On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters

Karthick Rajamani and Charles Lefurgy

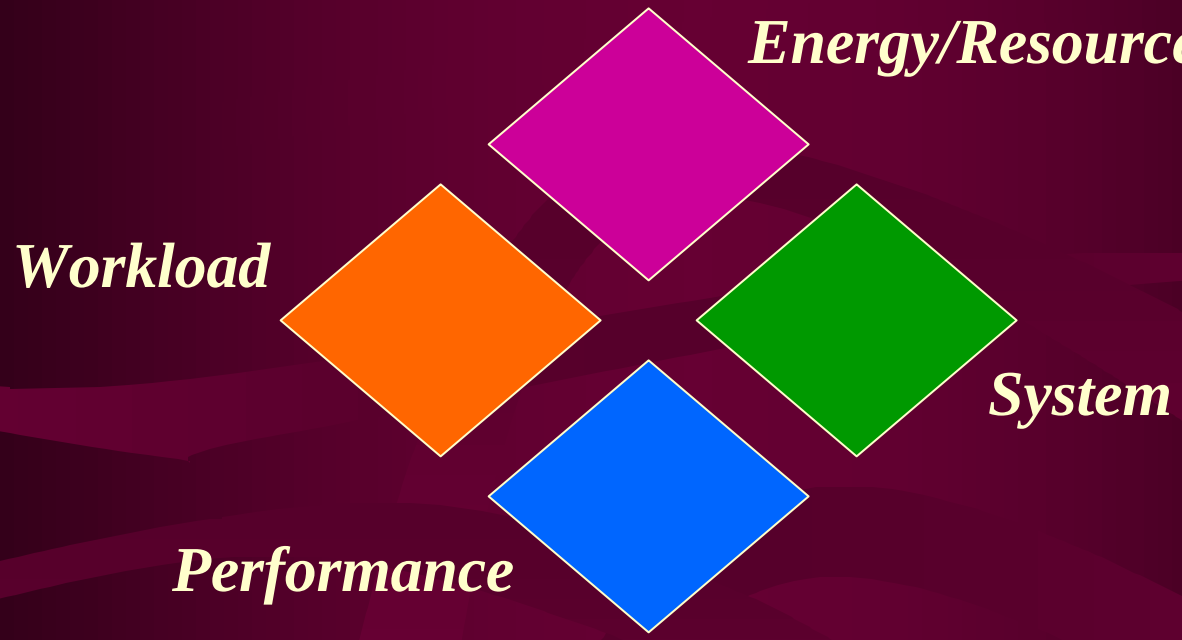
{karthick,lefurgy}@us.ibm.com

IBM Austin Research Lab

Power-Aware Request Distribution (PARD)

- Environment
  - Cluster of Servers
  - Request distribution infrastructure balances load among servers.
- PARD Idea
  - Save energy, matching resource usage to load while providing required performance.
    - Monitor load in terms of number of active connections.
    - Given upper bound on connections per server (for required performance), power-off excess servers.

Problem Dimensions



Energy Savings vs Performance trade-off is the focus of energy-saving studies, incl. PARD.

System-workload context impacts the energy-performance trade-offs and the energy-saving strategies that can be explored.

Contributions

- Identified key system and workload factors that impact PARD strategies.
- Exploited knowledge of system-workload context to derive better energy consumption estimates – quantified impact and verified effectiveness of models.
- Developed novel approach for exploiting conventional benchmarks for resource-/energy-saving studies.

System Factors

- **Cluster unit**
  - Capacity: number of connections that a cluster unit/server can service with required performance.
- **Startup delay**
  - Time to bring-up a powered-down server into the service.
- **Shutdown delay**
  - Delay in powering-down a server after removing it form the active server pool.
- **Ability to migrate service/connections**
  - Affects flexibility in reducing number of active servers.

Workload Factors

- **Workload unit**
  - Size of schedulable work, unit for *capacity*.
- **Load profile – load versus time**
  - peak and shape.
- **Relation between load and rate of change in load**
  - The same rate of change in load could have different impact depending on current load.

Energy Consumption Model

- Powered-on servers have fixed energy cost, independent of load.
- Powered-off servers have zero energy cost.

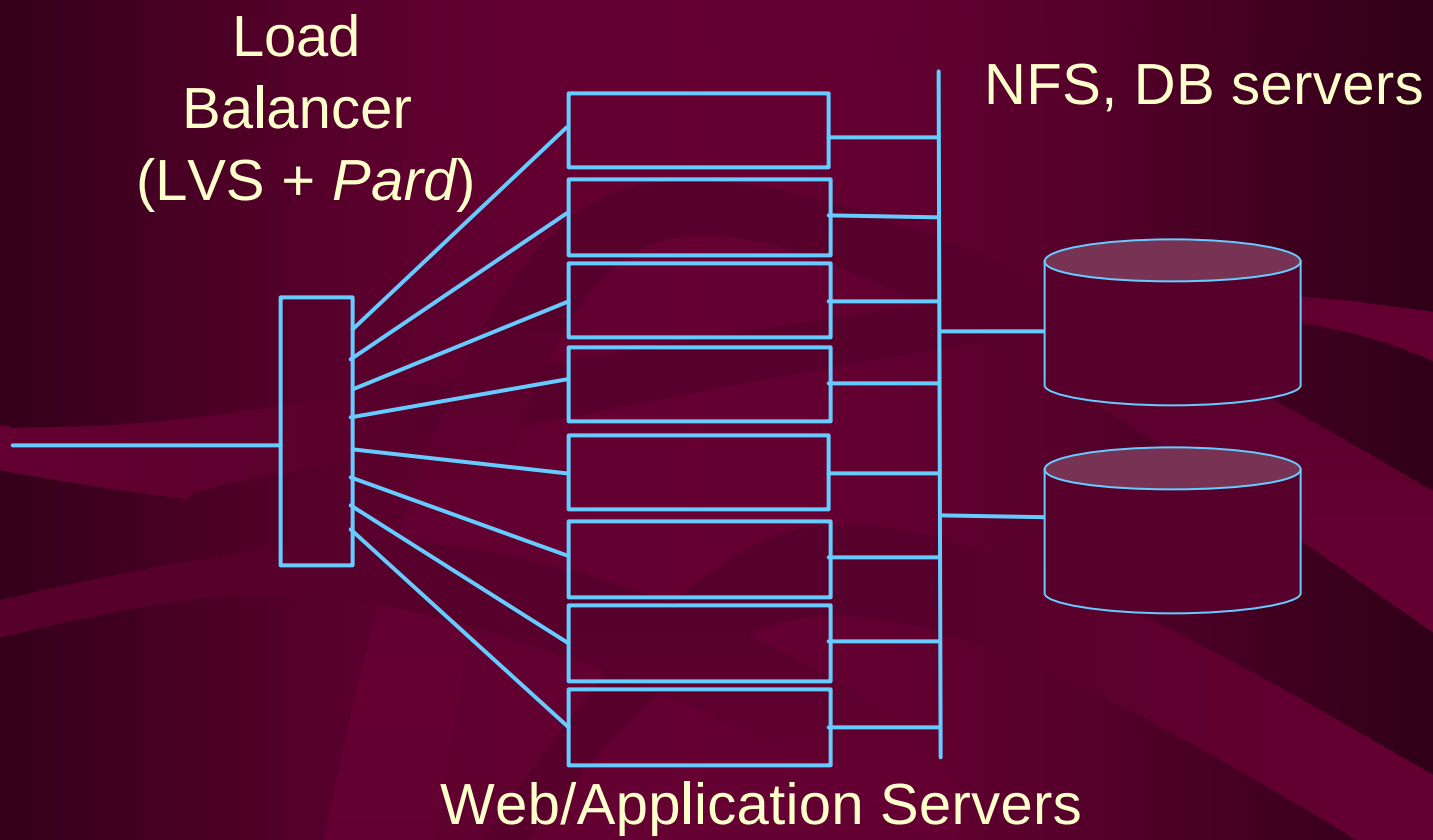
- Assumption valid for our platform, could require extensions for others.

PARD and SW context

$$L + SD < NC$$

- Workload
  - Load L (number of connections),
  - Slope of load curve, S
- System
  - Startup delay D,
  - Capacity of server, C .
- Energy
  - N - number of active machines.

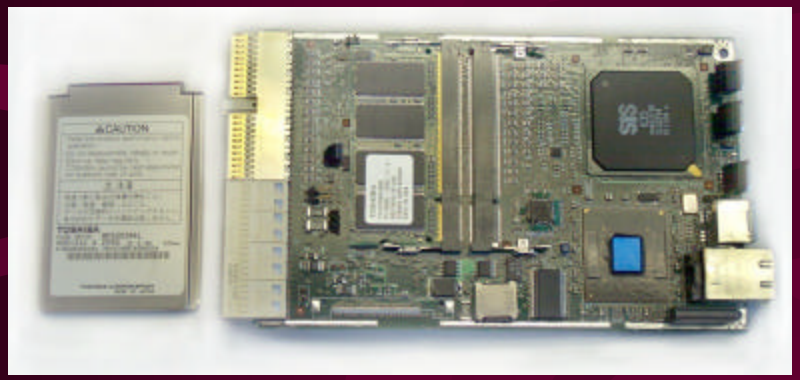
Experimental Setup

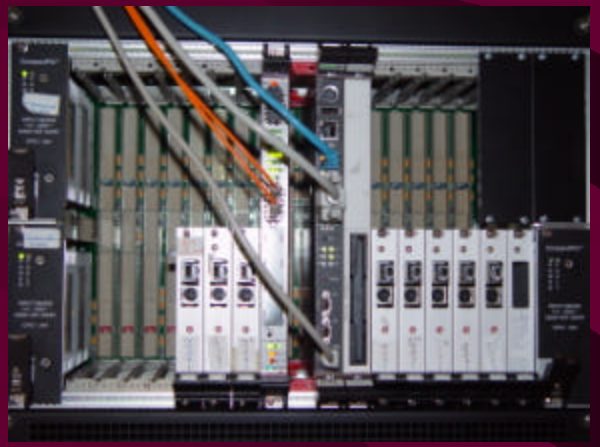


PARD employed for power-managing application servers.

Evaluation Platform

- Super-Dense blade Servers (SDS) – Application servers





- Separate LVS server, Image and Database servers.
- Wall-power energy measurement for SDS.

Workload for Energy Studies

Problem

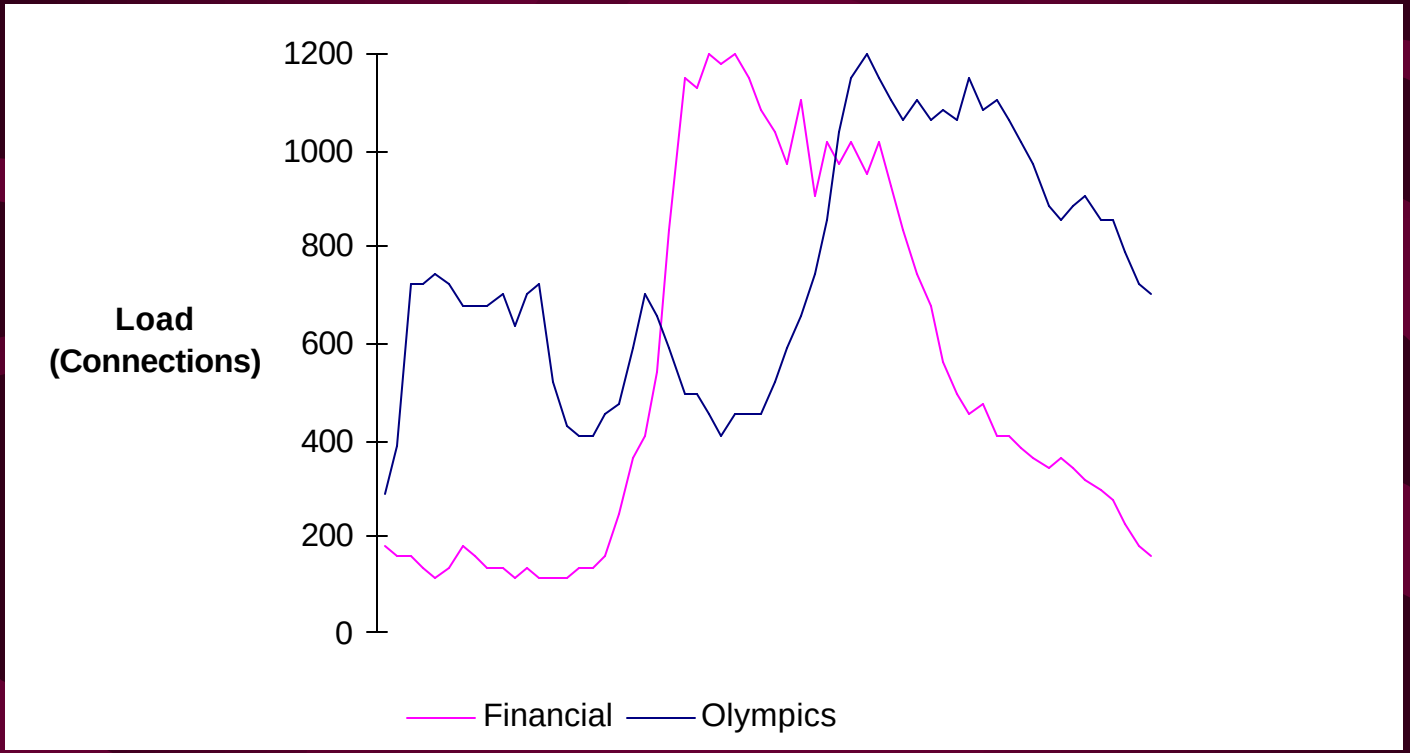
- Workloads must have variation in load for powering-down/lowering performance-energy levels.
- Benchmarks lack load variation, focus on peak performance.

Solution: combine application characteristics captured by benchmarks with real variations in load.

- TPC-W: e-commerce benchmark, client emulators send dynamic page requests to modeled e-commerce site.
- Modify clients at time t, according to desired load profile – from logs of real web sites.
- Scale load profile to *capacity* of the system.

Load Profiles

One day's web logs of a Financial web site, Winter Olympics '98.



PARD based on Simple Threshold (ST)

- React to observed load.
- Threshold T – new server powered-on when number of active connections/server exceeds T.

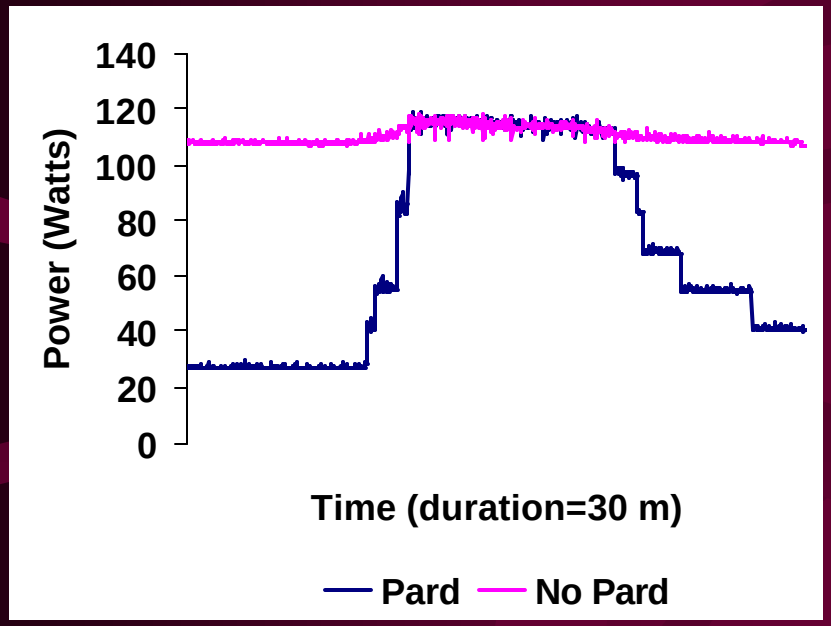
$$N_t = L_t/T \qquad (1)$$

$$N_t >= \max(L_{t->t+D})/C \qquad (2)$$

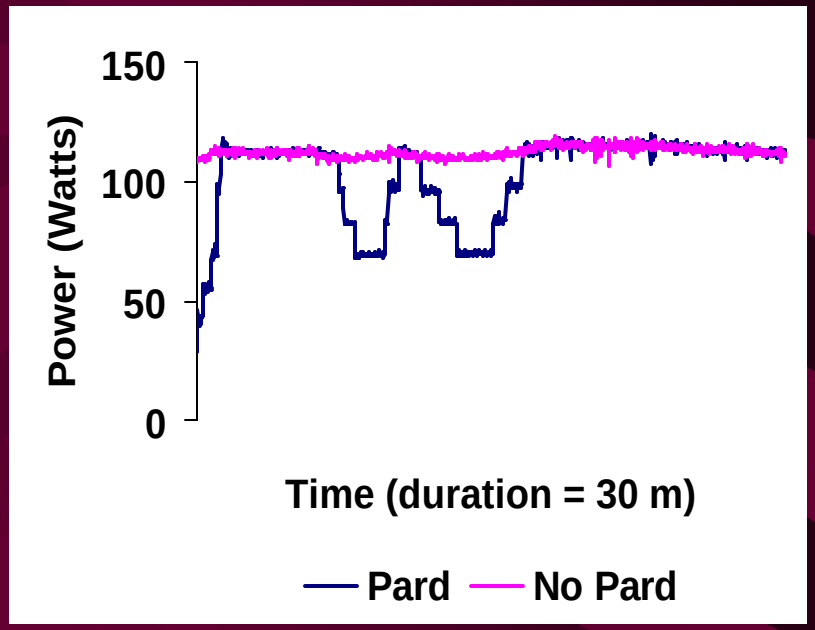
Using (2), provides T for maximum energy savings.

Simple Threshold Results

Financial



Olympics



PARD (ST) saves 36.7%

PARD (ST) saves 7.1%

Model vs Measured

Load	Traditional estimate	SW estimate	Measured Savings
Financial	51%	40%	36.7%
Olympics	32%	10%	7.1%

Traditional estimate is the savings estimated as L_t/C.

New Schemes from SW insight

- Spare Servers – spare active servers accommodate sharper increases in load, leading to higher thresholds and potentially higher energy savings.

- History-based – Use different extent of knowledge of workload to appropriately start and stop servers. e.g. Perfect knowledge: start a server just *startup-delay* seconds before it will be required.

Results for New Schemes

- Spare Servers – shown for 3 spares

Load	Traditional estimate	SW estimate	Measured Savings
Financial	51%	27.3%	25.1%
Olympics	32%	19.8%	17.7%

- Perfect Knowledge

Load	Traditional estimate	SW estimate	Measured Savings
Financial	51%	49.2%	45.6%
Olympics	32%	30.4%	26.1%

ST savings are 36.7% and 7.1% for Financial and Olympics, respectively.

Conclusions

Established importance of system-workload context

- Better estimates, evaluation.
- Better strategies.

New Schemes

Spare Servers

$$Max(L_{t,t+D}) / C <= (S + L_t / T_S = N_t)$$

History-based Servers

$$N_t = (L_t + S_{max} * D) / C$$

$$N_t = (L_t + S_N * D) / C$$

$$N_t = (L_t + S_t * D) / C = Max(L_{t->t+d}) / C$$

SDS Blade Power Budget

Processor	6.402
SODIMM 256MB	1.000
Voltage Regulator	0.005
North/South/Ethernet	1.980
Ethernet PHY	0.660
LPC Flash Memory	0.033
EEPROM	0.007
PCI to PCI Bridge	0.173
Supervisory Processor	0.330
Ethernet Controller	0.743
Voltage Monitor - I2C	0.008
Clock Generator	0.693
Disk	1.485
90% efficient power supply	1.352
Total	14.871

SDS vs Conventional Racks

	SDS Cluster	Conventional
CPUs	360	42
CPUs/U	8.57	1
Processor speed	180 GHz (x-86) (500 MHz each)	101 GHz (x-86) (2.4 GHZ each)
Main memory	184 GB	168 GB
Ethernet	71.4 Gb/s	84 Gb/s
L2 cache	92 MB	42 MB
I/O buses	360	42

Software on SDS

- Linux Diskless Server Architecture
  - Single system image for all blades
  - Boot from management blade disk
  - Blades are diskless and boot in 20 seconds
- Ethernet block device
  - High performance swap
  - Serving web content
- Blade management across I2C bus
  - H8 microcontroller on blades acts as power switch
- Console over Ethernet
- Power-Aware Request Distribution
  - Quick boot time reduces “idle” power

States of a PARD-managed Server