Energy savings for servers parks

Damien Borgetto

Jean-Marc Pierson
Georges Da Costa
Amal Sayah

IRIT - équipe ASTRE
Université Paul Sabatier

E2GC2
Octobre 2009
Large scale systems

- Provide computing resources to large scale groups of users or services
- Geographically distributed
- Hardware either heterogeneous or homogeneous
- Heterogeneous usage
Large scale systems

Energy point of view

- Electricity cost of a cluster for 1 year is the same scale as its price
- Electricity consumption impact on environment is important
- Possible to reduce this energy consumption at several levels
  - Infrastructure around the hardware
  - Hardware
  - Software (compilation, algorithmic)
  - Middleware
Introduction of a new metric: yield

\[ Y_{ij} = \sum_{j=1}^{H} \left( \frac{\alpha_{ij}}{\alpha_i} \right) \]

H: number of hosts  
\( \alpha_i \): fraction of the CPU the job \( i \) requested  
\( \alpha_{ij} \): fraction of the CPU of the host \( j \) actually allocated to job \( i \)  
Model the job yield, its satisfaction rate.

Example

Let’s take 2 hosts and 4 jobs requesting each 60% of CPU resources ($\alpha_i = 0.6$).

\[
Y_{ij} = \frac{0.25}{0.6} \approx 0.42
\]

\[
Y_{ij} = \frac{0.5}{0.6} \approx 0.83
\]
Objectif : maximize the minimum \textit{yield}

1. Apply an algorithm that takes a yield as an objective and computes a placement.
2. Iterate this algorithm to find the optimal yield.
Placement limits

- This approach allows to switch off hosts as a collateral effect.
- Does not take into account hosts heterogeneity.
- Does not take into account the energy required to compute the placement.

⇒ Energy has to be taken into account globally.
(H1) Hosts are heterogeneous energy wise, but provide the same computing resources.

(H2) An host consumes $C^{min}$ watts when idle and $C^{max}$ when fully loaded.

(H3) $\delta C_{ij} = \alpha_{ij} \times f(j)$ linear, $f(j) = (C^{max} - C^{min})$.

(H4) Infinite jobs.

(H5) No migration.
Metric redefinition to take into account energy consideration.

\[ YE_{ij} = \frac{(Y_{ij})^{1-k}}{(E_{ij})^k} \]

\[ = \frac{\left[ \sum_{j=1}^{H} \left( \frac{\alpha_{ij}}{\alpha_i} \right) \right]^{1-k}}{\left[ \lambda(C_j^{\text{max}} - C_j^{\text{min}}) \times \alpha_{ij} + (1 - \lambda) \left[ A_j(1 - \sum_{i'=1,i'\neq i}^{J}(\alpha_{i'j})) \right] \right]^k} \]

With \( 0 \leq \lambda \leq 1 \) and \( 0 \leq k \leq 1 \).
(P1) Favors an energy efficient host.

\[
\frac{\left[\sum_{j=1}^{H} \left( \frac{\alpha_{ij}}{\alpha_i} \right) \right]^{1-k}}{\left[ \lambda(C_j^{\text{max}} - C_j^{\text{min}}) \times \alpha_{ij} + (1 - \lambda) \left[ A_j(1 - \sum_{i' = 1, i' \neq i}^{J} \alpha_{i'j}) \right] \right]^k}
\]

With \(0 \leq \lambda \leq 1\) and \(0 \leq k \leq 1\).
\begin{equation}
(P2) \text{ Favors an already loaded host.}
\end{equation}

\[
\left[\sum_{j=1}^{H} \left(\frac{\alpha_{ij}}{\alpha_i}\right)\right]^{1-k}
\left[\lambda(C_j^{\text{max}} - C_j^{\text{min}}) \times \alpha_{ij} + (1 - \lambda) \left[A_j(1 - \sum_{i'=1, i' \neq i}^{J} \alpha_{i'j})\right]\right]^k
\]

With $0 \leq \lambda \leq 1$ and $0 \leq k \leq 1$
and $A_j = C^{\text{min}}$
(P3) Tradeoff between energy savings and performance ($k$)

(P4) Tradeoff between Placement and consolidation ($\lambda$)

\[
\left[ \sum_{j=1}^{H} \left( \frac{\alpha_{ij}}{\alpha_i} \right) \right]^{1-k} \left[ \lambda(C_j^{\text{max}} - C_j^{\text{min}}) \times \alpha_{ij} + (1 - \lambda)(A_j(1 - \sum_{i'=1, i' \neq i}^{J} \alpha_{i'j})) \right]^k
\]

With $0 \leq \lambda \leq 1$ and $0 \leq k \leq 1$. 

\[
\sum_{j=1}^{H} \left( \frac{\alpha_{ij}}{\alpha_i} \right)
\]
2 other differences with the base algorithm:

- New dichotomy bounds:

  \[ YE_{sup} = \frac{\left[ \min\left\{ \sum_{i=1}^{H} (\alpha_i) , 1 \right\} \right]^{1-k} }{\min_j \{ (1-\lambda) \times A_i + \lambda \times \min\{1, \sum_{i=1}^{J} (\alpha_i) \} \times (C_j^{max} - C_j^{min}) \}^k} \]

- Sort of the hosts at the algorithm beginning to place on the best hosts firsts:
  - TH1 : \( C^{min} \), increasing;
  - TH2 : \( C^{max} \), increasing;
  - TH3 : \( C^{min} + C^{max} \), increasing;
  - TH4 : \( C^{max} - C^{min} \), increasing;
• Same methodology as defined in Stillwell and al (to allow comparison).
• Small problem instances: 4 hosts, 1440 simulations.
• Campaign of more than a million simulations.
• Experimentation done on the french Grid’5000 platform.
Influence of the hosts sorts on the energy consumption

**Influence of the hosts sorts on the energy consumption**

**System Energy (Watts)**

**Increasing order**

**Parameters variation**

- TH1: $C^{min}$
- TH2: $C^{max}$
- TH3: $C^{min} + C^{max}$
- TH4: $C^{max} - C^{min}$
Improvement of the energy consumption and reduction of the yield
Energy consumption in function of $\lambda$ and $k$
Yield in function of $\lambda$ and $k$
Current state:

- Definition of a placement metric *energy-aware*
- Metric validation using simulation
Perspectives

**Short-term:**
- Job sorts *energy-aware*
- Model of the optimal placement with energy/performance constraint

**Mean-term:**
- Remove some hypotheses: performance homogeneity and infinite jobs
- Job migration
- Overhead estimation
- Heuristic comparison

**Long-term:**
- Autonomous system of energy management
Questions?

Questions
### Task placement

**Initial algorithm**

**Objective**: maximize the minimum *yield*

1. Fix a value $Y$ of the aimed *yield*;
2. For each jobs, deduce from $Y$ new CPU requirements to guaranty jobs attain this objective;
3. Sort jobs into two lists;
4. Place jobs using *multi capacity bin-packing*;
5. Restart 1, 2, 3 and 4 by doing a dichotomy on $Y$ to find the optimal $Y$.

**Several jobs sorts:**
- MCB4: $\text{max(\text{mémoire},CPU)}$, increasing;
- MCB8: $\text{max(\text{mémoire},CPU)}$, decreasing.