SWINBURNE UNIVERSITY OF TECHNOLOGY

Energy efficiency in multi-core supercomputing

Tuan V. Dinh

Supervisors: A/Prof. Lachlan Andrew, Dr. Philip Branch and Dr. Yoni Nazarathy

Motivations

Energy consumption in supercomputer

(1)

Electricity bills

Heat – thermal management

Investment – cooling systems, hardware, etc.

SWINBURNE UNIVERSITY OF TECHNOLOGY

(1) The Green Machine I – http://astronomy.swin.edu.au/supercomputing/

Motivations

- Trend: desired for more powerful computing,
  - climate change, astronomy
  - "exascale" supercomputing
- Graphics Processing Units to replace ordinary CPUs

(1) Green Machine

(2) GStar – Green II

SWINBURNE UNIVERSITY OF TECHNOLOGY

(1) ,(2) http://astronomy.swin.edu.au/supercomputing/

Prior work

Design and thermal management.
(Al-Fares et. al 2008, Sharma et. al, 2005; Patel et. al, 2002...)

SWINBURNE UNIVERSITY OF TECHNOLOGY

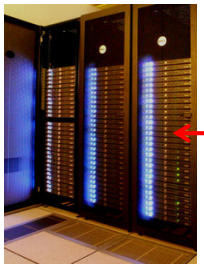
Prior work

Energy efficiency in single processor
(Yao 1995; Wierman et. al 2009;...)

SWINBURNE UNIVERSITY OF TECHNOLOGY

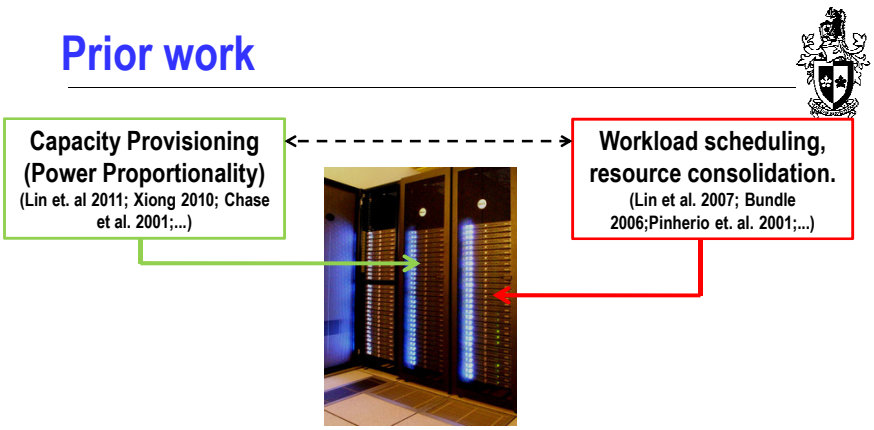
Prior work

Workload scheduling, resource consolidation.
(Lin et al. 2007; Bundle 2006;Pinherio et. al. 2001;...)



SWINBURNE UNIVERSITY OF TECHNOLOGY

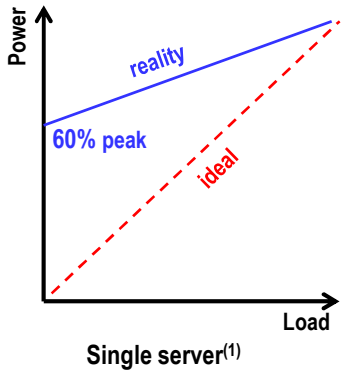
Prior work



Power proportionality

Capacity Provisioning (Power Proportionality)

- When idle, server still consumes at least 60% of peak power
- Save large amount of energy by powered them off
- Challenges:
  1. Short of supply in near future
  2. Increase hardware failure rate

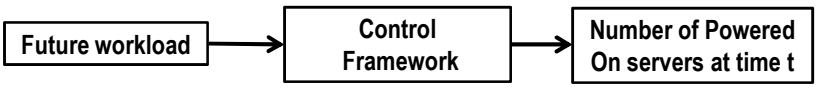


Single server(1)

(1) Bassoro, "The case for energy proportional", 2007.

Research question

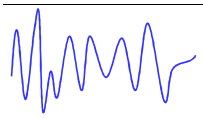
Capacity Provisioning
(Power Proportionality)



Outline

- Control framework
- Future load estimation
- Energy efficiency for single core
- Future work and challenges

Control framework



Future arrival load **A(t):**

Number of active server at time t

$$\min_{n(0),n(1),\ldots} \sum_{t=0} cos$$

Server "On" > A

Cost = Energy (E) + Switching cost (SC) + Performance degradation cost (PC)

∝ number of servers

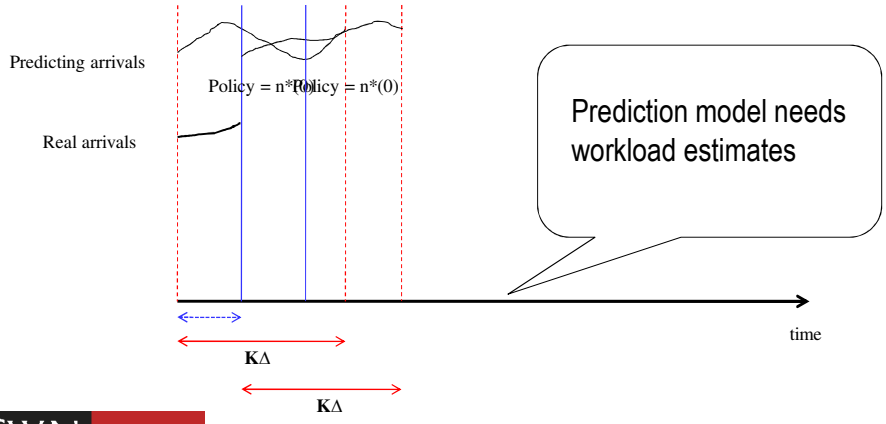
∝ number of switching servers

∝ times of short of supply

Challenges:
- Long future load estimation horizon
- Intractable for implementation

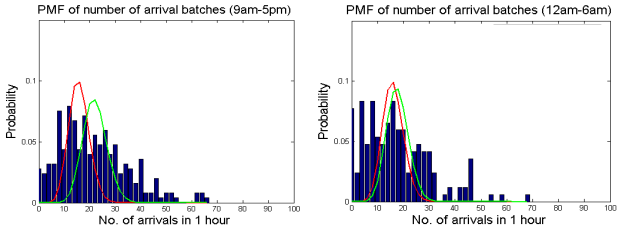
Model Predictive Control (MPC)

- MPC is a control method for stochastic control problem



Green Machine historical workload

- Our experiments show that Green Machine's workload is not Poisson



- Characteristics:
  - Jobs arrive in **"batch"**. (with similar runtime and cluster size)
  - The arrival intensity rate is varying

SWINBURNE UNIVERSITY OF TECHNOLOGY

M^t_x /G/∞ with Model Predictive Control

- **A(t)** is modelled as a **batch** Poisson process with time varying rate (M_x^t)
- Service time distribution is stationary
- Sufficient supply

Only consider K slots:

$$\min_{n(0),n(1),n(K-1)\ldots} \sum_{k=0}^{K-1} g(k)$$

- Only use ***n`(0)*** and discard the rest
- Use the same process at the next slot, i.e for slot k=1 to k = K+1

SWINBURNE UNIVERSITY OF TECHNOLOGY

Case A

Cost(t) = "On" Servers + server switching + Not meeting the demand

jobs that arrived after t = 0 and still running

$$cost(t) = n(t) + u(t) + E[(X(t) + U(t) - n(t))^+]$$

PD cost

jobs that arrived before t = 0 and still running

Case B

Cost(t) = "On" Servers + server switching

$$cost(t) = n(t) + u(t)$$

No. "ON" servers

No. "switching" servers

but apply the constraint:

$$P[\{X(t) + U(t)\} \geq n(t)] < 1 - \varepsilon$$

PD cost is decoupled to be a constraint

$M^t_x/G/\infty$ with Model Predictive Control

Case A: $\min_{n(0),n(1),n(K-1)..} \sum_{k=0}^{K-1} g(k)]$	Case B: $\min_{n(0),n(1),n(K-1)..} \sum_{k=0}^{K-1} \beta_1 n(k) + \beta_1 u(k)$ $s.t \Pr[\{x(t) + v(t)\} < n(k)] > 1 - \varepsilon$
• Unconstrained optimisation problem • PD cost is integrated in objective cost • Can be cast as shortest path problem and solved by Dynamic Programming	• Constrained optimisation problem • Solved using Linear Programming • The performance constraint is decoupled from objective cost • Cost is deterministic • Give explicit bound of performance (ε)

Solving case A:

No. "ON" servers

No. "switching" servers

X(t) is a compound Poisson RV

$$cost(t) = n(t) + u(t) + E[(X(t) + U(t) - n(t))^+]$$

PD cost

U(t) is a binominal RV

- Use Gaussian approximations $\tilde{X}(t)$ and $\tilde{U}(t)$
- Cost(t) can then be calculated based on:
  - service time distribution, batch size distribution
- As cost per stage cost(t) is now determined, use Dynamic Programming

Outline

- Control framework
- Future load estimation
- Energy efficiency for single core
- Future work and challenges

SWINBURNE UNIVERSITY OF TECHNOLOGY

Historical workload study

- Concentrate on the arrival process A(t)
  - estimating the future load
- Swinburne Supercomputer maintains accounting logs
  - keep job records (arrival time, runtime, cluster size, used resources, etc.)
  - guidance for workload modelling, simulation and validation.
- Attempts:
  - statistical models for interarrival time, batch size, runtime and arrival intensity
  - clustering users based on job statistics

SWINBURNE UNIVERSITY OF TECHNOLOGY

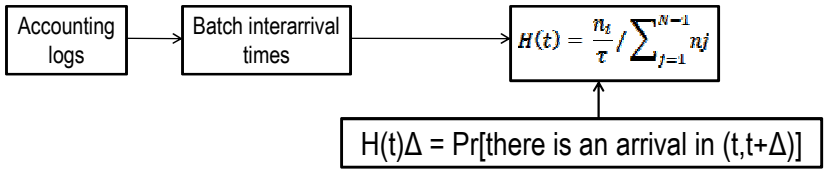
Future workload estimation

- Users have different submission behaviours
  - but it seems consistent for each user.
- Is knowing per user information helpful ?
  - method to predict future load
    - hazard rate function of interarrival time distribution
  - how to testify that ?
    - real trace experiment

SWINBURNE UNIVERSITY OF TECHNOLOGY

Future workload estimation

- Hazard rate function of the interarrival time distribution:

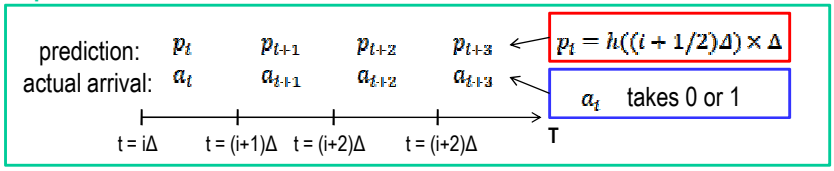


- The likelihood of having an arrival in short future, provided the time of the last submission

SWINBURNE UNIVERSITY OF TECHNOLOGY

Evaluating the load estimator

Experiment:



Length/Scenario	Not knowing user behaviours $\frac{\Delta}{T}\sum_i^{T/\Delta}(p_i^1 - a_i)^2$	Knowing per user behaviours $\frac{\Delta}{T}\sum_i^{T/\Delta}(p_i^2 - a_i)^2$
6 months	0.2207	0.1559
10 months	0.2114	0.1993

- ~ 5% improvement when knowing user behaviours
- the fitting is not perfect
- only predict the arrival likelihood
- the ultimate prediction is number of servers needs to be powered ON

Next steps:

- Predict the required servers
- How?: required servers = likelihood x average batch size

SWINBURNE UNIVERSITY OF TECHNOLOGY

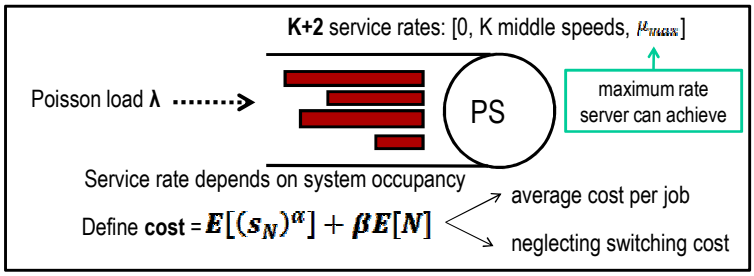
Outline

- Control framework
- Future load estimation
- Energy efficiency for single core
- Future work and challenges

SWINBURNE UNIVERSITY OF TECHNOLOGY

Energy efficiency in single core

M/G/1-PS:



Research questions:

1. Knowing system load λ_d, which service rates to use ?
2. The benefit of having more choices of rates (larger K)

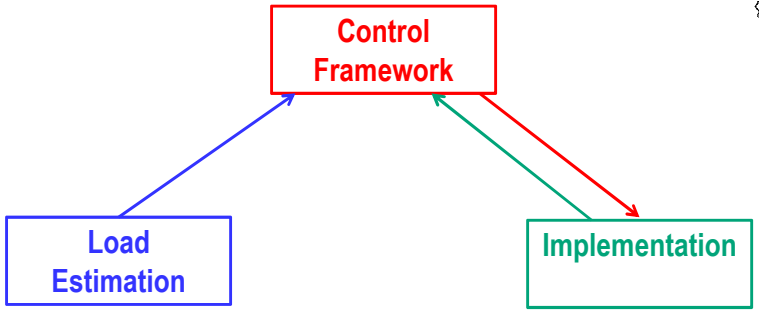
Energy efficiency in single core

Prior work	Our contributions
1. Not much cost improvement when K = ∞ comparing when server only runs at one optimal rate, if $\lambda_a = \lambda_d$. [Wierman. et al. 2009]	1. Given a λ_d , find the optimal rates for server with $1 \le K < \infty$ 2. Our numerical results suggest that the local robustness (small variation) increases as K increases.
2. When $\lambda_a \neq \lambda_d$, server with K = ∞ is more **robust** than server has only one (optimal) rate, in terms of **cost variation.** [Andrew. et al., 2010]	3. K = 8 is as good as K = ∞ in terms robustness 4. Flexibility in choosing rate in runtime gives significant cost improvement

Outline

- Control framework
- Future load estimation
- Energy efficiency for single core
- Future work and challenges

Future work



Control Framework – Future work

Control Framework

SOLVING ALL OPTIMIZATIONS

- Penalty of using Gaussian approximations
- Find suitable model for service time distribution (apply for both cases)

SIMULATION

- An event driven simulator has been written
  - Using the Green Machine scheduling policy
- Compare performance of the controller to the "halfway-house" heuristic
  - If found no improvement , examining where the energy savings is claimed.

Future work

Load Estimation

- Instead of predicting arrival likelihood, predict the requesting number of servers
  - still use the hazard rate function
  - mean batch size for each user
- Modelling the service time distribution (find the c.d.f) and the batch size distribution
  - starting with Weilbull, Gamma and Log-Normal distribution

Integrating the controller

- Challenges:
  - Not all machines can be powered OFF: storage, special reservation
  - The scheduler Moab is a commercial software
    - Require to understand Moab decision
    - But Moab logs are accessible
- Future tasks (time permitting)
  - Integrate the load estimation into the controller
  - Develop the mechanism to turn server ON and OFF
    - Not interfere with normal operations
  - Develop interfaces between the controller and Moab

Implementation

SWINBURNE UNIVERSITY OF TECHNOLOGY

Summary

- Power proportionality can achieve much energy saving for supercomputer.
- A control framework has been developed using Model Predictive Control, but it needs to be tested.
  - A controller is developed based on the framework
- An estimation of future workload is required in the control framework
  - Knowing per user behaviour can improve the estimation accuracy

SWINBURNE UNIVERSITY OF TECHNOLOGY