Silverlining: A Cloud Forecaster Using Benchmarking and Simulation

Lawrence Chung  
Department of Computer Science  
The University of Texas at Dallas  
Dallas, Texas, USA  
chung@utdallas.edu

Tom Hill  
Department of Computer Science  
The University of Texas at Dallas  
Dallas, Texas, USA  
tom.hill.fellow@gmail.com

Nary Subramanian  
Department of Computer Science  
The University of Texas at Tyler  
Tyler, Texas, USA  
nsubramanian@uttyler.edu
Outline

• Motivation
• Cloud Configuration
• Goal-Oriented Approach
• NFR Approach
• Simulation-based Design
• Benchmark Testing on GAE
• CloudSim Simulations
• Conclusion
Motivation

• CIO would like to know run-time performance and operating costs for an application before deciding to migrate it to a cloud service.

• A cloud service provider would want to estimate how well the particular resource allocation will meet the performance and cost requirements.
Cloud Configuration

• IAAS, PAAS, SAAS
• Hardware components
• Operating system, database type
• Number of front ends, application instances, database connections
• RAM, Bandwidth, and CPU cycles used
Goal-Oriented Approach

• System development is aligned with the objectives (or goals)
• Performance and cost are considered primary goals for cloud-based system
• Each configuration is evaluated against the goals
The NFR Approach

• Non-Functional Requirements (NFR) approach is goal-oriented
• NFRs are quality attributes of a system e.g. adaptability, changeability, security, etc.
• Performance and cost are considered as NFRs
• The NFR Approach provides an evaluation framework
Softgoal Interdependency Graph

• NFR Approach uses SIG for evaluation
• SIG captures the relationship between different softgoals
• SIG captures the synergistic and conflicting contributions to softgoals
• Concept of satisficing used – satisfaction within limits instead of absolute satisfaction
Useful SIG
NFR Approach Ontology

- **NFR Softgoal**
  - **Operationalizing Softgoal**
  - **Claim Softgoal**

  **Strongly Positively Satisficing or MAKE Contribution**

  **Negatively Satisficing or HURT Contribution**

  **AND Decomposition**

  **OR Decomposition**

  **Criticality**

**Satisfied Softgoal**

**Weakly Satisfied Softgoal**

**Weakly Denied Softgoal**

**Denied Softgoal**
NFR Approach Steps

• Develop NFR softgoals and their decompositions (AND, OR, EQUAL)
• Develop operationalizing softgoals and their decompositions (AND, OR, EQUAL)
• Determine contributions between operationalizing softgoals and NFR softgoals (MAKE, HELP, HURT, BREAK)
• Assign labels (satisficed/denied)
• Evaluation and analysis – apply propagation rules
## Example Propagation Rules

<table>
<thead>
<tr>
<th>Source Label</th>
<th>Contribution</th>
<th>Destination Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satisfied</td>
<td>MAKE</td>
<td>Satisfied</td>
</tr>
<tr>
<td>Satisfied</td>
<td>BREAK</td>
<td>Denied</td>
</tr>
<tr>
<td>Denied</td>
<td>MAKE</td>
<td>Denied</td>
</tr>
<tr>
<td>Denied</td>
<td>BREAK</td>
<td>Satisfied</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Child Contributions</th>
<th>Decomposition</th>
<th>Parent Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>All satisfied</td>
<td>AND</td>
<td>Satisfied</td>
</tr>
<tr>
<td>One denied</td>
<td>AND</td>
<td>Denied</td>
</tr>
<tr>
<td>All denied</td>
<td>OR</td>
<td>Denied</td>
</tr>
<tr>
<td>One satisfied</td>
<td>OR</td>
<td>Satisfied</td>
</tr>
</tbody>
</table>
Issues

• In order to evaluate a proposed cloud configuration, we need
  – The label of the operationalizing softgoal
  – The contribution of this softgoal to performance and cost softgoals
  – The claim associated with the label
  – The claim associated with the contribution
Label Determination

• Is this a valid configuration?
• Is the application’s operating system supported?
• Is the application’s database supported?
• Does the expected usage profile fit the SLA?
• Do we satisfy copyright regulations?
• Are security/privacy requirements supportable?
Contribution Determination

• Use benchmarking to identify expected traffic parameters
• Execute benchmarks on the cloud to get performance data
• Run simulations to duplicate benchmark data
• Baseline simulation parameters for future environment modification
• Use answers from simulation for contributions
Claims Determination

• Justify the reason for label based on current knowledge
• Justify the reason for contribution based on simulation results
• If configuration changes
  – Re-evaluate label for the configuration
  – Re-evaluate contribution from the configuration
SIG Evaluation

• Continuous process
• Any change to SIG element data triggers the (re)evaluation process
• Latest performance and cost satisficing extent always known
• Based on this knowledge decide on the suitability of cloud configuration
Goal-Oriented Simulation-Based Design Process

1. Identify stakeholders and their goals, domain characteristics
2. Refine Stakeholder Goals and Analyze for Conflicts
3. Identify and Use Workload Characteristics from the Domain to Quantify Goals
4. Match Constraints and Workload to Standard Benchmark
5. Run Simulation and Refine Model Iteratively
6. Translate Simulation Model into System Architecture
7. Real-world testing and experimentation with derived architecture
Transaction Processing Benchmarks

Transaction Processing Benchmarks [TPC-C11]

+ Standard objective verifiable performance and cost OLTP, RDB since 1992
+ Business throughput metrics; number of orders processed per minute with cost
- OLTP and relational database only
- High cost to benchmark, high cost to customize

Transaction Processing Council Benchmarks [TPC-C13]

+ 274 client server benchmarks documented
+ 9 cloud benchmarks using Amazon cloud created by Stony Brook University
- No cloud benchmarks for Google, Microsoft, HP
Google Cloud infrastructure

Google Cloud Platform
Abstraction Layer

Silver lining
PaaS Abstraction
Layer

Mapping
Layer

Infrastructure Abstraction
Hardware/software Layer

Performance
Implementation
Hardware Layer

Google App Engine
Frontend / Backend Service

Google CloudSQL Database Engine Service

1. Google - Average Seek 10ms or 100 seeks/sec
2. MySQL - big tables log(rows 500,000)/log(index block bytes 1024/3*2/(index bytes 3 + pointer bytes 4)) + 1 = 4 seeks/read, 6 seeks/write
3. Application instruction path length aggregation [MIPs] can be matched with processor capacity [MIPs]
4. .....
TPC-C Benchmarks

<table>
<thead>
<tr>
<th>Rank</th>
<th>Company</th>
<th>System</th>
<th>Performance (tpmC)</th>
<th>Price/tpmC</th>
<th>Watts/Kilopmd</th>
<th>Database</th>
<th>Operating System</th>
<th>TP Monitor</th>
<th>Date Submitted</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ORACLE</td>
<td>GPARC TS-8 Server</td>
<td>6,552,533</td>
<td>.55 USD</td>
<td>N</td>
<td>Oracle Database 11g Release 2 Enterprise Edition</td>
<td>Oracle Solaris 11.1</td>
<td>Eagle</td>
<td>03/26/13</td>
<td>N</td>
</tr>
<tr>
<td>2</td>
<td>ORACLE</td>
<td>Sun Server X2-8</td>
<td>5,030,888</td>
<td>.69 USD</td>
<td>N</td>
<td>Oracle Database 11g R2 Enterprise Edition with Partitioning</td>
<td>Oracle Linux</td>
<td>Tuxedo CRS</td>
<td>03/27/12</td>
<td>N</td>
</tr>
<tr>
<td>3</td>
<td>IBM</td>
<td>IBM System x3650 X5</td>
<td>3,014,684</td>
<td>.39 USD</td>
<td>N</td>
<td>IBM DB2 ESE 9.7</td>
<td>SUSE Linux Enterprise Server 11.x</td>
<td>Tuxedo CRS</td>
<td>03/27/12</td>
<td>N</td>
</tr>
<tr>
<td>4</td>
<td>HP</td>
<td>ProLiant BL8655 G7</td>
<td>1,263,599</td>
<td>.31 USD</td>
<td>N</td>
<td>Microsoft SQL Server 2008 R2 Enterprise Edition x64 SP3</td>
<td>Microsoft Windows Server 2008 R2 Enterprise Edition</td>
<td>03/23/11</td>
<td>03/23/11</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>HP</td>
<td>ProLiant DL380 G7</td>
<td>1,024,380</td>
<td>.63 USD</td>
<td>N</td>
<td>Microsoft SQL Server 2008 R2 Enterprise Edition x64 SP3</td>
<td>Microsoft Windows Server 2008 R2 Enterprise Edition</td>
<td>03/24/11</td>
<td>03/24/11</td>
<td>N</td>
</tr>
</tbody>
</table>

Note: The TPC believes it is not valid to compare prices or price/performance of results in different currencies.
TPC-C Benchmark Example
Running Benchmarks

Google Cloud Graphical View of a Statistical Model to Estimate Performance and Cost of Running a Software Application

Conceptual diagram of Google Cloud Infrastructure
Transaction resource usage
Estimate of costs

Transaction characteristics
Production characteristics

Estimate of transaction throughput
User Interfaces

Google Cloud Project Benchmark “New Order” Transaction Window Design

1. Design the New Order transaction user interface in accordance with TPC-C benchmark specifications
2. Design and implement the New Order benchmark web program in Java and Python
3. Design and implement the remaining transactions
Executing Benchmarks on GAE

Google Cloud Project Benchmark Modified Stress-testing Tool to Generate 40 Concurrent User’s Transactions

Pylot.py, open source web stress testing tool, modified to generate TPC-C benchmark transactions with random database keys, keying time and think time

Number of concurrent users (agents) to generate transactions - 40

300 seconds benchmark duration – 300

Response time and throughput calculated and reported

Response message size in bytes

Statistics for users (agents) 1 through 40
Lab Setup

Google Cloud Project  UTD Benchmark Generating Lab Map - 500 Concurrent Users through 7,500 Users

A map of 15 Lab computers generating benchmark transactions for 500 users each

Google Cloud Project Benchmark  Requests per Second Strip-chart

Benchmark experiments with varying number of computers (2 versus 4 shown in the Google-provided strip-chart) generating the same total transaction volume. Test the lab sensitivity to generating environment changes. 2 computers versus 4 computers demonstrated no sensitivity (27 requests per second)
Results of Benchmark Testing

Google Cloud Project Benchmark Record of Experiments Example

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Users (count)</th>
<th>Duration (seconds)</th>
<th>@ISE Start</th>
<th># Users Accum</th>
<th>Duration Hours</th>
<th>Duration Minutes</th>
<th>Actual Start</th>
<th># GAE FE Instances</th>
<th># Requests Processed</th>
<th>Bandwidth Download Mbps</th>
<th>Bandwidth Upload Mbps</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>HM1</td>
<td>500</td>
<td>5100</td>
<td>0</td>
<td>500</td>
<td>85</td>
<td>1.42</td>
<td>1:42</td>
<td>255</td>
<td>92,660</td>
<td>56</td>
<td>20</td>
<td>Start of bench</td>
</tr>
<tr>
<td>HM2</td>
<td>500</td>
<td>4580</td>
<td>50</td>
<td>1,000</td>
<td>81</td>
<td>1.36</td>
<td>1:36</td>
<td>322</td>
<td>89,935</td>
<td>61</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>HM3</td>
<td>500</td>
<td>4880</td>
<td>100</td>
<td>1,500</td>
<td>81</td>
<td>1.35</td>
<td>1:35</td>
<td>161</td>
<td>87,215</td>
<td>63</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>HM4</td>
<td>500</td>
<td>4740</td>
<td>150</td>
<td>2,000</td>
<td>79</td>
<td>1.32</td>
<td>1:32</td>
<td>173</td>
<td>84,464</td>
<td>56</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>HM5</td>
<td>500</td>
<td>4620</td>
<td>200</td>
<td>2,500</td>
<td>77</td>
<td>1.28</td>
<td>1:28</td>
<td>445</td>
<td>81,629</td>
<td>65</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>HM6</td>
<td>500</td>
<td>4590</td>
<td>250</td>
<td>3,000</td>
<td>75</td>
<td>1.25</td>
<td>1:25</td>
<td>404</td>
<td>79,392</td>
<td>60</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>HM7</td>
<td>500</td>
<td>4380</td>
<td>300</td>
<td>3,500</td>
<td>73</td>
<td>1.22</td>
<td>1:22</td>
<td>487</td>
<td>76,369</td>
<td>59</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>HM8</td>
<td>500</td>
<td>4260</td>
<td>350</td>
<td>4,000</td>
<td>71</td>
<td>1.18</td>
<td>1:18</td>
<td>482</td>
<td>73,614</td>
<td>60</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>HM9</td>
<td>500</td>
<td>4140</td>
<td>400</td>
<td>4,500</td>
<td>69</td>
<td>1.15</td>
<td>1:15</td>
<td>559</td>
<td>70,932</td>
<td>60</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>HM10</td>
<td>500</td>
<td>4020</td>
<td>450</td>
<td>5,000</td>
<td>67</td>
<td>1.12</td>
<td>1:12</td>
<td>577</td>
<td>66,282</td>
<td>59</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>HM11</td>
<td>500</td>
<td>3900</td>
<td>500</td>
<td>5,500</td>
<td>65</td>
<td>1.08</td>
<td>1:08</td>
<td>605</td>
<td>65,940</td>
<td>60</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>HM12</td>
<td>500</td>
<td>3780</td>
<td>550</td>
<td>6,000</td>
<td>63</td>
<td>1.05</td>
<td>1:05</td>
<td>628</td>
<td>61,348</td>
<td>55</td>
<td>20</td>
<td>Level Instances</td>
</tr>
</tbody>
</table>

Lab computer name
Accumulated number of users
Number of Front End instances assigned by Google

Benchmark Data Collection Resource
Usage Time Strip-charts Provided by Google

6,000 concurrent users level benchmark
Cloud FE instances 628
240 requests per second
CloudSQL number of reads and writes
Plots from GAE
Benchmark Result Analysis

Google Cloud Project Preliminary Benchmark Results for a D1 CloudSQL Database Instance

The performance throughput knee of the D1 CloudSQL server (338.1 transactions per minute with 320 concurrent users)

Benchmark throughput in transactions per minute (tpmC) for a variable number of concurrent users

The number of Front End instances allocated by the Google Cloud with pricing
Simulation Forecaster Design

Google Cloud Project Simulation Forecaster Function Design

Use XML that describes the experiment as function input and output performance metrics and cost.

Workload requests represent resource usage and architecture infrastructure components represent capacity.

Simulator generates workload of multiple users and collects metrics.

1. Encode a Simulation / Forecaster Function
   The function $f(g_p, W_f, a_j) \rightarrow (p_{mr}, p_d)$ describes Simulation/Forecaster. Where:
   - $g_p$ the input stakeholder performance goals
   - $W_f$ the input enterprise workloads
   - $a_j$ the input resource architecture
   - $p_{mr}$ the output performance metrics
   - $p_d$ the output performance costs

2. Model the Application Workload Resource Usage and Architecture Resource Capacity
   - Multiple Web Clients
   - Channel/Internet
   - Google App Engine Frontend
   - Google CloudSQL Database Engine
   - Performance Goals
   - Request Messages
   - Response Messages
   - SQL Request Statements
   - SQL Response Messages

3. Construct a Simulation/Forecaster to Produce a Performance Metrics and Costs Report
   - Workload Generation & Metrics Collection
   - FE Wait Queue Time
   - FE Service Time
   - DB Wait Queue Time
   - DB Service Time

Queues are created for a finite capacity model

Service times combine usage and capacity
Simulation Setup

Google Cloud Project Simulation Forecaster Experiment Three-step Process

Graphic User Interface used to design the simulation experiment; by describing: goals, application workload and the components of the infrastructure

The GUI generates the description of the simulation experiment in XML

XML is used as input to a discrete event simulator to produce a report of performance (throughput) and cost
GUI for Simulator
XML for Simulator

```
<simulationrun>
  <runtitle>TEST for New 3.3GB D1 DB Server 320 users 03/09/2013</runtitle>
  <simgoals>
    <responsesecs>5</responsesecs>
    <tpmc>400</tpmc>
  </simgoals>
  <applicationgroup>
    <grouptitle>TPCC-Benchmark</grouptitle>
    <operationdays>mtwtf</operationdays>
    <operationhoursday>24</operationhoursday>
    <houptimezone>PST</houptimezone>
    <databasename>TPCC Data Tables</databasename>
    <dbsizegbyte>3.3</dbsizegbyte>
  </applicationgroup>

  Similar application workload descriptions available for:
  Payment, Delivery, Order Status, Stock Level.

  <applicationworkload>
    <apptitle>New-Order</apptitle>
    <kreqhour01>0</kreqhour01>
    <kreqhour02>1</kreqhour02>
    <kreqhour03>2</kreqhour03>
    <kreqhour04>3</kreqhour04>
    <kreqhour05>4</kreqhour05>
    <kreqhour06>5</kreqhour06>
    <kreqhour07>6</kreqhour07>
    <kreqhour08>7</kreqhour08>
    <kreqhour09>8</kreqhour09>
    <kreqhour10>9</kreqhour10>
    <kreqhour11>10</kreqhour11>
    <kreqhour12>11</kreqhour12>
    <kreqhour13>12</kreqhour13>
    <kreqhour14>13</kreqhour14>
    <kreqhour15>14</kreqhour15>
    <kreqhour16>15</kreqhour16>
    <kreqhour17>16</kreqhour17>
    <kreqhour18>17</kreqhour18>
    <kreqhour19>18</kreqhour19>
    <kreqhour20>19</kreqhour20>
    <kreqhour21>20</kreqhour21>
    <kreqhour22>21</kreqhour22>
    <kreqhour23>22</kreqhour23>
    <kreqhour24>23</kreqhour24>
    <requestsmsgbytes>100</requestsmsgbytes>
    <responsemsgbytes>438</responsemsgbytes>
    <gaefrontendjavastmtsk>2</gaefrontendjavastmtsk>
    <cloudbenginesqlstmts>56</cloudbenginesqlstmts>
    <sqlmodifypct>60</sqlmodifypct>
  </applicationworkload>
```
# Report from Simulator

<table>
<thead>
<tr>
<th>I. Simulation-run-title</th>
<th>Run-date-time</th>
<th>Latency-goal</th>
<th>Throughput-goal</th>
<th>App-group</th>
<th>sme</th>
<th>Op-hours</th>
<th>#-users</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST for sim</td>
<td>2014-03-30 09:03:56</td>
<td>2 secs</td>
<td>Max tpm</td>
<td>TPCC-Benchmark</td>
<td>yyyy</td>
<td>24</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>App-txn-title</th>
<th>SIM-MINUTES</th>
<th>SIM-AVG-LATENCY-SECS</th>
<th>SIM-THROUGHPUT-PER-MIN</th>
<th>SIM-TXN-COUNT</th>
<th>Txn-workload-%</th>
</tr>
</thead>
<tbody>
<tr>
<td>New-Order</td>
<td>333.32</td>
<td>0.00</td>
<td>11.82</td>
<td>3939</td>
<td>45</td>
</tr>
<tr>
<td>Payment</td>
<td>333.32</td>
<td>0.00</td>
<td>13.89</td>
<td>4628</td>
<td>43</td>
</tr>
<tr>
<td>Delivery</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Order-Status</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Stock-Level</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>II. Simulation-forecast-of-GAE-frontend-variable-resource-usage</th>
<th>USED</th>
<th>CHARGE($) :</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Daily F4_1G 2 instances [F1 billing @ 2 (F4_1G) * 6 (power) * 24 (hours/day) ea $ 0.08/ Hour]</td>
<td>288</td>
<td>23.04</td>
</tr>
<tr>
<td>2. Daily Bandwidth Out average Gigabytes[$ 0.12/Gigabyte]</td>
<td>0.052</td>
<td>0.01</td>
</tr>
<tr>
<td>3. 30-day Month Total Estimate</td>
<td></td>
<td>691.39</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>III. Simulation-forecast-of-Datastore-variable-resource-usage</th>
<th>USED</th>
<th>CHARGE($) :</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Daily Datastore Writes $ 0.9 per million operations]</td>
<td>3.77</td>
<td>3.40</td>
</tr>
<tr>
<td>2. Daily Datastore Reads $ 0.6 per million operations]</td>
<td>0.77</td>
<td>0.46</td>
</tr>
<tr>
<td>3. Daily Datastore Storage $0.006 per GB per day]</td>
<td>1390.00</td>
<td>8.34</td>
</tr>
<tr>
<td>4. 30-day Month Total Estimate</td>
<td></td>
<td>365.99</td>
</tr>
</tbody>
</table>
Forecaster Interarrival Algorithm

Google Cloud Project Simulation Forecaster Mean Interarrival Algorithm

Example

Transaction definition for the “New Order” application workload contained in XML, includes: workload mix (45%), request keying time (18 seconds) and response think time (12 seconds)

Mean interarrival seconds for 640 users
Forecaster Database Algorithm

Google Cloud Project Simulation Forecaster Mean Database Service Time Algorithm Example

"New Order" application workload combined with cloud capacity, defined in XML, includes: mean number of cloud database read operations (23), mean number of SQL seek operations per read (4), mean number of cloud database write operations (24), mean number of SQL seek operations per write (6) and the mean seek time (10 milliseconds)
Baselining Simulator

Google Cloud Project Benchmark Performance Results versus Simulation Forecaster Performance Results

Benchmark performance data points and SimPY simulation data points (tpmC and Number of Concurrent Users) similar.

Additional SimPY simulation data points (tpmC and Number of Concurrent Users) must be implemented to demonstrate sensitivity to “server timeouts” and “100 user connection limit”
Simulator Use

• The calibrated simulator used for future cost and performance estimation.
• If threshold parameters for cloud change then simulator will be re-calibrated
• Results from simulator can be used with high degree of confidence
• Directly based on benchmarking data used to generate simulator parameters
Conclusion

• Performance and cost are important parameters before migrating to cloud
• Knowing which configuration provides best performance/cost tradeoff will help practitioners
• Using TPCC benchmarks against GAE gives parameters for SimPY simulator model
• CloudSim simulator model provides performance and cost estimates
• NFR Approach provides ability to choose optimum configuration based on tradeoff analysis
References

