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

Author Lin Yang

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

Background

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.

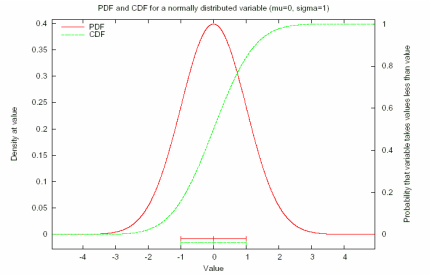
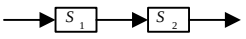


Figure 1. A Standard Bell Curve

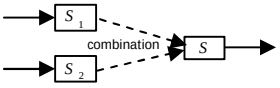
Definition

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

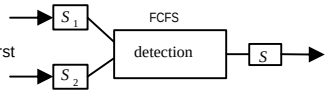
(1) Sequential



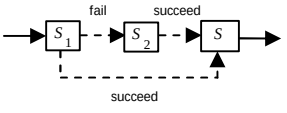
(2) Parallel_All



(3) Parallel_First



(4) Conditional



So there are 3x4=12 simple situations:

	Seq	Para_All	Para_Fir	Cond
run time	sum	max	min	cond1
accuracy	mult	combine1	varies?	cond2
reliability	mult	combine2	varies?	cond3

Table 1. Operations Of 12 Simple Situations

Current Work

1. Define $bc(m_m, s_m)$ as the bell-curve approximation of the combined curve $F(bc(m_r, s_r), bc(m_g, s_g))$. For each 12 functions defined above, induce the 24 function for m_m and s_m in terms of m_r, m_g, s_r and s_g.

e.g. for runtime in sequential structure:

$$m_m = m_r + m_g \qquad s_m = \sqrt{s_r^2 + s_g^2}$$

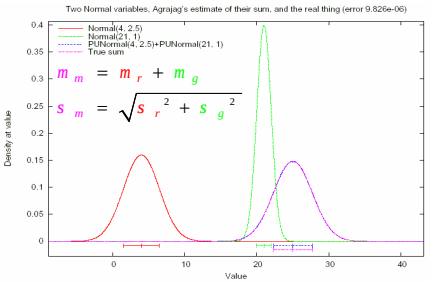


Figure 2. Sum Of Two Runtime Bell Curves

e.g. for runtime in parallel_all structure:

$$m_m = \max(m_r, m_g) \qquad s_m = \max(s_r, s_g)$$

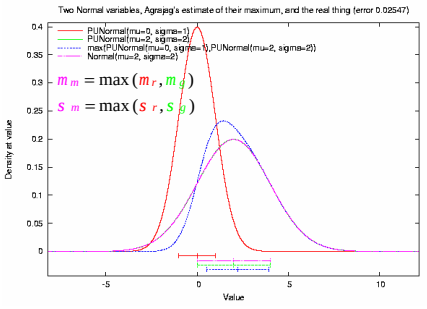


Figure 3. Maximum Of Two Runtime Bell Curves

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

e.g. for runtime in parallel_all structure using Combination Method2:

$$m_m = m_r + m_g \qquad s_m = s_r \text{ or } s_g \text{ (with bigger } m)$$

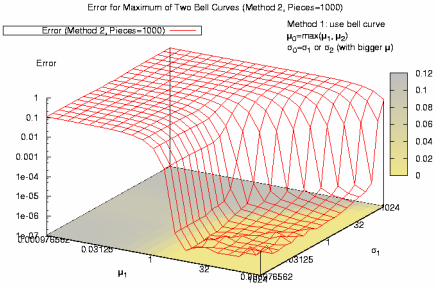


Figure 4. Maximum Of Two Bell Curves (Method2)

For a simple situation, several combination methods may be defined and tested to achieve the best results – the way to evaluate the results we use currently is to get the average error.

For instance, in the case of getting the maximum of two bell curves, among 12 combination methods we defined, Method1 and Method2 are the two best combination methods. We also made a comparison of the two methods.

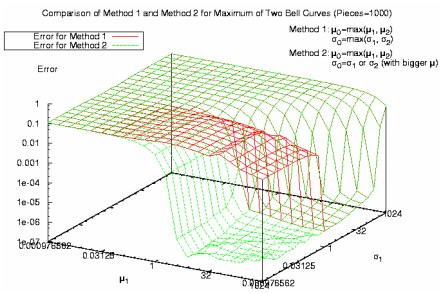


Figure 5. Comparison Of Max Method1 and Max Method2

During the investigation, we encountered some problems: in some areas, when we raised the resolution of our approximation, the values of errors remained almost unchanged. It means that in these areas, we did not find a proper combination method.

e.g. using Combination Method2 to get the maximum of two bell curves (resolution value = 10/100/1000)

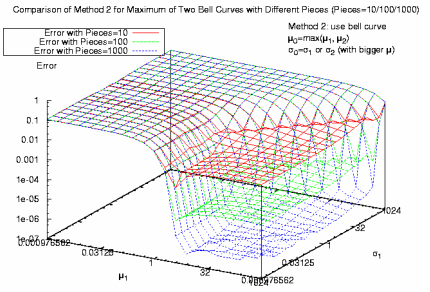


Figure 6. Comparison Of Max Method2 With Different Resolution Values

Future Work

1. Continue defining and testing systematically in Agrajag

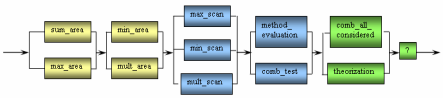


Figure 7. Short-term Work Plan Of QoS Properties Investigation

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

* *Dependability Infrastructure for Grid Services* is an EPSRC project to deal with fault-tolerance and consider wide Quality of Service issues in Service-Oriented Architectures.

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

School of informatics