Resource Allocation with a Budget Constraint for Computing Independent Tasks in the Cloud

Weiming Shi and Bo Hong

School of Electrical and Computer Engineering
Georgia Institute of Technology, USA

2nd IEEE International Conference on
Cloud Computing Technology and Science
Nov. 30 – Dec 3, 2010
Indianapolis, USA

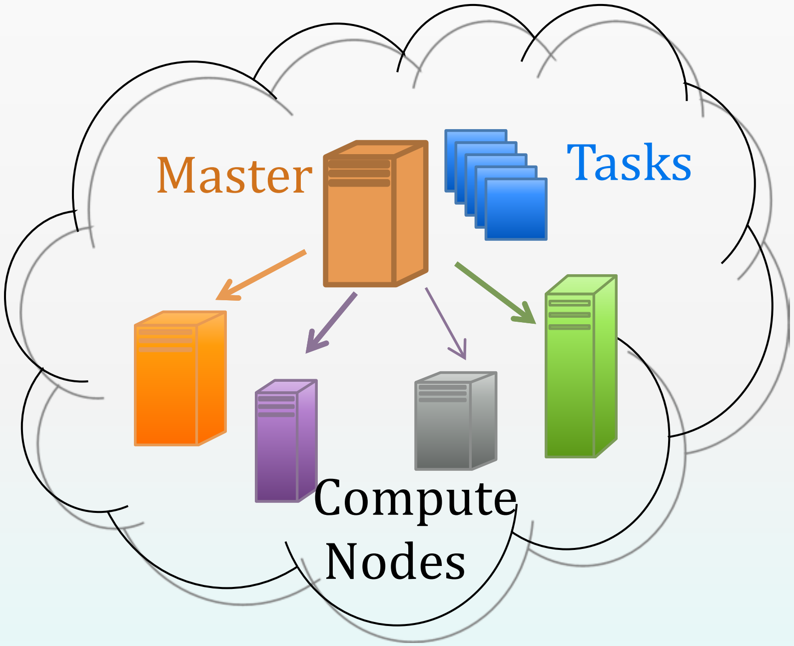
Outline

- Introduction
- System Model
- Problem Formulation
- Solution
- Simulation
- Conclusion and Discussion

Introduction

- **Motivation:**
  - Explore the resource allocation scheme from the perspective of the cloud users.
  - How to achieve the maximum return under the limited budget?
- **Approach:**
  - Consider the problem of running a large number of independent equal-sized tasks on the cloud infrastructure under the budget constraint.
  - Formulate and solve the problem based on a modeled cloud infrastructure.

Introduction



- Centralized work paradigm
- An application consisting of a large amount of independent, equal-sized tasks
- The granularity of the application is one task
- One-round distribution fashion
- Virtualized compute nodes with different CPU frequency, interconnect bandwidth and monetary charge rate

Introduction

- Previous works try to optimize their domain-specific utility function over the system parameters such as the CPU frequency, the memory size, the network bandwidth.
  - The bandwidth-centric allocation scheme favors the compute nodes with the maximum interconnect bandwidth.
- Things change when a new metric: the *monetary charge rate* is taken into account.

Introduction

- The application under our consideration embodies the divisible workload model.
  - Fundamental basis of the potential applications that can be ported to run on the cloud
- A natural approach to the problem is to minimize the ***makespan*** or the ***total-completion-time*** of all the tasks under the budget constraint.
  - Cloud users are usually charged by time
- However, these problems proved to be ***NP-complete***.
- An alternative approach is to maximize the ***steady-state throughput*** of the system.

Introduction

- A system with a **one-round** distribution fashion typically undergoes three stages:
  - **Start-up stage**
    - Some compute nodes are idle because they have not received the tasks to be processed
  - **Steady-state stage (Periodic stage)**
    - All the compute nodes are all fed with tasks and the amount of time spent on communication and computation become stable
  - **Clean-up stage**
    - Some compute nodes become idle again after finishing the assigned tasks while other compute nodes are still busy working on the assigned tasks

Introduction

- ***The steady-state throughput***
  - the number of tasks that can be completed by the allocated computing resources per time period in the steady state without taking into account the start-up and the clean-up stages of the application
- ***The budget-constrained steady-state throughput maximization problem* is a reasonable approximation of the *budget-constrained makespan minimization problem.***
  - the amount of tasks to be processed is huge
  - the time spent on the start-up and the clean-up stages become negligible compared with the overall computing time spent in the steady state

Outline

- Introduction
- System Model
- Problem Formulation
- Solution
- Simulation
- Conclusion and Discussion

System Model

- A cloud computing infrastructure typically consists of
  - the underlying data centers with virtualized computing resources
  - the storage nodes that host the tasks and the associated data to be processed
  - interconnect network equipments
- Assume that there is only one edge (communication link) between the master node and any compute node.

System Model

- The cloud infrastructure can be modeled as a node-weighted edge-weighted star-shaped graph $G = (V, E, B, P)$
  - $V = \{C_i \mid i \neq 0\}$: the set of allocated compute nodes
  - $E = \{e_i\}$: the set of edges (communication links) between C_0 and C_i
  - $B = \{b_i\}$: the maximum # of tasks transmitted from C_0 to C_i per time unit, whose value captures the difference in the communication bandwidths between C_0 and C_i
  - $P = \{p_i\}$: the maximum # of tasks finished by C_i per time unit, whose value captures the difference in the computing power of the compute nodes

Communication/Computation Model

- **Master node**
  - Multi port communication model would turn the problem to be NP-complete again!
  - The single port communication
  - No computation on the master node
- **Compute nodes**
  - Non-overlap communication model
  - No communication between each other as the tasks are assumed to be independent

Cost/Budget Model

- **The cost model**
  - Linear: The monetary charge rate m_i is proportional to the computing power p_i
  - Logarithm cost model: Model the scenario when the cloud service provider tries to promote the use of compute nodes with the better computation performance
- **The budget model**
  - Proportional: the budget (per time period) is proportional to the number of available compute nodes
  - Constant: the budget (per time period) is held constant regardless of the number of available compute nodes

Outline

- Introduction
- System Model
- Problem Formulation
- Solution
- Simulation
- Conclusion and Discussion

Problem Formulation

- Constraints under consideration:
  - **the conservation property of the steady state**, i.e., all the tasks received from the master node by any allocated compute node should be consumed by itself.

$$b_i t_i = p_i t_i^{'}$$

  - **the non-overlap communication and computation model**, i.e., the communication time and the computation time of any compute node can not overlap, and the sum of which can not exceed one time period

$$t_i + t_i^{'} \leq T$$

  - **the single-port communication model of the master node** indicates that the sum of the communication time of the allocated compute nodes can not exceed one time period.

$$\sum_{i=1}^{k} t_i \leq T$$

Problem Formulation

- Constraints under consideration (continued):
  - **the limited interconnect bandwidth** of the master node, i.e., the number of tasks that the master node can transmit to the allocated compute nodes during one time period is limited

$$\sum_{i=1}^{k} b_i t_i \leq B$$

  - **the monetary constraint** imposed by the limit of available budget, i.e., the money spent on the allocated compute nodes should not exceed the available budget per time period

$$\sum_{i=1}^{k} (t_i + t_i^{'}) m_i \leq M$$

Problem Formulation

- The steady-state throughput can be expressed as

$$R = \sum_{i=1}^{k} R_i$$

- The set of constraints:

$$(1)\quad R_i \le (b_i^{-1} + p_i^{-1})^{-1} \text{ for } 1 \le i \le k$$

$$(2)\quad \sum_{i=1}^{k} b_i^{-1} R_i \le 1$$

$$(3)\quad \sum_{i=1}^{k} h_i R_i \le 1$$

$$(4)\quad \sum_{i=1}^{k} R_i \le B$$

  - $R_i = b_i t_i$, the throughput contributed by compute node C_i
  - $h_i = M^{-1}(b_i^{-1} + p_i^{-1})m_i$, the ratio of the cost of finishing one task on C_i to the available budget M per time period

Outline

- Introduction
- System Model
- Problem Formulation
- Solution
- Simulation
- Conclusion and Discussion

Solution

- A linear programming problem *generally* does not have the analytic (closed-form) solution
  - No straightforward heuristic exists
- Under certain circumstances, the analytic solutions do exist
  - We identify two modes of the system wherein the analytic solutions exist
  - These solutions give us the straightforward heuristics to allocate compute nodes

Solution

- The solution to the original problem can be shown to be $R_m = \min(R_s, B)$
- R_s is the solution to the auxiliary problem:
  - Maximize $R = \sum_{i=1}^{k} R_i$, subject to:

$$(1)\quad R_i \le (b_i^{-1} + p_i^{-1})^{-1} \text{ for } 1 \le i \le k$$

$$(2)\quad \sum_{i=1}^{k} b_i^{-1} R_i \le 1$$

$$(3)\quad \sum_{i=1}^{k} h_i R_i \le 1$$

Solution

- Based on the relationship between the communication-to-computation ratio $\lambda_i = b_i / p_i$ and the monetary charge rate m_i, we identify two modes where closed-formed solutions exist.
- ***Budget-bound***: $\lambda_i > M / m_i - 1,\ 1 \le i \le k$
- ***Communication-bound***: $\lambda_i < M / m_i - 1,\ 1 \le i \le k$

Solution

- *When the system is **budget-bound** (resp. communication-bound) :*
  - *Sort the compute nodes by the **benefit-first** heuristic h_i (resp. **communication-first** heuristic b_i)*
  - *The maximum steady-state throughput can be obtained by sending the tasks to nodes in the order of increasing h_i (resp. decreasing b_i)*

Outline

- Introduction
- System Model
- Problem Formulation
- Solution
- Simulation
- Conclusion and Discussion

Simulation Setup

- Simulation is done in Matlab
- The simulated star-shaped graph consists of one master node and k compute nodes.
  - To test the scalability of the heuristics, k is set to be $10 \cdot 2^{l}$ ($l = 0,1,\ldots,8$)
- Four different computing powers are simulated by randomly picking the values from the set $\{v_{p,}\ 2v_{p,}\ 4v_{p,}\ 8v_{p}\}$ with equal probability
- Trigger the different modes of the system by setting the values of the corresponding bandwidths
- Compare the simulation results of our proposed heuristics with other straightforward heuristics

Proportional (resp. Constant) Budget and Linear Cost Model

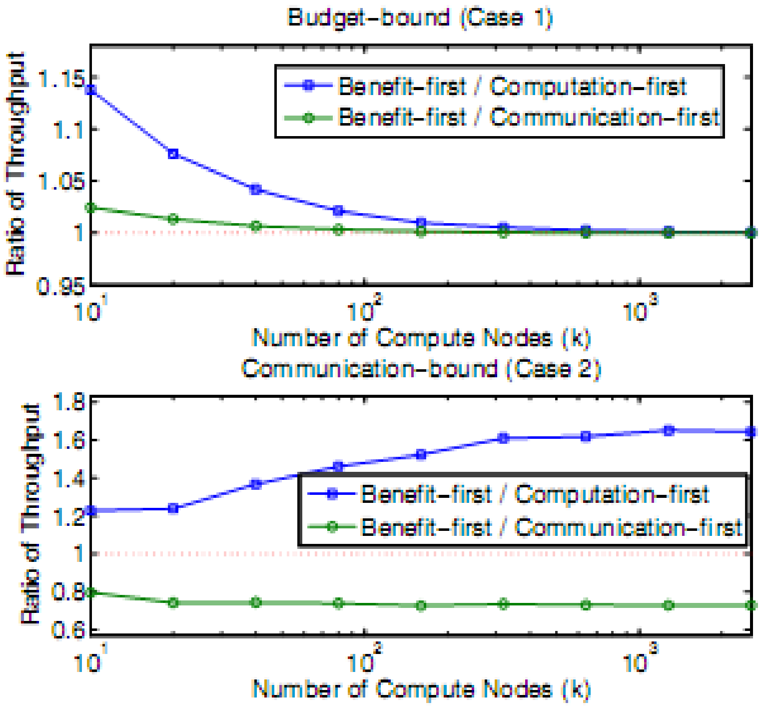


Figure 1. Comparison of the heuristics with the proportional budget and the linear cost model

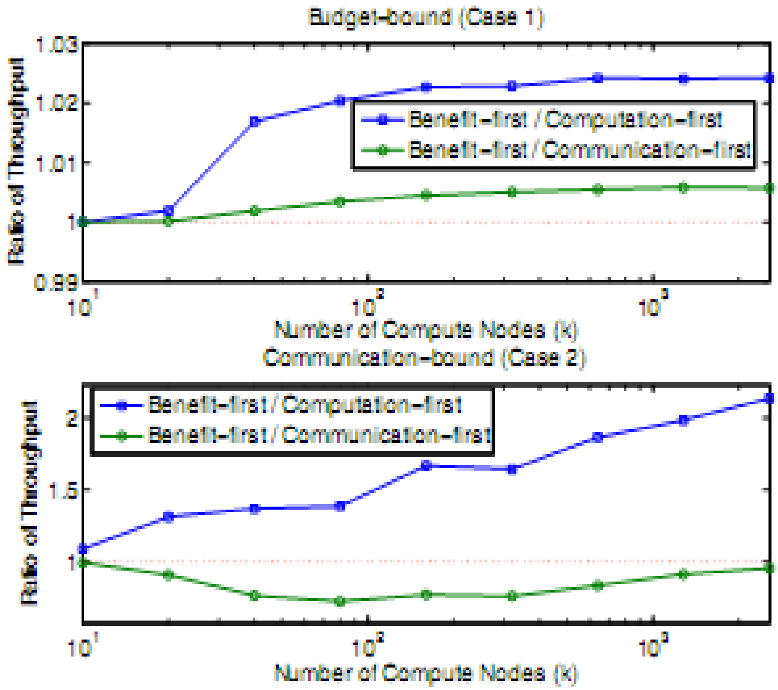


Figure 2. Comparison of the heuristics with the constant budget and the linear cost model

Proportional (resp. Constant) Budget and Logarithm Cost Model

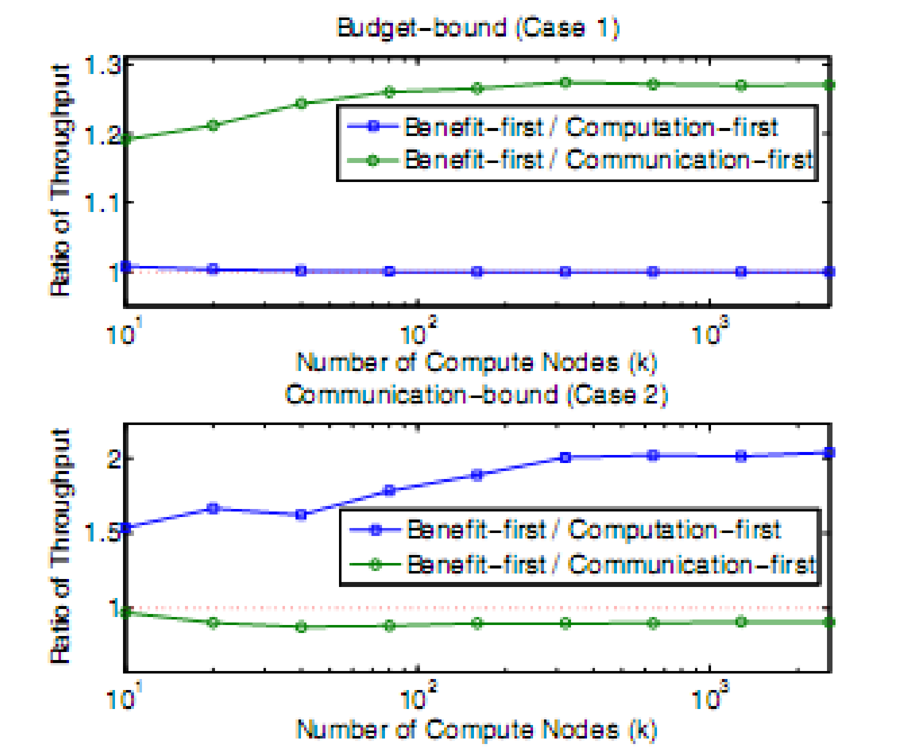


Figure 3. Comparison of the heuristics with the proportional budget and the logarithm cost model

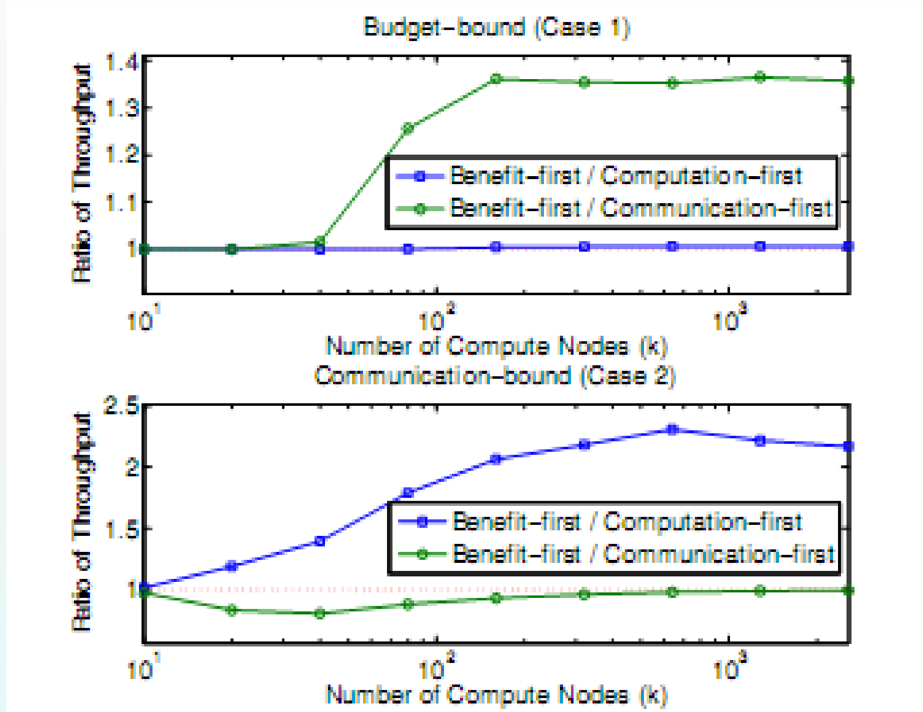


Figure 4. Comparison of the heuristics with the constant budget and the logarithm cost model

Outline

- Introduction
- System Model
- Problem Formulation
- Solution
- Simulation
- Conclusion and Discussion

Conclusion and Discussion

- Our intial goal has been reduced to the problem of maximizing the steady-state throughput of the allocated compute nodes in the cloud under the budget constraint.
- This problem can be formulated and solved efficiently as a linear programming problem under our model.
- We identify two modes of the system: ***budget-bound*** *and* ***communication-bound***
- The allocation scheme should be *benefit-aware*.
  - When the system is *budget-bound*, the *benefit-first* heuristics is the best
  - When the system is *communication-bound* , the *communication-first* heuristic is the best

Conclusion and Discussion

- The communication capacity has not been included in the cost model.
- Our model did not directly consider the dynamic nature of the cloud computing platform and cost spent on the start-up and clean-up stages.
- Yet, we provide an analytical framework and highlight an important metric that needs to be incorporated into the resource allocation scheme for the benefit of the cloud users.

- Q&A

Thank You