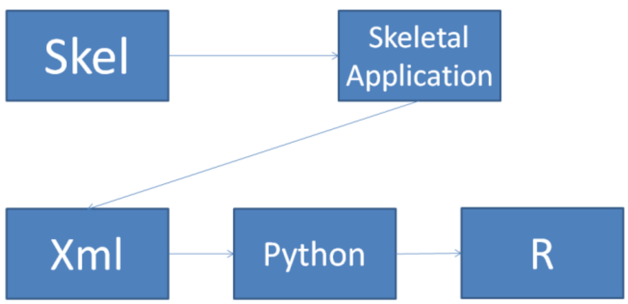
Graphical Analysis of I/O in High Performance Computing

Denver Coker
University of North Georgia
University of Tennessee
Oak Ridge National Laboratory
(706) 483-3980
dccoke3156@northgeorgia.edu

Abstract

In this paper, we describe methods of analysis used to understand and tune I/O in high performance computing. These methods help the user deal with large amounts of data and analyze it with some automation. While we apply these methods to I/O tuning, they are applicable to other settings in performance tuning.



1. Introduction

A common bottleneck in supercomputer applications is poor performance of I/O routines. The purpose of this work is to find ways to understand, refine, and ultimately automate tuning of I/O routines. This is achieved is through the use of statistical models and graphics. There are two kinds of variables used in these models, factor variables and response variables. The factor variables are parameters that control each run of an experiment, such as the I/O method, number of cores, and the machine selected. Response variables are the measured responses, such as the times of various I/O operations. The measured responses are obtained through the I/O framework, ADIOS [4].

2. Skeletal Application

Instead of running a full simulation, a skeletal application mimics the simulation by using insignificant application parameters, running the simulation, and observing the I/O performance only. When the simulation parameters are created the structure of the data is a focus and is preserved. This means that if the full simulation uses five 3D arrays and two scalars for example, that the skeletal application will also have five 3D arrays and two scalars. This keeps the integrity of the simulation intact.

3. Importing the Data

The data from the skeletal application is in xml format. In most cases there are hundreds of xml files, each of which contains different parameter combinations such as number of cores used, I/O method, or what machine it was executed on. To make these files more accessible to analysis with R they are converted to simpler text files with a python script. The newly created text files are read into R and are used to create one large data frame on which all of the analysis can be easily carried out.

4. Graphical Analysis and Automation

The data frame is a form that makes many analytical and graphical methods in R accessible. While this is convenient, still there are potentially too many parameter and response combinations to graph. We use analysis of variance (ANOVA) to select and order the factor and response graphs.

4.1 ANOVA

All factors are examined for each response with a "main-effects and two-way interactions" ANOVA model. Using R's "formula" method for two factors, this model is:

Response ~ Factor A + Factor B + Factor A * Factor B

A factor is worth plotting for a given response if it is "significant" in the ANOVA model. Significant means that the factor levels have a large enough difference above random fluctuations. We order the (jittered) scatter plots of a factor versus a response according to decreasing significance level. See Figure 1 for an example prepared with the ggplot2 package[3].

Similarly, an interaction is worth plotting for a given response if it is "significant" in the ANOVA model. Significant here means that the response to levels of one factor is not the same for at lease one level of the other factor and the difference is above random fluctuations. We order the (jittered) scatter plots of a factor versus a response, colored with the other factor, according to decreasing order of significance. See Figure 2 for an example.

5. Conclusion

We develop an automated way to produce graphical analysis output for tuning I/O performance. We use skel [1, 2] to create a skeletal application that runs much faster than the full simulation without sacrificing loss of relevant data. It takes the output and automatically creates a data frame within R for analysis. The analysis takes all combinations of variables and shows only the significant pairs in an easy to understand scatter plot, starting with the most significant factors and their interactions. This method is much quicker and lends more light to the understanding of the I/O output. With this tool, refinement, understanding and eventually the automation of I/O tuning is possible.

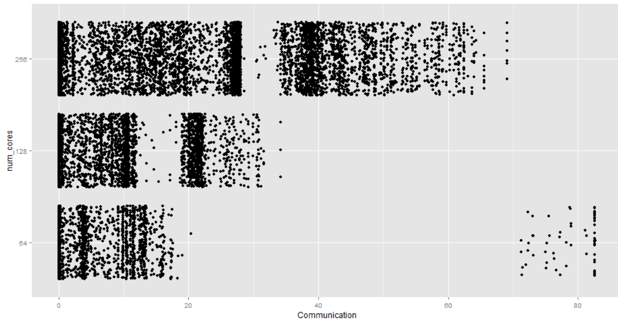


Figure 1. Single factor versus response jittered scatter plot.

Figure 2. A two factor versus response jittered scatter plot.

6. Acknowledgements

My thanks to Oak Ridge National Laboratory, RDAV, Scientific Data Group, University of North Georgia, University of Tennessee, Dr. Scott Klasky, Dr. Jeremy Logan, and Dr. George Ostrouchov.

7. REFERENCES

[1] Jeremy Logan, Scott Klasky, Jay Lofstead, Hasan Abbasi, Stephane Ethier, Ray Grout, Seung-Hoe Ku, Qing Liu, Xiaosong Ma, Manish Parashar, Norbert Podhorszki, Karsten Schwan, Matthew Wolf. 2011. Reasoning about naming systems. *Skel: Generative Software for Producing Skeletal I/O Applications* (Dec. 2011), 191-198.

[2] Jeremy Logan, Scott Klasky, Hasan Abbasi, Qing Liu, George Ostrouchov, Manish Parashar, Norbert Podhorszki,Yuan Tian, Matthew Wolf. 2012. *Understanding I/O Performance Using I/O Skeletal Applications*. 77-88

[3] Hadley Wickman. 2009. ggplot2: Elegant Graphics for Data Analysis (Use R!), 1st ed. 2009. Corr. 3rd printing 2010,

[4] S. Hodson, S. Klasky, Q. Liu, J. Lofstead, N. Podhorszki, F. Zheng, M. Wolf. 2009. *ADIOS 1.0 User's Manual*. http://users.nccs.gov/~pnorbert/ADIOS-UsersManual-1.0.pdf