Load Balancing and Unbalancing for Power and Performance in Clusters

Eduardo Pinheiro, Ricardo Bianchini,
Enrique V. Carrera, and Taliver Heath

Department of Computer Science
Rutgers University

Introduction

- Previous work focused on battery-operated devices
- We focus on power and energy research for clusters
- Why?
  - Large clusters consume lots of power and energy
  - Power consumption affects cost of cooling, backup cooling, and backup power generation
  - Energy consumption affects electricity costs
  - Most power-generation technologies are harmful to environment

Introduction

- Some interesting facts that motivate our research:
  - Each Google site: 40 racks with 80 PCs each, for a power consumption of 180 KWatts and an energy consumption of 130 MWh/month
  - California and New York went through energy crisis
  - Power requirement of 46 data centers planned for NYC is 500 MWatts (500,000 households)
  - Data center projects cancelled or delayed due to inability of states to meet these requirements

Our Approach

- Observation: Cluster resources are widely replicated
- Idea: Schedule demand so that resources become idle and, thus, can be moved to a lower power mode

- Two main facts about our cluster nodes:
  - Small difference (24 Watts) in power between idle and fully utilized
  - Power consumed high, even when idle (70 Watts)

- Power supply maintains charged capacity to respond rapidly to variations in demand, even when idle

Our Approach

- Approach: Turn entire nodes on and off
- Technique: Load Concentration = concentrate load on fewer nodes and turn others off
- Algorithm: concentration and balancing decisions, according to load and expected power, performance
- Implementations: clustered WWW server and operating system for clustered cycle servers

Outline

- Introduction
- Our Approach
- Cluster Configuration and Load Distribution
- Experimental Results
- Related Work
- Conclusions and Future Work

Cluster Configuration and Load Distribution

- Key principle: adjust #nodes and load distribution to conserve power w/o performance degradation

- Need to predict power and performance
- Power: eliminating nodes is always good
- Throughput: degradation is excess demand for a configuration. Ex: demand for disks = 220% for 2 nodes -> degradation = 10%
- Run time: optimistically assume degradation is excess demand for a configuration

Cluster Configuration and Load Distribution

```
Periodically do
    if node removal is acceptable
        choose victim nodes with low demand
        if necessary, determine nodes to receive load and
            ask victims to migrate their loads out
        ask victims to turn themselves off
    else
        if addition is necessary
            turn on new nodes
            if necessary, determine load for new nodes and
                ask nodes to share their load with new nodes
```

Implementations

- Master node runs the algorithm; nodes inform master about their demands
- Only one node added/removed at a time
- Power-aware WWW server based on PRESS [PPoPP01] and implemented at user-level
- Power-aware OS based on Nomad [IWCC99] and implemented at OS-level
- Original systems implement load balancing

Power-Aware WWW Server

- No throughput degradation is acceptable
- For removal, master determines max demand at each node and selects node with lowest max demand as victim
- Addition needed if demand for any resource at any node is higher than threshold
- Load is naturally redistributed

Power-Aware OS

- Run time degradation > 0 can be selected
- For removal, master selects pair of nodes with low resource demands
- Addition needed if degradation is not acceptable and more than one application is responsible for excessive demand
- Load is balanced by the OS

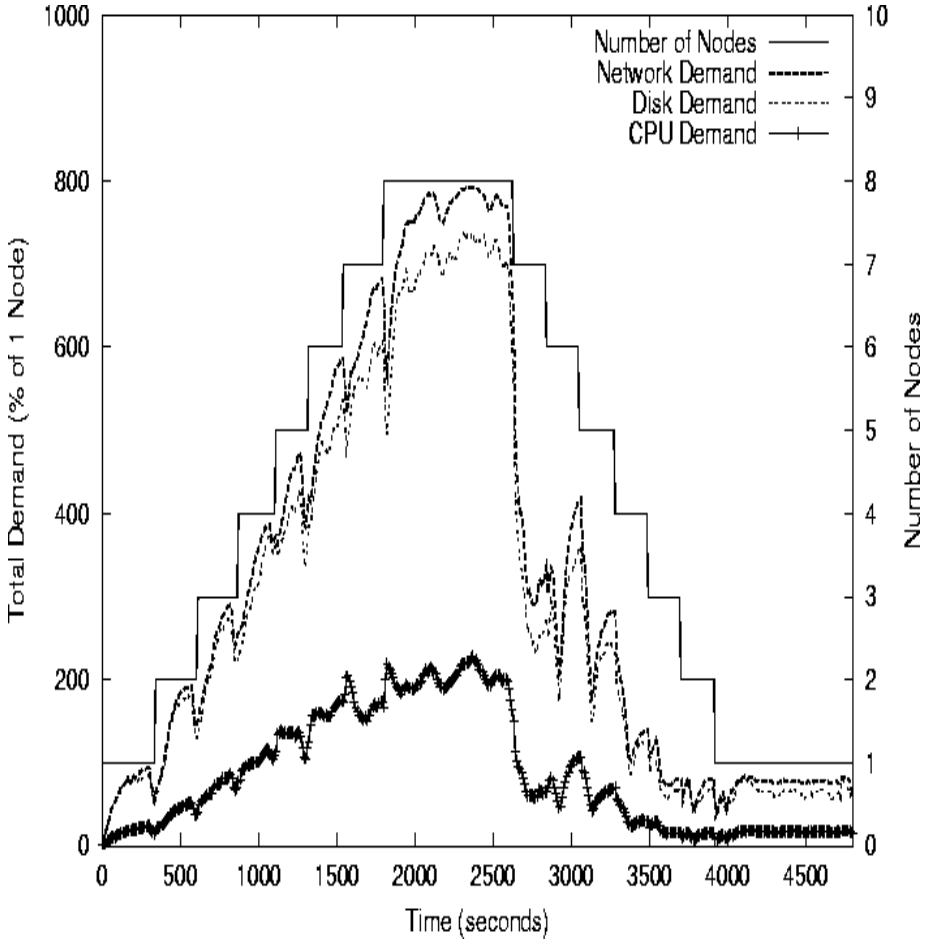
Experimental Setup

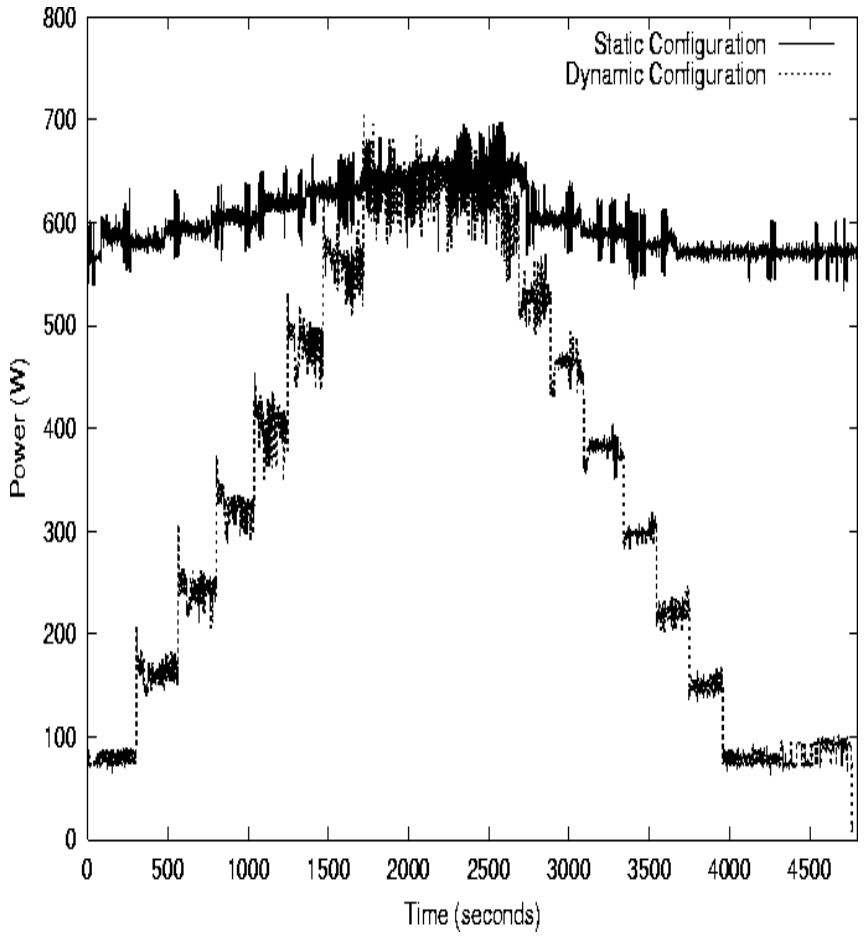
- Cluster of 8 single-processor PCs
- Machines connected to a smart power strip
- Shutting down takes 45 seconds; booting up takes 100 seconds
- Power consumption monitored by a multimeter connected to the power strip
- Log sent to another node for storage

Experimental Setup

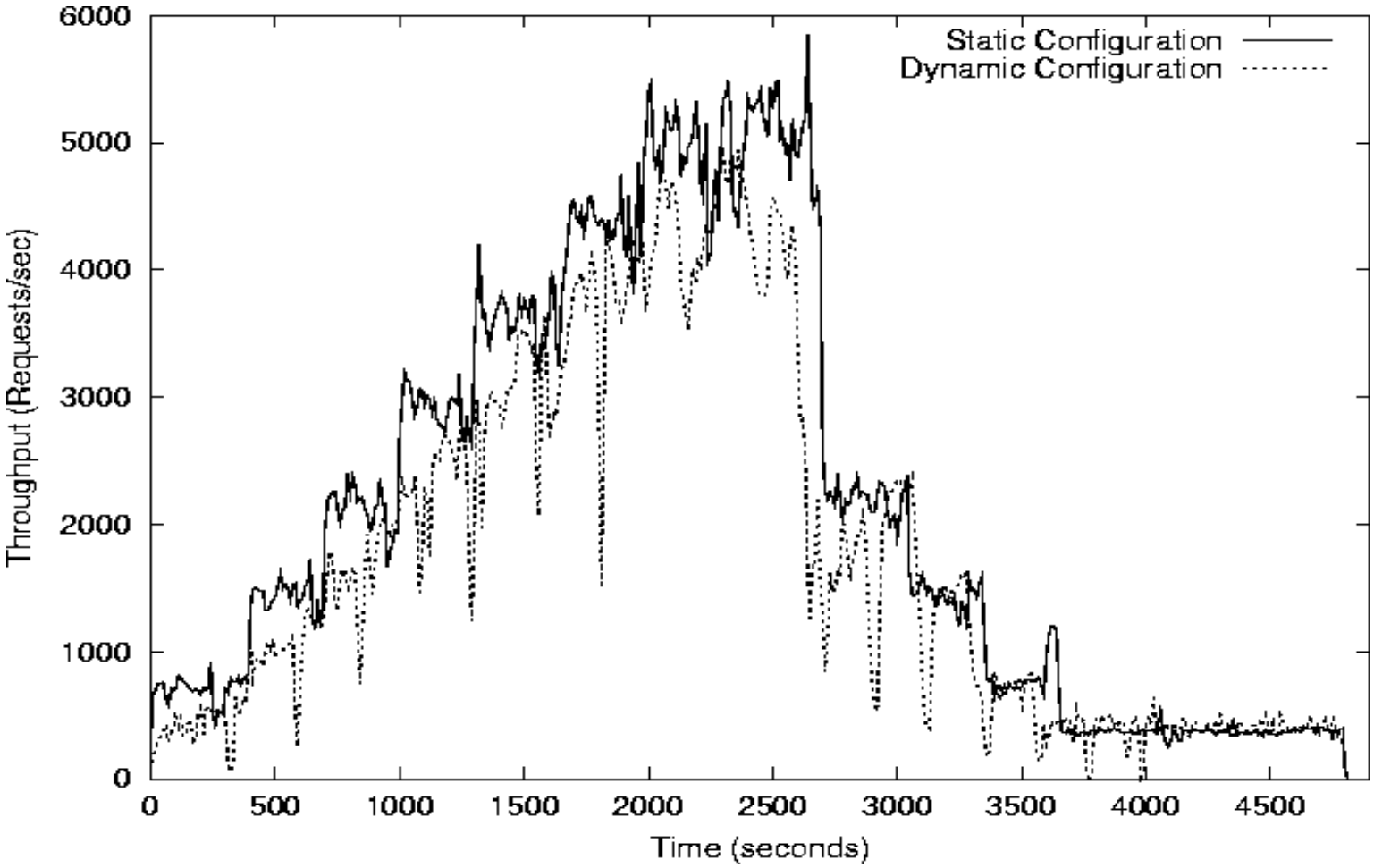
- Power-aware WWW server experiments
  - 12 other PCs generate load to cluster
  - Real trace but accelerated and bell-shaped
  - Avoid throughput degradation
- Power-aware OS experiments
  - Synthetic workload (SPEC2000, MPEG, I/O)
  - Bell-shaped with quick and significant changes
  - Time degradation < 20% is acceptable
- Comparisons against static counterparts

Power-Aware WWW Server

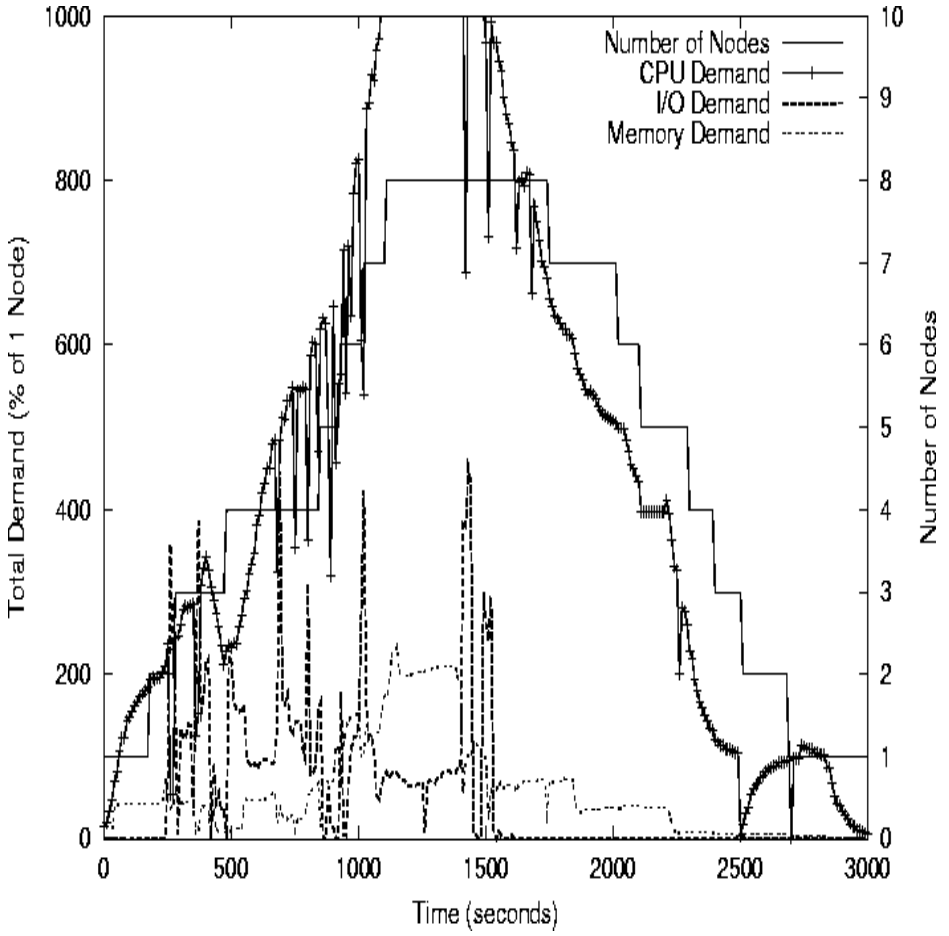


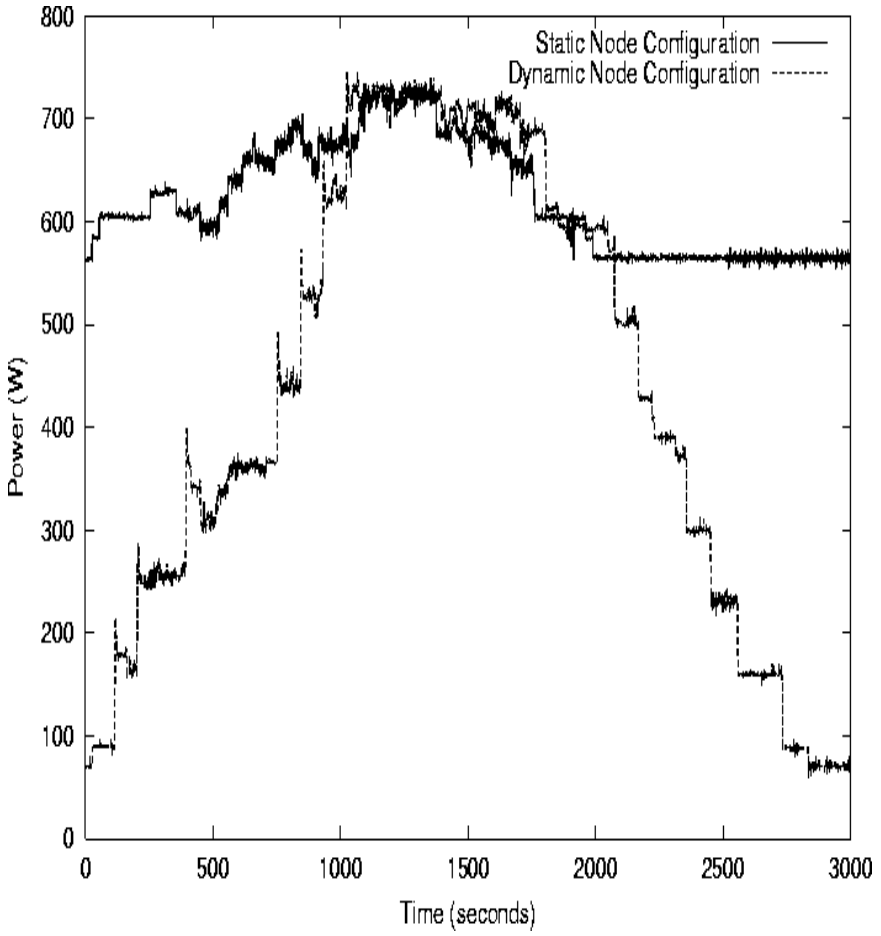


Power-Aware WWW Server



Power-Aware OS





Summary of Results

- Power-aware WWW server
  - Power consumption reduced by up to 86%
  - Energy consumption reduced by 43%
- Power-aware OS
  - Power consumption reduced by up to 86%
  - Energy consumption reduced by 32%

Related Work

- Orthogonal to advances in single-processor or battery-operated devices
- [SOSP'01] studies resource allocation policy for clustered WWW server. Similar to our algorithm
- Load concentration inspired by load balancing. Used only as remedial, e.g. [Mosix '98], or management technique
- Cluster reconfiguration, e.g. [Oceano '01]
- Closest in spirit to load concentration: remote execution [Rudenko '98, Kremer '00]

Conclusions and Future Work

- Simple technique + algorithm for power & performance
- Two interesting implementations
- Experiments show power and energy gains

- Future work:
  - Transition multiple devices between multiple modes
  - Consider migration and transition costs explicitly
  - Consider energy tradeoffs explicitly
  - Detailed model of power and energy consumption
  - Real, non-reshaped workloads and parameters

More Information

http://www.darklab.rutgers.edu/