A Hybrid Cross-Entropy Cognitive-based Algorithm for Resource Allocation in Cloud Environments

Gaetano F. Anastasi, Pietro Cassarà, Patrizio Dazzi, Alberto Gotta, Matteo Mordacchini, Andrea Passarella

Summarized by Pranav Veldurthy
11/20/2017
Overview

• Introduction
• Models and Notations
• Architecture
• Algorithm
  o Cognitive Heuristics
  o Priority Heuristics
• Experimental Evaluation
• Conclusion
Introduction

• Cloud Computing is seen as utility computing
• Users often use providers offering advanced and tailored QoS
• Providers should consider heterogeneity and dynamicity.
• Self-Adaptive and Self-organizing can be effective
• Cognitive-based heuristics can be implemented to optimize which are more suitable in a particular time instance.
  o Cloud Resource Manager autonomously and adaptively estimate the risk.
  o Based on the risk estimation, cost and the selected risks are combined to form the best possible resource allocation
Introduction

• Cross Entropy algorithm combines the resource allocation and the associated cost to address the stochastic optimization problem.

• Contributions by the Resource Manager (RM):
  o Definition of a Cross-Entropy based approach for resource allocation
  o Exploitation of Cognitive Heuristics for choosing among different allocations
Model and Notations

Applications and Resources

• Applications has tasks that must be assigned to resources of the underlying infrastructure.

• The pdf (probability distribution function) is characterized by the $d_\tau$, $c_\tau$ and $\omega_\tau$ randomly from the previous measurements performed on different activations on the same task.

Infrastructure

• Modelled with a datacenter ($p$) with set of computer nodes.

• Physical machines ($m$) $\in \{1,\ldots,M\}$ characterized by its memory capacity $\omega_m$ and computing capacity $U_m$
Model and Notations

Problem Formulation
• Resource Manager inputs the set of tasks and machines and outputs a possible allocation.
• \( Y = \{ a_{ik} \} \)
• \( \sum_{k=1}^{M} a_{ik} = 1 \)

Objective Function
• Resource Usage Minimization
  o Minimize the number of machines used for executing tasks to reduce the amount of resources used by the Cloud provider.
• Threshold-Balanced Resource Usage
  o Elasticity without migration is guaranteed by allocating free resources for each machine with a certain value
• Allocation by Task Duration
  o Resource allocation is performed when certain amount of machines are free such that \( m \) spends same amount of time to complete the task allocation.
• Resource Manager activates $M_{\text{free}}$ resources.
• $T_{\text{ToMap}}$ are extracted from the queue and allocated according to the objective functions.
• Cognitive Heuristic informs about the availability of the resources and the task inputs.
• K different invocations with K different objective functions are performed returning a different allocation plan by the Cross-Entropy allocator.
• Best plan is chosen and enforced on the Cloud Resources.
Algorithm

• Cross Entropy Algorithm
  - Optimization Problem
    \[
    \min_{x \sim g(x)} \{F(x)\}
    \quad \text{s.t.}
    \sum_{j} res(m)_j \leq res_m, \forall m \in 1 \cdots M
    \sum_{\{j\}} m_j \leq M
    \text{time duration}(j_i) \sim f_i(\cdot)
    \text{usage resource}(j_i) \sim h_i(\cdot)
    \]
  - Stochastic Network Node (SNN)
    \[
    P_{g(x)} \{F(x) \leq \gamma\} = \mathbb{E}_{g(x)} \{1_{F(x) \leq \gamma}\}
    \]
Algorithm 1 Optimal pdf Algorithm

1. Generate the initial distribution $g(x)_0$ by which any machine can be assigned to any task with the same probability;

   while $D_{KL} (\hat{g}(x)^k, \hat{g}(x)^{k-1}) \geq \epsilon$ do

   2. Generate $S$ allocations of resources for the active tasks $\{x^1 \cdots x^S\} \sim g(x)^{k-1}$;

   For each $x^a$ are drawn from $f_i$ and $h_i$ the time duration and the amount of allocated resources for the active tasks, respectively;

   3. Evaluate the $(1-\alpha)$-quantile of the samples of allocation resources, for which the objective function is lesser than $\hat{\gamma}^{k-1}$;

   4. Update the estimation $\hat{g}(x)^{k-1}$ to $\hat{g}(x)^k$ through the samples that belong to the $(1-\alpha)$-quantile

   5. Evaluate the KL-Distance between the probability density functions at $k$ and $k - 1$

$k = k + 1$

end while

$\hat{\gamma} = \min \{ \alpha | P_{\hat{g}(x)}(F(x)) \leq \alpha \} \leq \alpha$

Kullback-Leibler distance

$D_{KL}(\hat{g}(x)^k, \hat{g}(x)^{k-1}) = \sum_i g_i^{k-1} \log \left( \frac{\hat{g}_i^k}{\hat{g}_i^{k-1}} \right)$

$\hat{g}_j^k(m) = \frac{\sum_{s=1}^S \mathcal{I}_{\{F(x^s) \leq \hat{\gamma}^k\}} \mathcal{I}_{\{m\}}}{\sum_{s=1}^S \mathcal{I}_{\{F(x^s) \leq \hat{\gamma}^k\}}}; m = 1 \cdots M; j = 1 \cdots J;$
Cognitive Heuristics

• Cross entropy allows the Resource Manager to determine the configuration satisfying the constraints for each available objective functions.
• Problem is evaluated based on cost and how the function is generated such that it wont lead to optimal solution.
• Resource Manager should autonomously fulfill two tasks.
  o Determine the risks associated with each objective function.
  o Optimal configuration must be chosen by considering costs and risks.
Cognitive Heuristics

• It is assumed that the RM maintains a memory of the previous usage of each objective function.
• Memory is modelled based on the ACT-R (Adaptive Control of Thought—Rational) memory model:

\[
A_i(t) = \ln \left[ \sum_{k=1}^{K} (t - t_k)^{-d} \right]
\]

• For a given objective function, the associated activation time \( t \), \( A_i(t) \):

\[
A_i(t) = \ln \left( \frac{p_i^t}{1 - p_i^t} \right)
\]
Cognitive Heuristics

• RM should evaluate the specific risk allocated to objective function.
• The probability of risk associated for not needing the objective function:

\[ r_i^t = 1 - p_i^t = \frac{1}{1 + e^{A_i(t)}} \]

• The RM can now make the final decision among the proposed allocations based on the cost and risk associated to each objective function.
Priority Heuristics

- Priority Heuristic is a cognitive model based on the risky choices made by the humans were choices are characterized by a gain (loss or cost) and a probability to achieve it.

\[ r_i^t = 1 - p_i^t = \frac{1}{1 + e^{A_i(t)}} \]

- Uses a lexicographic heuristic, i.e., each attribute is used only if the previous one was not sufficient to determine the final outcome.

```latex
\begin{algorithm}
\caption{Priority Heuristic at time $t$}
1: Let $F = \{F_1, \ldots, F_k\}$ be the set of obj. functions
2: Let $A = \{A_1, \ldots, A_k\}$ be the set of poss. allocations
3: Let $C = \{C_1, \ldots, C_k\}$ be the set of associated costs
4: Compute $R = \{r_1^t, \ldots, r_k^t\}$, the set of assoc. risks
5: Let $C_{max} = \max_{i \in [1,k]} C_i$
6: Let $\theta_c = C_{max}/10$
7: Let $S_1 = \{A_i | C_i < \theta_c\}$
8: if $|S_1| = 1$ then
9: Return $S_1$
10: else if $S_1 = \emptyset$ then
11: \textbf{$S_1 = A$}
12: end if
13: Let $R_{min} = \min_{i \in S_1} r_i^t$
14: Let $S_2 = \{A_i \in S_1 | r_i^t = R_{min}\} \cup \{A_j \in S_1, A_j \neq A_i | r_j^t - R_{min} \leq 0.1\}$
15: if $|S_2| = 1$ then
16: Return $S_2$
17: else
18: Let $S_3 = \{A_i \in S_2 | C_i = \min_{j \in [1,k]} C_j\}$
19: Return $S_3$
20: end if
\end{algorithm}
```
Experimental Evaluation

• Experiments are conducted to give better mappings between applications and machines.
• Google cluster data is used to characterize the workload. Although anonymized, can be used to evaluate resource management strategies.
• Data owners refuse to share data due to the risk of reverse engineering trade secrets (size, center and performance).
• Data is collected over a period of six hours and 15 minutes.
• Each row in the data set during a five minute interval has:
  o **Time** – time in seconds (int)
  o **JobID** – unique job identifier
  o **TaskID** – unique task identifier
  o **JobType** – job type, an integer in [0, 1, 2, 3], a representation of job semantics
  o **Normalized Task Cores** – Normalized number of CPU cores used
  o **Normalized Task Memory** – Normalized value of the average memory consumed by the task
Experimental Evaluation

• The trace contains 75 five-minute reporting intervals. There are a total of 3,535,029 observations, 9,218 unique jobs and 176,580 unique tasks.

• From the data exploitation, similar assumptions made by Chen et al were made. The assumptions made on some tasks are:
  o Tasks have different job type markings during different time intervals
  o Tasks have time gaps
  o Tasks use zero cores but non-zero memory
  o Tasks use zero core and zero memory
Experimental Evaluation
Experimental Evaluation

• Assumption is made such that RM is activated when
  o The number of machines available is 100
  o Maximum number of task extracted from the queue is 1000 clustered into four classes 0-3.

• Developed in MATLAB.

• The final cost of a configuration is expressed by the sum of each objective functions.

• The final configuration is chosen on the mean value analysis based on two simple strategies
  o Allocation is done randomly on the available machines with check constraints on the maximum computational and memory capabilities.
  o Comparisons are made using three different solutions referring to three different objective functions.
Experimental Evaluation

- $d = 0.5$
Experimental Evaluation

- $d = 0.1$
Experimental Evaluation

- $d = 1.0$
Conclusion

• The proposed Cognitive based heuristics can overcome the lack of information by reproducing human behavior dealing with risk minimization and cost optimizations.

• The resource allocation is implemented as a stochastic optimization problem solved using the Cross-Entropy method.

• The implemented approach is capable of auto-configuration to chose the best objective function outperforming the preconfigured or random choices.