Modeling and Optimization of Resource Allocation in Cloud
PhD Thesis Progress – First Report

Atakan Aral

Thesis Advisor: Asst. Prof. Dr. Tolga Ovatman

Istanbul Technical University – Department of Computer Engineering

January 8, 2015
Outline

1 Introduction
2 Contribution to the Thesis
   - Studies
   - Time Plan
3 Details of the Studies
   - Data Collection
   - Simulator Development
   - Performance Criteria
   - Baseline Methods
4 Preliminary Evaluation
5 Future Work
Summary

- Optimization on 3 related phases of the cloud software life cycle
  1. Map Reduce Configuration
  2. Resource Selection and Assignment
  3. Work Distribution and Load Balancing

- Graph based modeling of the cloud environment and RA problem

- Inclusion of cloud computing aspects to the RA problem

- Cloud providers will benefit since they will be able to use their resources more efficiently and will be able to serve more customers without violating the SLAs.

- Cloud users will also benefit in terms of shorter application execution time and less infrastructure requirement (lower cost).
Life Cycle of Cloud Software

Development of Distributed Software
- MapReduce Configuration
  Phase 2

Resource Allocation
- Resource Selection and Optimization
  Phase 1

Load Balancing
- Distribution of Work to Resources
  Phase 3
Motivation

The cost of using 1000 machines for 1 hour, is the same as using 1 machine for 1000 hours in the cloud paradigm (Cost associativity).

Aim

Maximizing the utilization of all nodes for a Hadoop job by calculating the optimum parameters i.e. number of mappers and reducers

- Optimum parameters depend on the resource consumption of the SW.
- Design model of the software will be statically analyzed in order to guess resource consumption pattern and bottleneck resources.
Motivation
In distributed computing environments, up to 85 percent of computing capacity remains idle mainly due to poor optimization of placement.

Aim
Optimally assigning interconnected virtual nodes to substrate network with constraints

- VN requests and DC network will be represented as weighted graphs.
- Graph similarity – subgraph matching algorithms will be employed.
- There are several possible constraints to consider / optimize.
Motivation

Data flows between nodes only in the shuffle step of the MapReduce job, and tuning it can have a big impact on job execution time.

Aim

Analysis of the shuffle and scheduling algorithms of Apache Hadoop framework

- Selection of node roles is critical to reduce network traffic.
- Dynamic re-distribution of work during execution results in better load balancing and better performance.
- Cost based analysis of the existing schedulers will be performed.
We decided to use a real-world cloud datacenter topology for the experiments. Performed tasks include:

- Finding candidate topologies by examining similar studies
- Choosing the most appropriate topology
- Converting the topology into a machine-readable format
Simulator Development

- CloudSim Framework is chosen to simulate the cloud infrastructure.
- A general purpose tool which allows:
  - DC and VM creation
  - DC topologies
  - Load generation
  - Simple scheduling and logging
- Improvements and extensions:
  - Interrelated VMs and VM topologies
  - Delayed and continuous load generation
  - Automated topology creation
  - Visualization of resource allocation
Experimental Setup

- 9 baseline methods (6 of them are implemented) for resource selection and VM placement
- 16 metrics (10 of them are implemented) to evaluate the performance of these methods
- Analysis and hierarchical categorization of the metrics
- Analysis of the baseline methods and their expected effects on the metrics
Preliminary Evaluation

- Validation of the baseline methods
- Comparison of the baseline methods and the initial version of the suggested algorithm
- Validation and improvement of the suggested algorithm
- Evaluation of different configurations on each method’s performance
Introduction
Contribution to the Thesis
Details of the Studies
Preliminary Evaluation
Future Work

Studies
Time Plan

Gannt Chart

<table>
<thead>
<tr>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
</tr>
</tbody>
</table>

- Data Collection
- Simulator Development
- Experimental Setup
- Preliminary Evaluation
- Documentation

Atakan Aral
Modeling and Optimization of Resource Allocation in Cloud
Other Approaches

- Randomly generated DC topology [Ilyashenko, 2008] and [Ghazar, 2013]
- Proprietary data [Agarwal et al., 2010]
- DC topology generated based on cities, their populations and locations [Larumbe and Sanso, 2013]
- GÉANT network map
  - A European research and education network that interconnects National Research and Education Networks
FP-7 Project completed in 2010
- Hosts experimental activities on new Internet architectures and protocols
- Available data:
  - 4 core and 10 non-core Point of Presences
  - Dedicated 1 Gbps channels
  - Core PoPs contain 2+ virtualization nodes (servers) with VMware
  - Non-core PoPs contain 1 virtualization nodes (servers) with VMware
  - Node and link specifications
Output format of the Boston University Representative Topology Generator

```
Topology: (15 Nodes, 21 Edges)
Model (1 - RTWaxman): 5 5 1 2 0.150000000536046448 0.200000000298023224 1 1 10.0 1024.0

Nodes: (15)
0 1 3 4 4 -1 GARR
1 1 3 4 4 -1 DFN
2 1 3 4 4 -1 CESNET
3 1 3 5 5 -1 PSNC
4 1 3 1 1 -1 FCCN
5 1 3 2 2 -1 GRNET
6 1 3 3 3 -1 HEANET
7 1 3 2 2 -1 I2CAT
8 1 3 3 3 -1 ICCS
9 1 3 2 2 -1 KTH
10 1 3 2 2 -1 NIIF
11 1 3 1 1 -1 PSNC-2
12 1 3 6 6 -1 RedIRIS
13 1 3 3 3 -1 SWITCH
14 1 3 4 4 -1 NORDUNET

Edges: (21)
0 0 1 3.0 1.1 10.0 -1 -1 STANDARD
1 0 2 3.0 1.1 10.0 -1 -1 STANDARD
2 0 3 3.0 1.1 10.0 -1 -1 STANDARD
3 1 2 3.0 1.1 10.0 -1 -1 STANDARD
4 1 3 3.0 1.1 10.0 -1 -1 STANDARD
```
CloudSim

- A framework for modeling and simulation of cloud computing infrastructures and services [Calheiros et al., 2011]
CloudSim Features Benefited from

- Cloud entity classes: Datacenter, Host, Vm, Cloudlet
- Message passing mechanism for the communication among cloud entities
- Datacenter network topology
- Simulation
Design Decisions

- A host represents the aggregation of all computing capacity at a particular DC.
  - Hosts are mapped one-to-one to DCs.
  - Allows abstraction of host allocation within DCs
- A cloudlet represents the aggregation of all the load for a particular VM.
  - Cloudlets are mapped one-to-one to VMs.
  - Allows abstraction of cloudlet allocation within VMs
New CloudSim Features Implemented

- Automatic creation of datacenters and brokers at each location
New CloudSim Features Implemented

- **Interrelated VMs**
  - VM topology generation (Complete, Linear, Circular, Star, Custom)
  - Directed–undirected dependencies
  - Delay for communication

- **VM Life-cycle**
  - Deferred initialization for related VMs
  - Destruction after the completion of cloudlets

- **Statistics logging**
New CloudSim Features Implemented

- Visualization of the topology and allocation
Categorization

Quality of Service
- Temporal
  - Latency
    - AUL
    - MUL
    - ADL
    - MDL
  - Duration
    - HPC
    - JRT
    - JCT
    - WTT
    - SLA
    - UTR
    - TRP
    - CPR
  - Duration
    - HPC
    - JRT
    - JCT
    - WTT
    - SLA
    - UTR
    - TRP
    - CPR
  - Duration
    - HPC
    - JRT
    - JCT
    - WTT
    - SLA
    - UTR
    - TRP
    - CPR

- Volumetrical
  - UTR
  - TRP
  - CPR
  - DSF
  - LDB
  - RJR
  - FRN
  - PSP

- Distributional
  - DSF
  - LDB
  - RJR

- Other
  - DSF
  - LDB
  - RJR
  - FRN
  - PSP

Introduction
Contribution to the Thesis
Details of the Studies
Preliminary Evaluation
Future Work

Data Collection
Simulator Development
Performance Criteria
Baseline Methods
Average User Latency (AUL)

This criterion is the average time (in ms) it takes for cloud users to access their VMs. Here, average value is calculated both for the users and for their VMs. It should be measured with equal load of VM requests to compare two methods. A high value of user latency means the input and output transfer between user and the processing units is slow.
Maximum User Latency (MUL)

Even if the average user latency is low in a system, one of the users may be starving to access one of its VMs. To reveal such problems, we suggest the maximum user latency metric. Here, high values indicate starvation.
If two VMs are adjacent in the topology requested by the user, then these VMs need to transfer data or communication signals to each other. Average inter-DC latency is the time it takes to complete transfer between such nodes on average. It is critical to keep inter-DC latency low for the performance of the cloud software. Inter-DC latency is minimum when all VMs of a user are placed to the same DC.
Similar to maximum user latency, maximum inter-DC latency is used to detect outlier cases where it takes too much time to transfer data or communication signals between two VMs.
Hop count is the average number of hops per dependencies between VMs. Minimum values of this criterion (which is 0) is achieved when all dependent VMs in the request are placed to the same DC.
Job Run Time (JRT)

This criterion is the average time it takes to complete the jobs divided by their size (in million instructions - MI). Since a host has a finite number of CPU cores and a CPU core can be used by a single VM at a time, time-sharing is used when number of VMs in a host is greater than the number of CPU cores in that host. In such cases, running time of the job is longer because it can be executed only part-time and is paused at other times. For this criterion, we ignore the time spent in data transfer between user and VM since it is measured in latency criteria. Additionally, we consider each VM individually since the communication with other VMs also depends on latency.
Job completion time includes both latency and running times. It is an average value for all users but is not normalized by job size. It indicates the promptness in responding user requests.
Apart from latency, some VM requests will be delayed due to lack of available capacity. Waiting time criteria measures the time it takes until a job restarts after it is postponed.
Service level agreements (SLAs) usually define a deadline for job completion time and a penalty is issued if it is not respected. This criterion is the cumulative number of SLA violations due to tardiness.
Utilization rate is the percentage of the resources in a given DC topology that are assigned to VMs. It is measured after the first time the placement algorithm fails to assign a VM topology to any DC (resources achieve saturation). A different utilization rate exists for each type of resource e.g. memory utilization, bandwidth utilization etc. A high utilization rate indicates that the resources are assigned efficiently so that a great number of VM requests are responded successfully.
We measure throughput as the millions instructions processed in the whole system per second (MIPS). Indeed, it is equal to the total CPU utilization of all DCs. High throughput means a better placement of VMs and less wasted resources.
Cumulative Profit (CPR)

This criterion aims to represent the profit of the cloud provider in a fixed duration given prices for each resource type. Cumulative profit is directly proportional to throughput.
Distribution Factor (DSF)

Distribution factor is the extent that the VMs of a user are placed to separate DCs. We define distribution factor in Equation 1 where $U$ is the set of users whose VM requests are responded, $VM(u)$ is the VMs requested by the user $u$, $DC(u)$ is the DCs to which these VMs are placed. Range of the $DF$ is $(0, 1]$. }

\[
DF = \frac{\sum_{u \in U} |DC(u)|}{|VM(u)|} \quad (1)
\]

Higher values of distribution factor indicates that the VMs are highly distributed to separate DCs. Benefits of high distribution are high parallelization and low failure cost.
Load Balance (LDB)

This criterion is the evenness of load assigned to all DCs in terms of utilization rate. We use standard deviation to measure the amount of variation from the average utilization.
Rejection Rate (RJR)

This is the percentage of VM requests that are failed to responded immediately and postponed until completion of other requests make enough room in DCs.
Fairness (FRN)

Fairness is used to determine whether users are receiving a fair share of available resources. Since not all users require the same amount of resources, fair share does not mean equality in our case.
Path Splitting (PSP)

Path Splitting is the average number of inter-DC flows utilized to respond a inter-VM link request. This criterion is valid only if flow bifurcation is allowed. Otherwise, output is always 1.
Remarks

- Strategy design pattern is used for the DC selection policy.
- It is possible to decide and use a different policy for each VM group in the runtime.
- It is also possible to use different policies single VMs and VM groups.
- Each broker has a list of available DCs in an arbitrary order.
Baseline Methods

- Random Choice (RAN)
- Arbitrary First-fit (AFF)
- Arbitrary Next-fit (ANF)
- Latency based First-fit (LFF)
- Latency based Next-fit (LNF)
- Capacity based Best-fit (CBF)
- Load Balancing (LBG)
- Minimum Span (MSP)

- Shortest Path (SP) vs. Multicommodity Flow (MF)
Related VMs are considered as a group rather than independent individuals.

It tries to find a subgraph of the DC topology that is isomorphic to the requested VM topology.

Among the candidate subgraphs, VMs are assigned on the subgraph which has the lowest average latency from the broker.

Rejected VMs are assigned to the next DC that has the lowest latency from the accepted VMs’ DCs on average.
### Metric Expectations of the Baseline Methods

<table>
<thead>
<tr>
<th></th>
<th>AFF</th>
<th>ANF</th>
<th>LFF</th>
<th>LNF</th>
<th>CBF</th>
<th>TBF</th>
<th>LBG</th>
<th>MSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUL</td>
<td>▲</td>
<td>❄</td>
<td>▲</td>
<td>▲</td>
<td>✈</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>MUL</td>
<td>❄</td>
<td>❄</td>
<td>▲</td>
<td>❄</td>
<td>☁</td>
<td>☁</td>
<td>❄</td>
<td>❄</td>
</tr>
<tr>
<td>ADL</td>
<td>▲</td>
<td>❄</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>MDL</td>
<td>❄</td>
<td>❄</td>
<td>❄</td>
<td>❄</td>
<td>✈</td>
<td>☁</td>
<td>☁</td>
<td>❄</td>
</tr>
<tr>
<td>HPC</td>
<td>▲</td>
<td>❄</td>
<td>❄</td>
<td>❄</td>
<td>☁</td>
<td>☁</td>
<td>❄</td>
<td>❄</td>
</tr>
<tr>
<td>JRT</td>
<td>▲</td>
<td>❄</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>JCT</td>
<td>❄</td>
<td>❄</td>
<td>❄</td>
<td>❄</td>
<td>☁</td>
<td>☁</td>
<td>☁</td>
<td>❄</td>
</tr>
<tr>
<td>WTT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>SLA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>•</td>
<td>•</td>
<td>•</td>
<td></td>
</tr>
<tr>
<td>UTR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>TRP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>CPR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>DSF</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>LDB</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>RJR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>FRN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
<tr>
<td>PSP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>▲</td>
<td>▲</td>
<td>▲</td>
<td></td>
</tr>
</tbody>
</table>
Configuration

- DC MIPS: 1538 MIPS
- DC RAM: 16 GB
- DC Storage: 1 TB
- DC BW: 1 Gbps
- VM MIPS: 50 MIPS
- VM RAM: 1 GB to 8 GB
- VM Size: 10 GB
- VM BW: 10 Mbps
- Cloudlet Length: 150 MI
- Cloudlet Input: 15 MB
- Cloudlet Output: 15 MB
- Group Count: 2
- Group Size: 3 and 2
- Group Topology: Both Linear
Introduction
Contribution to the Thesis
Details of the Studies
Preliminary Evaluation
Future Work

Results

Atakan Aral
Modeling and Optimization of Resource Allocation in Cloud
Results

![Graph 1: ADL vs. VM RAM](image1)

- AFF
- ANF
- LBG
- LFF
- RAN
- TBF

![Graph 2: MDL vs. VM RAM](image2)

- AFF
- ANF
- LBG
- LFF
- RAN
- TBF
Results

Atakan Aral
Modeling and Optimization of Resource Allocation in Cloud
Results

Graph 1: TRP (MIPS) vs. VM RAM (MB)

Graph 2: RJR (%) vs. VM RAM (MB)
Results

![Graph 1](image1)

![Graph 2](image2)
## Changes to the Proposal

<table>
<thead>
<tr>
<th>2014</th>
<th>2015</th>
<th>2016</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Resource Selection
- **Research**
- **Modeling**
- **Development**
- **Documentation**

### MapReduce Configuration
- **Research**
- **Modeling**
- **Development**
- **Documentation**

### Work Distribution
- **Research**
- **Modeling**
- **Development**
- **Documentation**

---

**Atakan Aral**  
Modeling and Optimization of Resource Allocation in Cloud
Possible Directions

- MapReduce Configuration
- Resource Selection
  - More evaluation
  - Comparison against more advanced methods in the literature
  - A probability distribution for the cloudlet load
  - Time-sharing of the CPU
  - Vertical scaling
  - Focus on bandwidth allocation
  - Homeomorphism
- Multi-objective optimization
- Hybrid methods and Meta-heuristics
Thank you for your time.
MapReduce

**Definition**

A programming model for processing large data sets with a *parallel, distributed* algorithm on a cluster.
MapReduce Entities

- **DataNode**: Stores blocks of data in distributed filesystem (HDFS).
- **NameNode**: Holds metadata (i.e. location information) of the files in HDFS.
- **Jobtracker**: Coordinates the MapReduce job.
- **Tasktracker**: Runs the tasks that the MapReduce job split into.