Towards a Bell-Curve Calculus and its Application to e-Science

Author Lin Yang

Supervisors A. Bundy; D. Berry; S. Huczynska; C. Hughes

Quality of Service (QoS) properties such as accuracy, reliability and runtime, all have a range of possible values. Bell curves (normal distributions) are one way of modeling these ranges. They give more information than error bounds, as error bounds only give a worst-case analysis, whereas bell curves give an indication of the probability of each value occurring.

Introduction

We define a bell-curve calculus for reasoning about QoS properties. The advantages of using bell-curve calculus are:

1. Since there are only two parameters in the expression of a bell curve, it is easy to store and propagate;
2. Bell curves can deal with complex workflows efficiently;
3. According to Central Limit Theorem and experimental result from DIGS*, bell curves commonly occur in the real world.

Definition

Three QoS properties are investigated in our project – accuracy, reliability and runtime, with four ways of combining Grid services:

1. Sequential
2. Parallel_All
3. Parallel_First
4. Conditional

So there are 3x4=12 simple situations:

<table>
<thead>
<tr>
<th></th>
<th>Seq</th>
<th>Para_All</th>
<th>Para_Fir</th>
<th>Cond</th>
</tr>
</thead>
<tbody>
<tr>
<td>run time</td>
<td>sum</td>
<td>max</td>
<td>min</td>
<td>cond1</td>
</tr>
<tr>
<td>accuracy</td>
<td>mult</td>
<td>combine1</td>
<td>values1</td>
<td>cond2</td>
</tr>
<tr>
<td>reliability</td>
<td>mult</td>
<td>combine2</td>
<td>values1</td>
<td>cond3</td>
</tr>
</tbody>
</table>

Table 1: Operations Of 12 Simple Situations

Current Work

1. Define \( bc(\mu, \sigma) \) as the bell-curve approximation of the combined curve \( f(k_1 \cdot \mu_1, \sigma_1) \cdot f(k_2 \cdot \mu_2, \sigma_2) \). For each 12 functions defined above, induce the 24 function for \( \mu_1, \mu_2, \sigma_1, \sigma_2 \). e.g. for runtime in sequential structure:

\[
\mu = \mu_1 + \mu_2 \quad \sigma = \sigma_1 + \sigma_2
\]

2. Test these functions in Agrajag** and determine the ranges of acceptable error.

* A framework developed by DIGS to express some basic distribution functions and define operations on them using Perl and C languages.

Future Work

1. Continue defining and testing systematically in Agrajag
2. Solve current problems
   1. aid theoretical analysis;
   2. to import an additional parameter to describe how good;
   3. to update our calculus to other form, e.g. log-normal.
3. Do evaluation on the Bell-Curve calculus, for example, comparing with other methods by recording CPU time
4. Apply the Bell-Curve calculus to use cases
5. Find support on infrastructure/framework

Figure 1. A Standard Bell Curve

Figure 2. Sum Of Two Runtime Bell Curves

Figure 3. Maximum Of Two Runtime Bell Curves

Figure 4. Maximum Of Two Bell Curves (Method2)

Figure 5. Comparison Of Max Method1 and Max Method2

Figure 6. Comparison Of Max Method2 With Different Resolution Values

Figure 7. Short-term Work Plan Of QoS Properties Investigation

Oriented Architectures.