Partnership for International Research and Education
A Global Living Laboratory for Cyberinfrastructure Application Enablement

Application Profiling and Prediction in the Grid Environment

Marlon Bright, Undergraduate; Javier Delgado, Graduate; Florida International University
Advisors: S. Masoud Sadjadi, Ph.D.; Florida International University (FIU)
Rosa Badia, Ph.D.; The Barcelona Supercomputing Center (BSC)

National Science Foundation

I. Research Overview and Outcome

A. Motivation

Grid Enablement of Weather Research and Forecasting Code (WRF)

▪Weather prediction can save lives and help business owners and emergency responders in the case of inclement weather.

▪The goal of WRF is to improve weather prediction, especially in the case of hurricane mitigation. Optimum forecasting consists of:

- Accurate and Timely Results
- Precise Location Information

▪***Grid-Enablement*** is the practice of taking existing applications that are currently configured to run on a single machine or cluster and adapting them to run on a non-homogeneous resources connected via the internet with the goal of improving its performance (execution time and resource utilization).

▪The expected benefits of successful *grid-enablement* to WRF will be:

- Access to a greater number of compute resources due to the utilization of more domains.
- Faster results, by virtue of having more compute power (**more power = less simulation time**)
- Improved precision of results (**more power = higher domain granularity**)

▪The **meta-scheduler** is the "global" scheduler of the grid environment – above the local schedulers. Its function is to select the best resources for a job to run on by analyzing application and target architecture characteristics to find the best match for optimal resource usage and job return time. **Performance prediction** provides:

- the meta-scheduler with accurate prediction run times - "smarter" scheduling.
- (at local level) improved resource allocation and local scheduling techniques as many users overestimate job execution times for precautionary reasons.

▪Leverage many years of experience from BSC faculty towards these goals and make personal connections that will last throughout our careers.

B. Goals

- Learn to use *MareNostrum,* which is a much larger scale than the clusters we had used before. The test application will be the same one we have been working on, Weather Research and Forecasting (WRF), version 2. We will extend previous *amon/aprof* research (which used up to 16 nodes) to a larger number of nodes (up to 128). The former will test our model's scalability; we will also be testing its ability to work on different architectures (formerly Intel-only, now Power PC), and a different version of WRF (from Version 2.2 to 2.2.1). Being at BSC not only enabled us to use *MareNostrum*, but also gave us the opportunity to work with many faculty and students that use the system regularly and were able to give us quick support. This way, we learned to use the supercomputer very fast and were much more productive while there.

- Become more familiar with similar software developed at UPC, *Dimemas* and *Paraver.* Compare/contrast *aprof* predictions to *Dimemas* predictions in terms of accuracy and prediction computation time. Thanks to PIRE, we could sit in the same room with the developers to achieve this goal.

- Find ways to combine *aprof* and *Dimemas*

- Do all this while at the same time learning about the culture we are in. For Javier, this was also an opportunity to introduce an undergraduate student, Marlon, to the research lifestyle. For Marlon, this trip provided a unique way of learning about research by participating in two programs at once: REU and PIRE.

C. Profiling and Prediction Tools

Amon / aprof

Software tools developed by collaborators at IBM Research, Japan.
▪*Amon* is a monitoring program that runs on each compute node recording new processes
▪*Aprof* – regression analysis program running on head node; receives input from *amon* to make execution time predictions (within cluster & between clusters)

Paraver / Dimemas

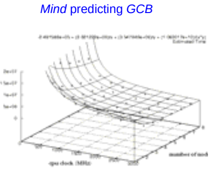
Software tools developed at Barcelona Super Computing Center (BSC), Spain.
▪*Dimemas* - simulation tool for the parametric analysis of the behavior of message-passing applications on a configurable parallel platform.
▪*Paraver* – tool that allows for performance visualization and analysis of trace files generated from actual executions and by Dimemas
▪Trace files for *Paraver* and *Dimemas* generated by mpitrace that is linked into execution code.

D. Previous Research

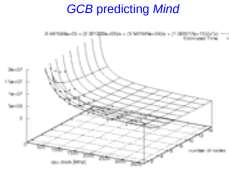
▪Through a *remote*† collaboration with IBM's Tokyo Research Lab, a model was developed for prediction execution time of a WRF simulation, based on statistical learning.
▪Experiments were performed on two clusters at FIU—*Mind* (16 nodes) and *GCB* (8 nodes)
▪Experiments were run to predict for different number of nodes and CPU loads (i.e. 2,3,…,14,15 nodes and 20 – 100 percent CPU utilization)
▪*Aprof* predictions were within 10% error versus actual recorded execution times within *Mind* and *GCB* and between *Mind* and *GCB*
•Results published in the HPGC workshop of IPDPS 2008.

† We emphasize *remote* here since PIRE showed us how much better of a collaboration is possible after physically meeting someone and working with them.

Mind predicting *GCB*



GCB predicting *Mind*



E. Challenges

▪**The high latency of internet connection in comparison to Local Area Network connections** poses problems to requirements of WRF code.

▪**The high volume of WRF's source code**. WRF is comprised of around 165,000 lines of code with another 40,000 lines generated at compile time.

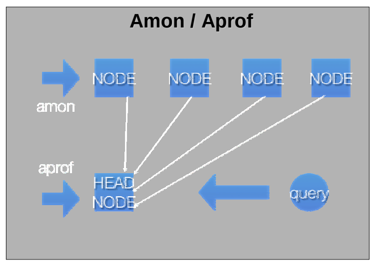
▪**The tediousness of compiling WRF versions on unsupported platforms.** NCAR has supported several platforms, but unfortunately compilation is not straightforward on the platforms used at FIU. Fortunately, WRF was already available in *MareNostrum*, which saved us many hours of work.

▪**Adapting effective processing power limitation tool, *cpulimit*, to the super computer architecture** of the *MareNostrum* super computer (2,547 nodes; 4 Power PC processors per node) at BSC.

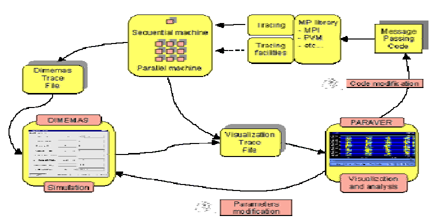
▪**Large size requirements of Paraver trace files.** *Paraver* trace files generated by *MPItrace* tool can be in the range of gigabytes causing space limitations to be a serious issue on some clusters.

▪**Intricacies of the new Infrastructure.** Since *MareNostrum* is a large, production supercomputer, there are restrictions in accessing nodes and executing programs on them, which is a requirement for *amon*.

▪**Working around the Queue**. Having previously worked only on small clusters we had full access to, we took for granted that simulations could be executed whenever. In a production environment, an accurate prediction of queue time is necessary in order to avoid overly long queue times and killed processes.



Paraver / Dimemas



Source: Barcelona SuperComputing Center – http://www.bsc.es/plantillaA.php?cat_id=479

F. Experimental Process

▪Overview
•The same simulations used for testing the model on *GCB* and *Mind* were used on *MareNostrum*, but with a larger number of nodes
•Number of nodes: 8, 16, 32, 64, 96, 128
•CPU Utilizations: 100, 75, 50, 25

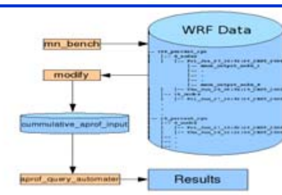
▪Developed a benchmarking script that edits and submits a job file to the *MareNostrum* scheduler

```
For each number-of-nodes and cpu-utilization combination:
    Create job to execute the simulation
    Record run-related data after it has completed
```

▪When all data was available, we developed a script to prepare data for *aprof*
▪Combines *amon* output to one data file
▪Filters processes in data file to solely WRF processes
▪Edits processes to *aprof* friendly format

▪Start *aprof*, load the data file as the initial data

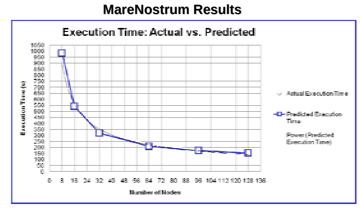
▪Execute *aprof* Query Automation script
▪Starts telnet session querying *aprof* for benchmarked scenarios in data file
▪Compares and tabulates predicted values to actual values

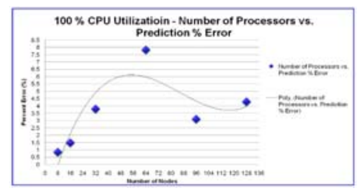


G. Results

After running our experiments on *MareNostrum,* we can now conclude that our model is accurate (i.e. less than 10 percent error in predicted execution time) for:
•Up to 128 nodes on Intel and PPC architectures, when using executions with different number of nodes as input.
•Different architectures (i.e. Intel-based Beowulf clusters and large-scale architectures like *MareNostrum,* which is PowerPC).
•Small (8-16 node) Intel clusters; also accurate with different CPU values

MareNostrum Results





Ongoing Challenges

▪At this time, D*imemas* has proven more effective at deriving the overall speed-up during development by simulating changes in an application and not as effective for the general case of application execution time. D*imemas takes much* longer (upwards of 5 minutes, depending on number of nodes involved) to predict execution time for a standard trace file, compared to aprof. However, the scenario of having D*imemas* generate input data points for *aprof* is currently being researched. This would improve the precision of *aprof's* real time predictions while not requiring an actual application execution to be made, since *aprof's* modeling capacity improves as more input data is given to it.

▪*Aprof* prediction data for CPU usage below 100% have been inhibited by complications in the adaptability to the *MareNostrum* architecture of the CPU limiting software (*cpulimit*) used in the previous research.

H. Lessons Learned

Amon / aprof

▪Pros:
▪Simpler to deploy in comparison to *Paraver/Dimemas*
▪Scalability of model is within target boundary (10%) when full CPU is utilized
▪Feasible solution for performance prediction purposes

▪Cons:
▪*Aprof* requires more base executions for accurate performance in comparison to *Dimemas*

Paraver / Dimemas

▪Pros:
▪More features—could be more useful to experienced user (i.e. adjustment of system characteristics)
▪Visualization and analysis of execution for analysis purposes
▪Graphical User Interface

▪Cons:
▪Requires special compilation of applications
▪Requires non-trivial-to-install kernel patch
▪Large trace files (in our case, gigabytes, for small WRF regions)

I. Future Work

•Short Term
•Determine problem with *cpulimit, so that accurate results can be evaluated with different simulated CPU speeds.*
•We have worked with the developer of cpulimit, our BSC adviser, Dr. Rosa Badia, and the support team of MareNostrum to resolve this issue
• Use *Dimemas* to simulate runs with a large delta-factor in terms of number-of-nodes. For example, 256, 512, and 1024. We hypothesize that using these as input to *aprof* will result in better accuracy, since *aprof* is less robust against non-linearities in scalability.
•Using *MareNostrum* and our existing clusters, test the prediction accuracy of *aprof* for shared memory executions of WRF.
•Our goal is to have this published in the HPGC workshop of IPDPS 2009.

•Long Term
•Apply more parameters to our prediction model (memory, bandwidth, etc.)
•Test the model with different inputs

•Continue collaborating with our new BSC partners

III. Acknowledgement

The material presented in this poster is based upon the work supported by the National Science Foundation under Grant No. OISE-0730065, OCI-0636031, IIS-0552555, and HRD-0833094. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.