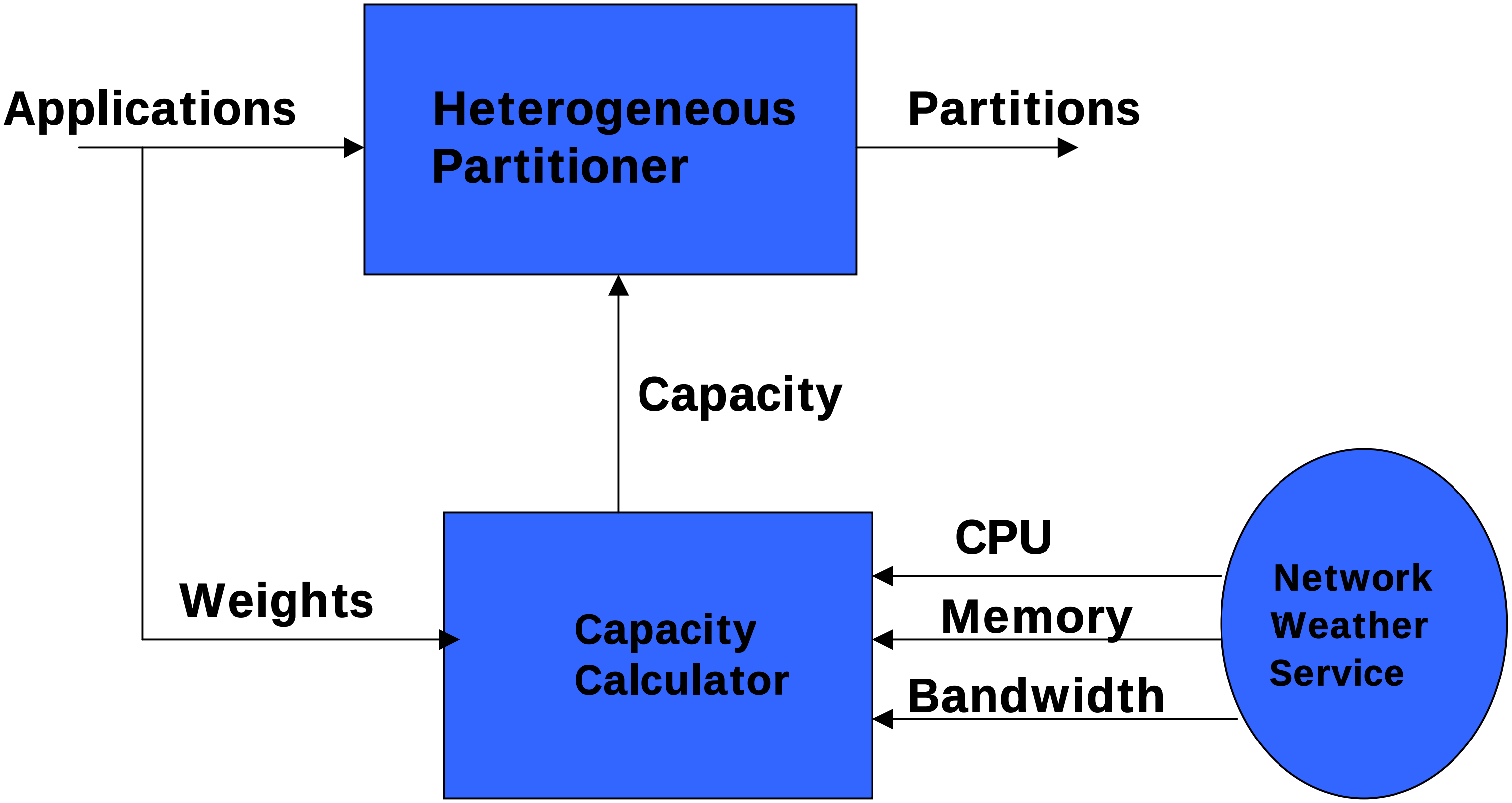
# AMR Grid Structure (2D Example)



# Partitioning Adaptive Grid Hierarchies

- Balance load and…
  - *Expose available parallelism*
  - *Minimize communication overheads*
    - *Inter-level prolongations/restrictions*
    - *Intra-level “ghost” communications*
  - *Enable dynamic load redistribution with minimum overheads*
- Parallel AMR costs
  - *Communications*
    - *intralevel “ghost” communication*
      - *along the surface of each block*
    - *interlevel prolongation/restriction communications*
      - *gather/scatter between parents/children*
  - *Grid recomposition*
    - *grid refinement/coarsening*
    - *redistribution and load-balancing*
    - *prolongation*
    - *data-movement*

# System Sensitive Partitioning



# Resource Monitoring Tool

- System characteristics determined at run-time using the Network Weather Service (NWS) from UCSD.
- NWS monitors:
  - *Fraction of CPU time available*
  - *End-to-end TCP network bandwidth*
  - *Free memory*
  - *Amount of space unused on disk*
- Predictive models
- http://nws.npaci.edu/NWS

# Cost Model

- For computing node k, let:
  - $p_k$: *CPU available*
  - $m_k$: *Memory available*
  - $b_k$ : *Bandwidth available*
- Then, the relative CPU availability of node k is:

$$P_k = \frac{p_k}{\sum_{i=1}^{K} p_i}$$

# Capacity Metric

- Relative capacity of node k can be written as:

$$C_k = w_p P_k + w_m M_k + w_b B_k$$

- where $w_p, w_m, w_b$ are the weights associated with relative CPU, Memory, and Bandwidth availabilities, respectively, where

$$w_p + w_m + w_b = 1$$

# Capacity Metric

- Using system information a relative capacity metric is computed for each processor

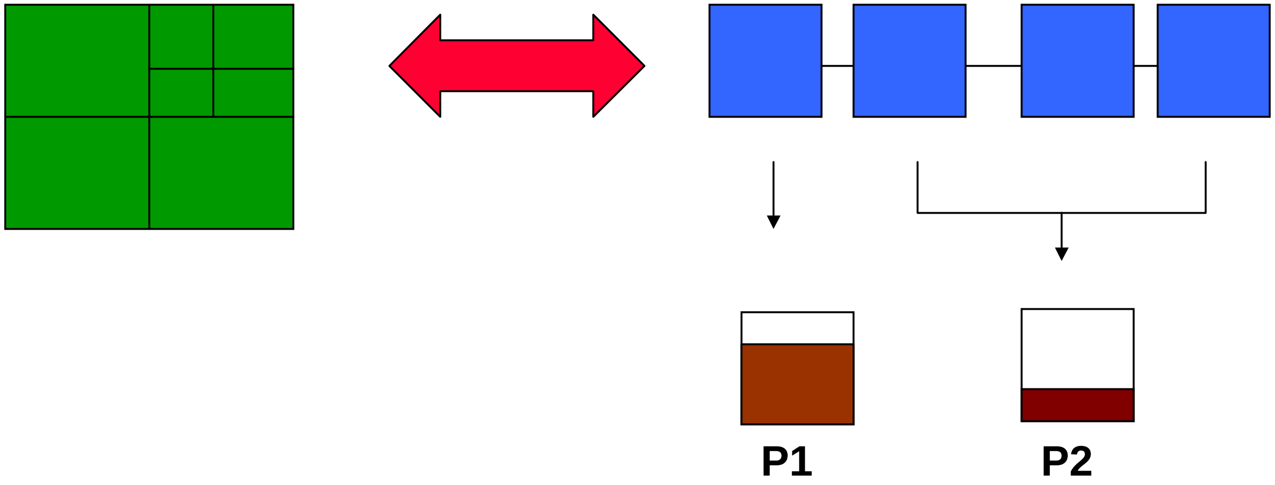
$$C(k) = w_p CPU(k) + w_m MEM(k) + w_l LINK(k)$$

$$w_p + w_m + w_l = 1$$

Weights are application dependent and reflect the applications computational, memory and communication requirements.

# The System Sensitive Partitioner

- In GrACE component grids in the adaptive grid hierarchy are maintained as a list of grid patches
  - – *It is a region in the computational domain*
  - – *Every time application regrids, the bounding box list is updated and passed to the partitioner for load balancing*



Jyoti Batheja, TASSL, Rutgers University

# System Sensitive Partitioning

- L is total work associated with the bounding box list that can be assigned to processors
- $L_k$ is work that can be assigned to kth processor. Computed as $L_k = C_k * L$ where
  - *$C_k$ is relative capacity of processor k*
- If work of bounding box > $L_k$

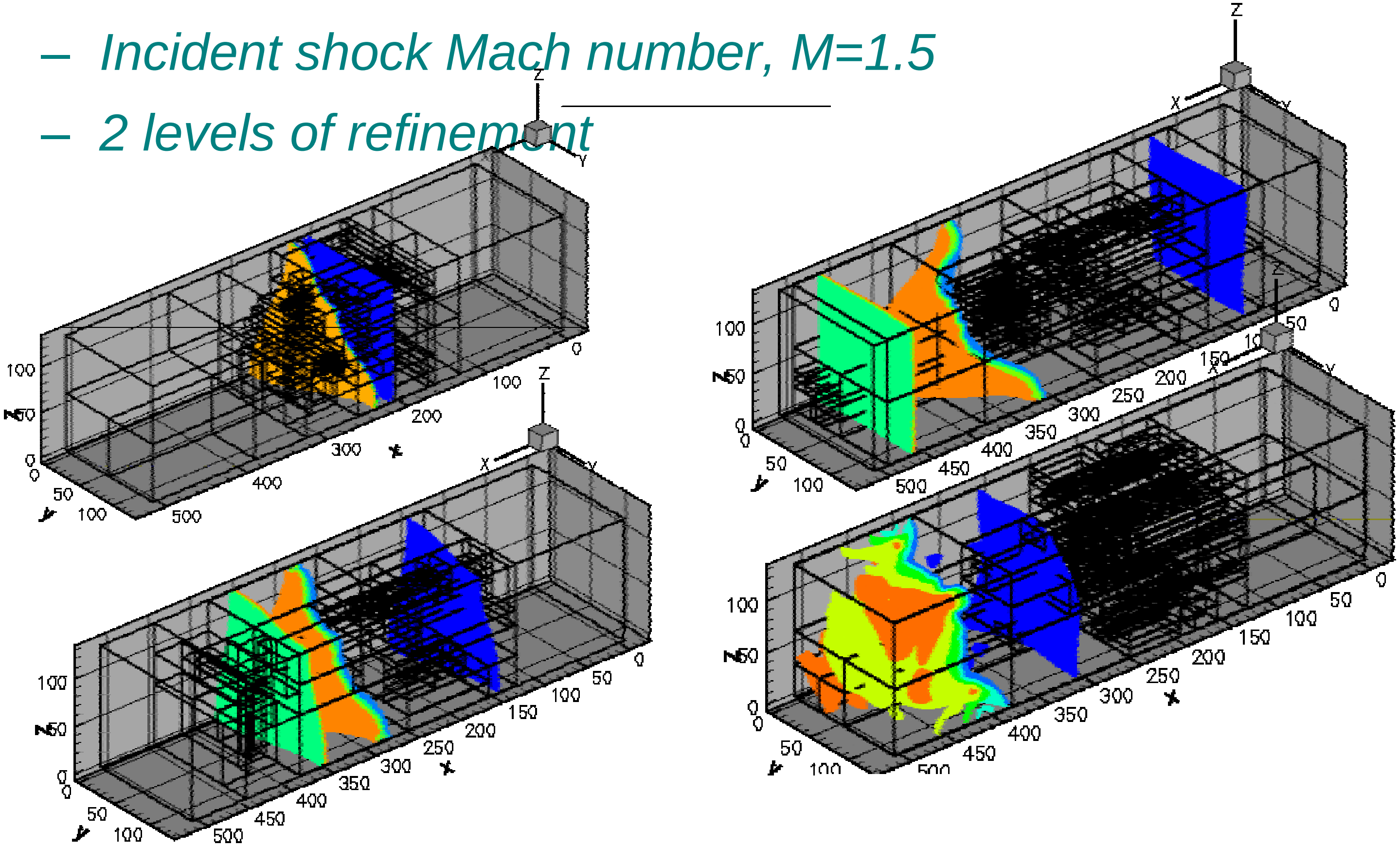
  Break the box under following constraints:
    - *Minimum box size*
    - *Aspect Ratio*

# System Sensitive Partitioning: Experimentation Setup

- Application - RM3D Compressible Turbulence Application
  - *Euler equations of motion for compressible fluid in three dimensions (Ravi Samtaney et al., Caltech)*
  - *128x32x32 base (coarse) grid*
  - *2 levels of factor 2 refinement*
  - *Refinement every 4 iterations*
- System
  - *Beowulf cluster at University of Texas at Austin (32 Nodes)*
- Synthetic Load Generation
  - *CPU and Memory usage are varied to change relative capacities of processors.*

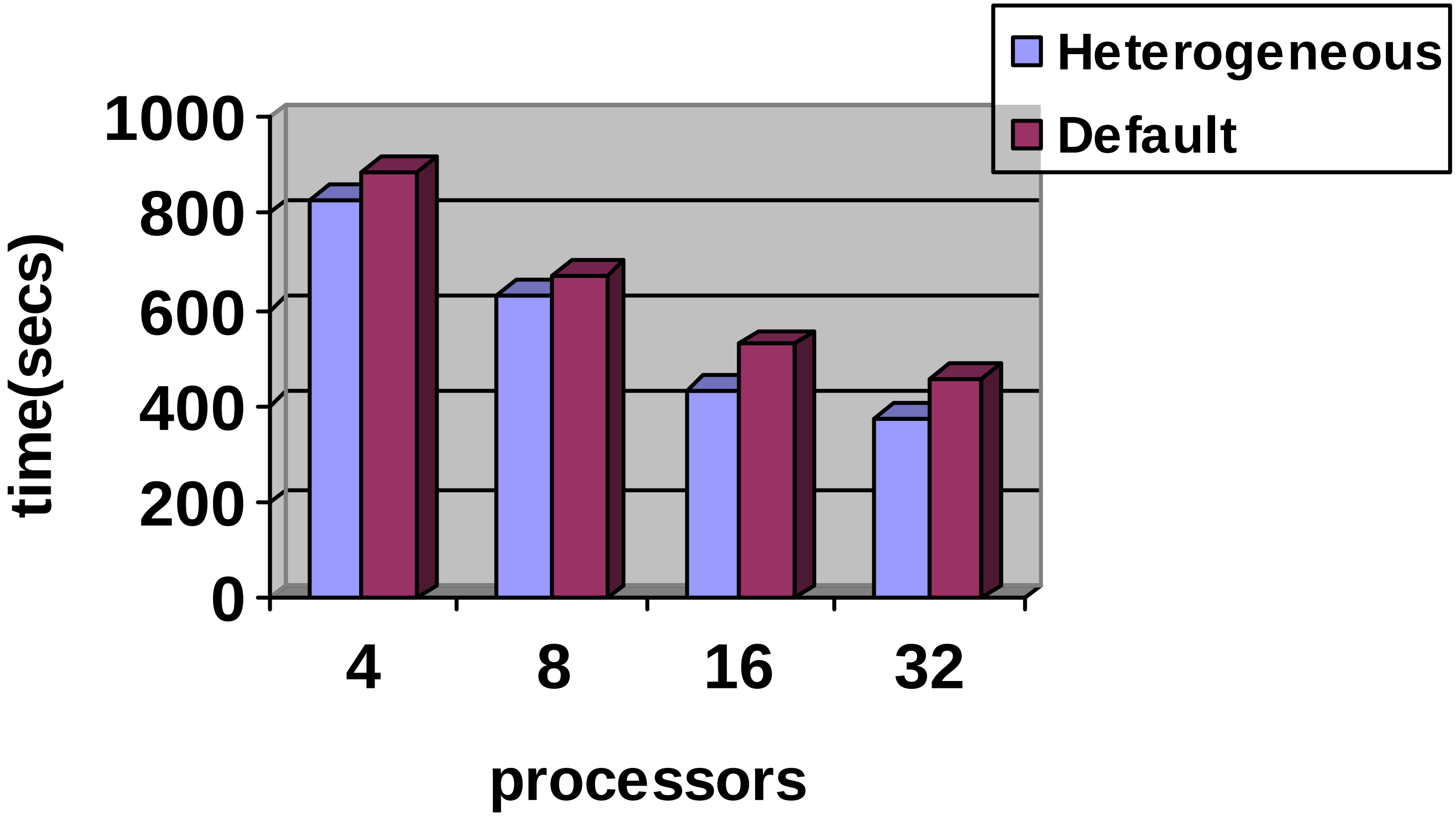
# RM3d: GrACE 3D AMR Example

- Richtmyer-Meshkov Instability
  - *Air-SF6 interface with single harmonic perturbation.*
  - *Incident shock Mach number, M=1.5*
  - *2 levels of refinement*



Jyoti Batheja, TASSL, Rutgers University

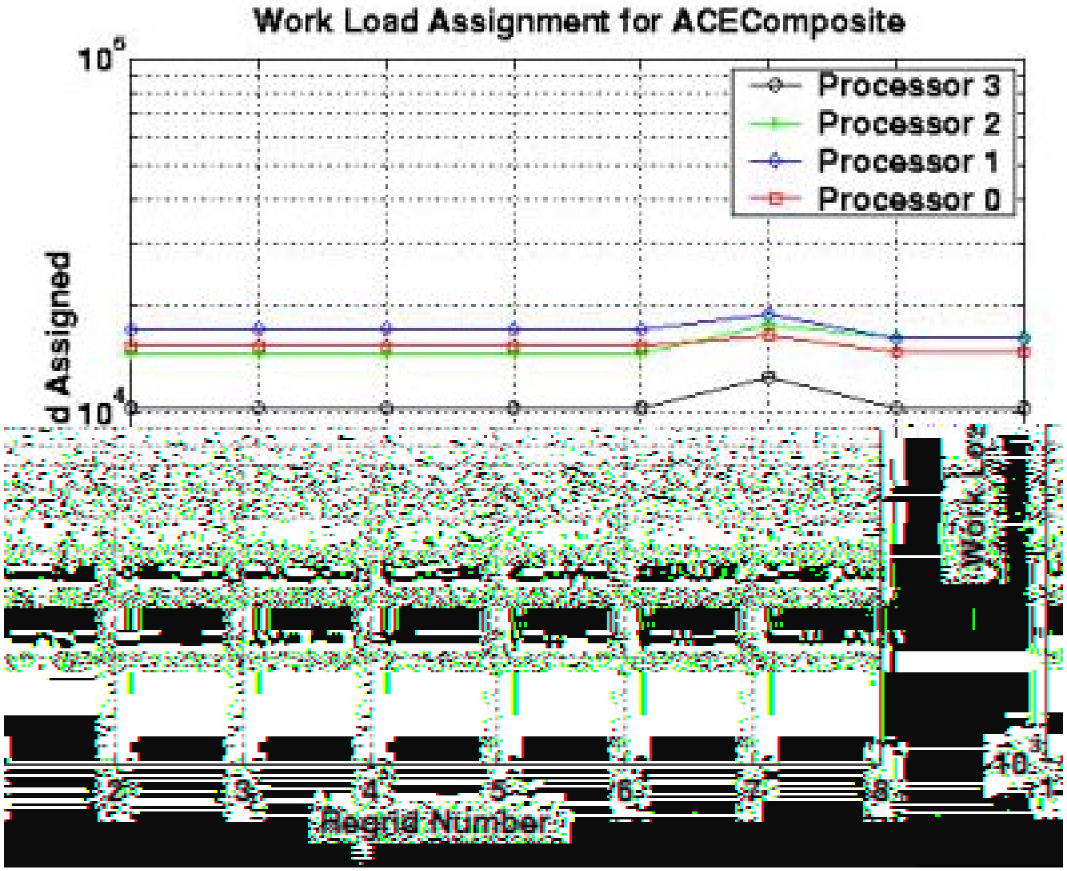
# System Sensitive Partitioning: Execution Time



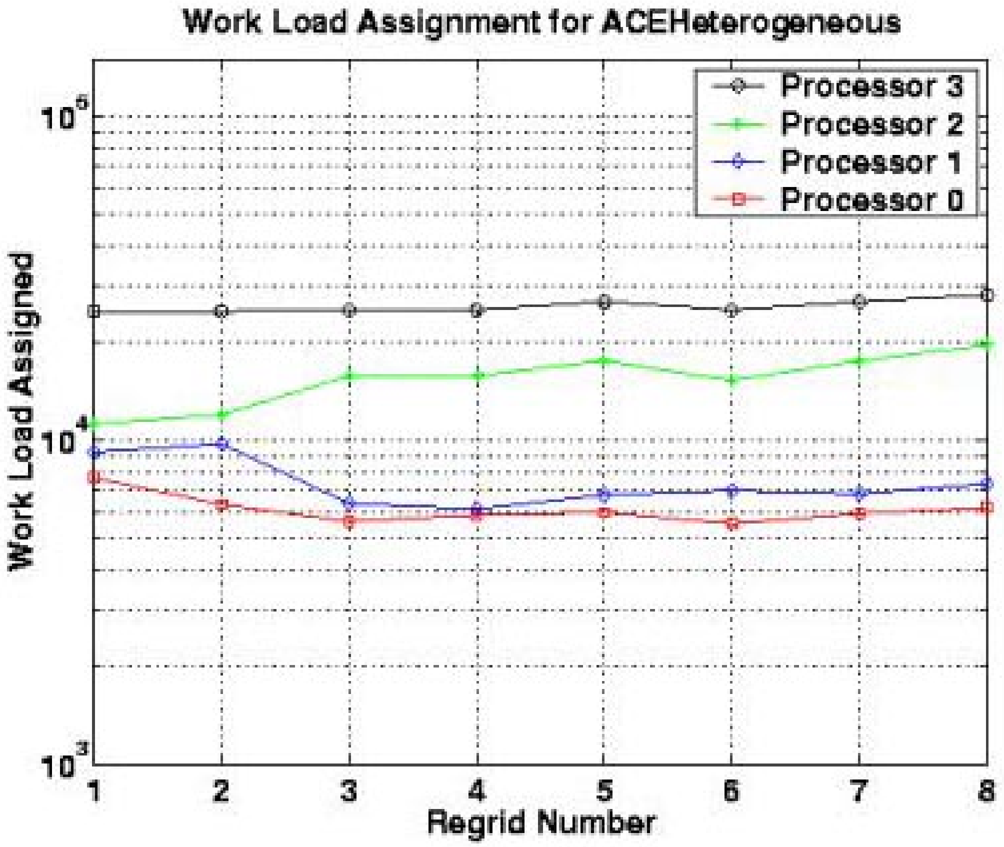
# System Sensitive Load Distribution

- Consider cluster with 4 nodes and synthetic load generator on 2 of the nodes.
- Relative capacity calculated as 16%, 19%, 31%, and 34%.
- Nodes are assigned work load proportional to .16L, .19L, .30L and .34L. Here L is the total work

# Work Assignment Using Default Partitioner (ACEComposite)



# System Sensitive Work Assignment



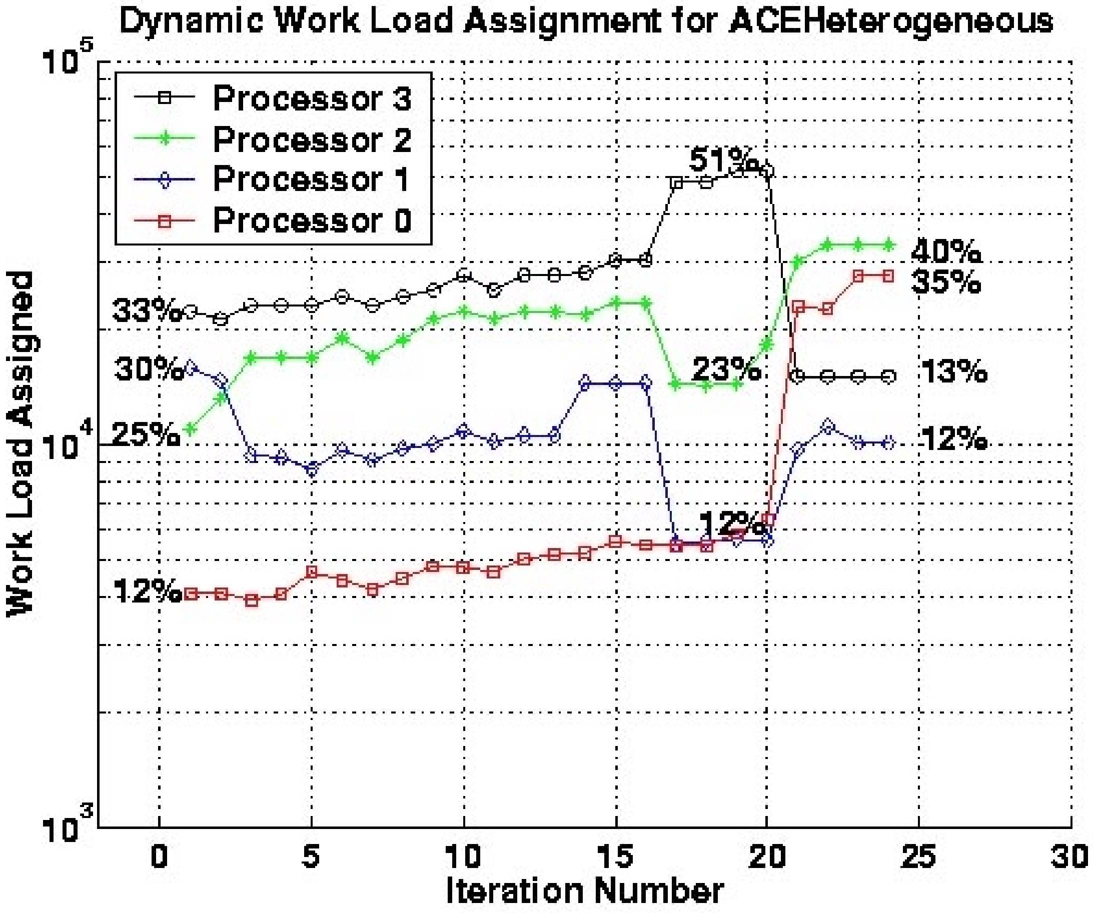
The relative capacities of processors are 16%, 19%, 30%, 34% and load is distributed accordingly.

Jyoti Batheja, TASSL, Rutgers University

# Adaptivity to Load Dynamics

- This experiment evaluates
  - *Ability of the system sensitive partitioner to adapt to the load dynamics*
  - *Overheads involved in sensing the current state*

# System Sensitive Partitioning: Dynamic Load Assignment



Jyoti Batheja, TASSL, Rutgers University

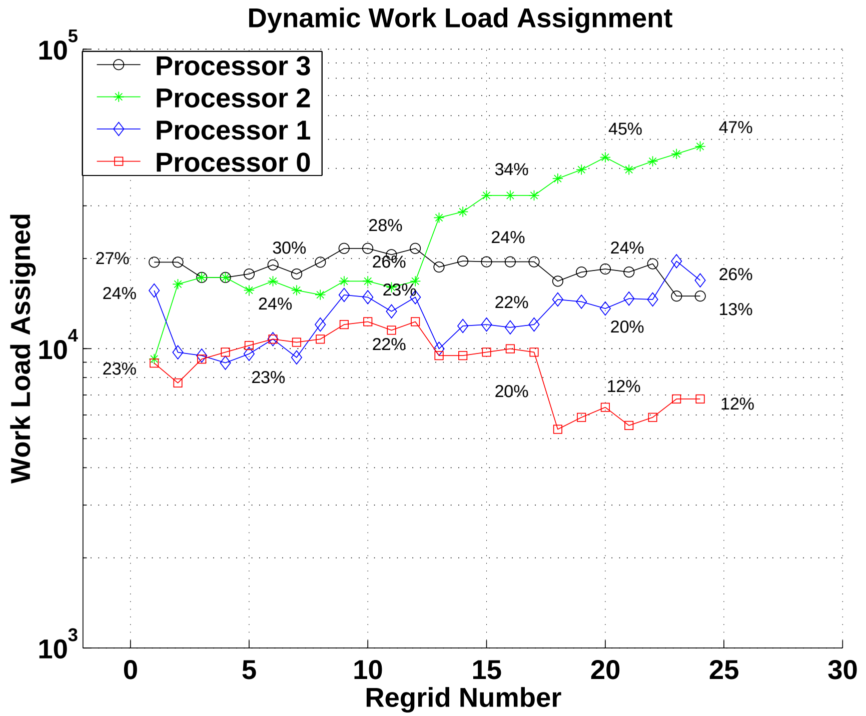
# Execution Times

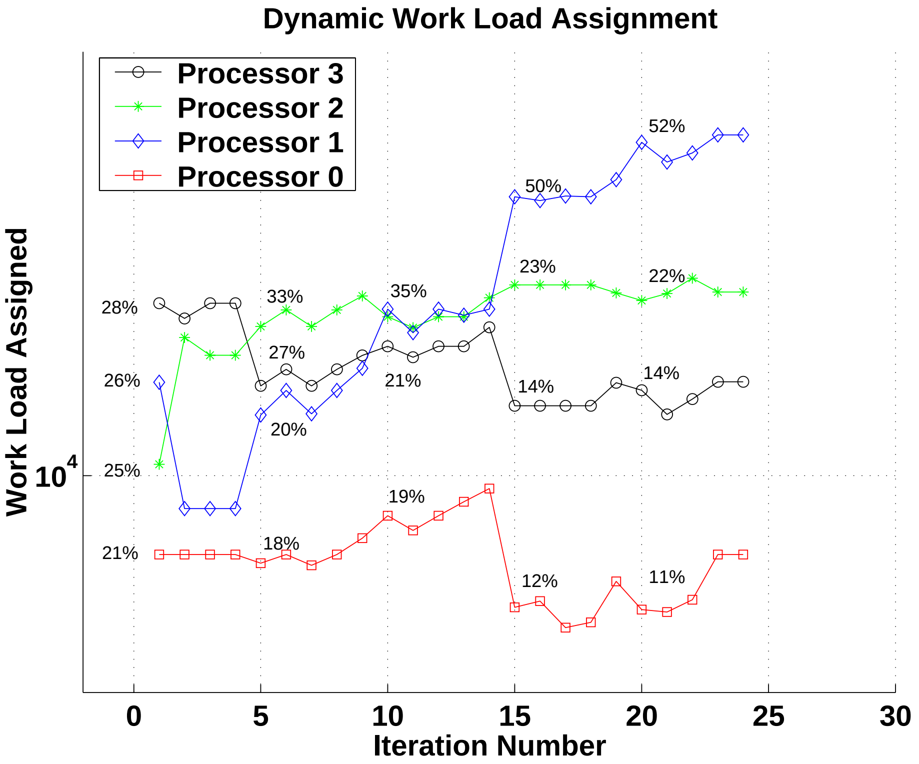
| Number of Processors | With Dynamic Sensing (every 20 iterations) (secs) | Static Sensing/ Sense only once (secs) |
|---|---|---|
| 2 | 423.7 | 805.5 |
| 4 | 292 | 450 |
| 6 | 272 | 442 |
| 8 | 225 | 430 |

# Overheads of sensing frequency

| Frequency of calculating capacities | Execution time (secs) |
|---|---|
| 10 iterations | 316 |
| 20 iterations | 277 |
| 30 iterations | 286 |
| 40 iterations | 293 |

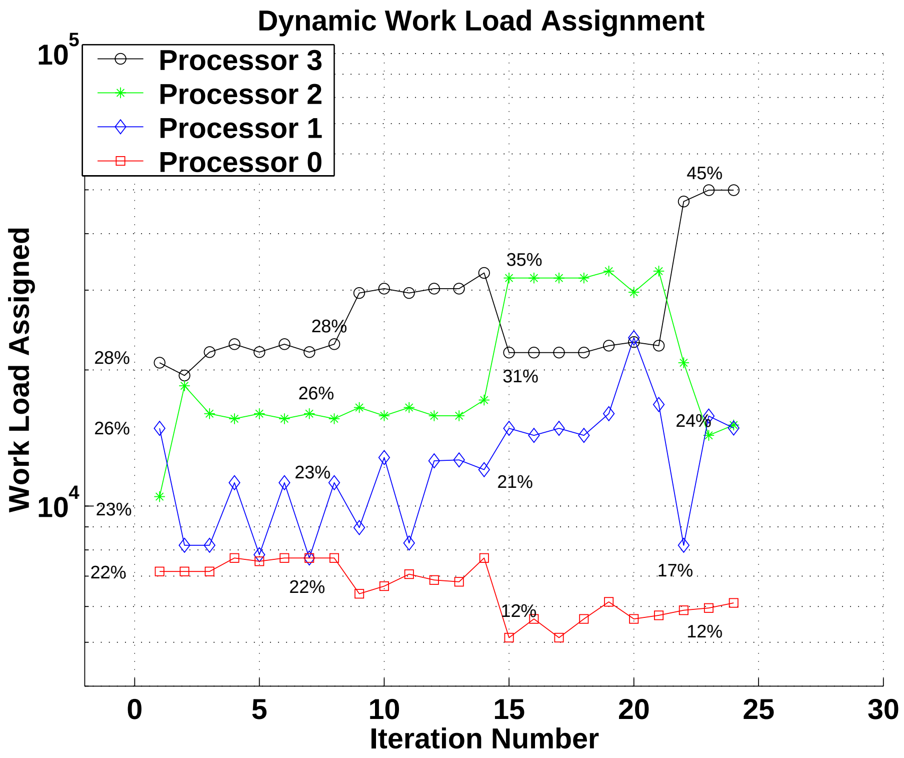
# System Sensitive Partitioning: Dynamic Load Assignment

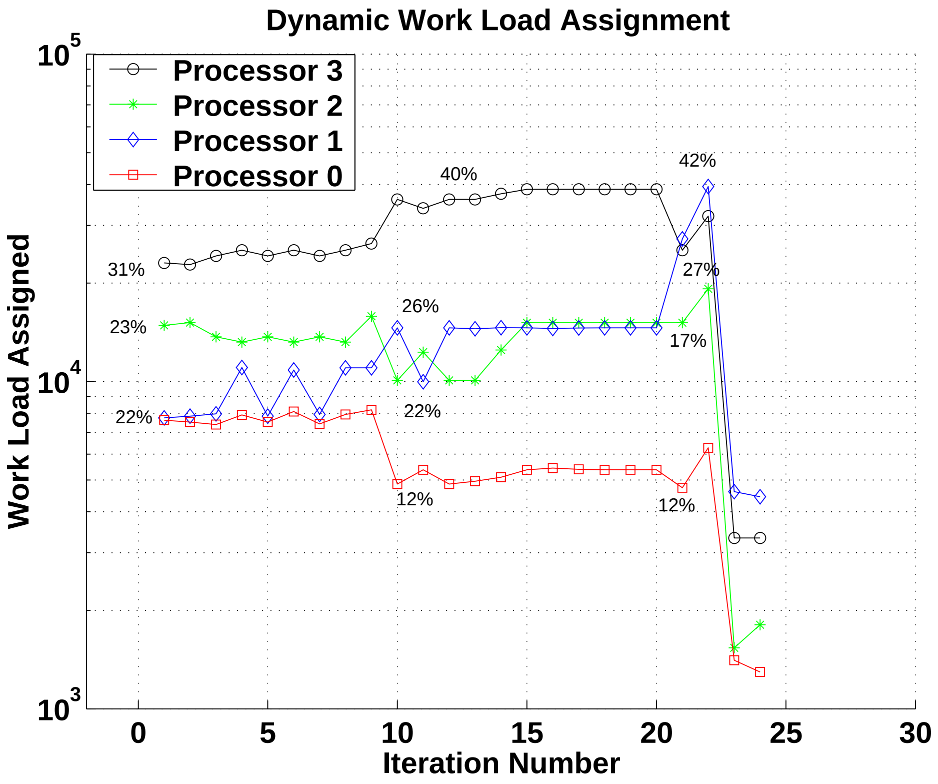




**Load sensing every 10 and 20 iterations**

# System Sensitive Partitioning: Dynamic Load Assignment





**Load sensing every 30 and 40 iterations**

# Summary of Results

- System-sensitive partitioner

| | |
|---|---|
| Execution time | reduced by 18% |
| Load Imbalance | reduced by 45% |
| Dynamic runtime sensing | reduced execution time by 45% |

  - *Distributed work load according to relative capacities of the computing nodes*
  - *Through dynamic-sensing it adapted to load dynamics of cluster*

# Conclusions & Future Work

- A System-Sensitive partitioner for AMR applications
  - *Adapt to system state in a heterogeneous networked environment*
  - *Uses NWS to query current system state*
  - *Use relative system capacity to drive load-balancing*
- System sensitive partitioning improves performance
  - *Speedup*
  - *Reduced load imbalance*
- Dynamic adaptation to changes in network/system environment
- Current Work
  - *Balancing monitoring overheads -frequency of sensing*
  - *Use NWS predictive capabilities*

# Email Contacts

- Shweta Sinha
  - *shwetas@caip.rutgers.edu*
- Manish Parashar
  - *parashar@caip.rutgers.edu*