Application Level Power Analysis and Power Management

Luiz DeRose, Cray Inc. (ldr@cray.com)

Power consumption is one of the biggest challenges facing exascale systems. While hardware plays a dominant role in this challenge, software can help by monitoring and proactively managing how power is being used in the system. Exascale nodes will potentially have heterogeneous processing elements and will certainly have deep memory hierarchies. The programmer must understand the tradeoffs of using these aspects of the node, especially with regards to the tradeoffs between power and performance. Exascale processors and other components are expected to have the ability to do power limiting and power sloshing, which will allow for more power to be used in certain circuits while reducing the power usage of other circuits. Hardware can attempt to manage the provisioning of power automatically, but this strategy will certainly be sub-optimal, especially since it tends to be a reactive strategy; moving power throughout the system as changes in activity is detected. In order to be efficient, power management must be proactive, which requires operating system and specially runtime support. This position paper proposes research and development to address issues of power measurement, analysis, and management at the runtime level. Certainly some compiler support as well as hardware infrastructure will be required, but they are outside of the scope of this call.

Power aware computing is an area of research that has been growing in the last few years. Several groups have been investigating Power-Aware MPI, in particular with the view of reducing power on idle processors, as well as prediction mechanisms for power based runtime decisions. In addition, several researchers have explored the use of hardware counters for characterizing power properties. However, despite this growing activity in the area, the research community is still in its initial steps to create a power aware computing environment. This is certainly an area that requires significant investment, in order to reach exascale computing without exceeding the power envelop.

Description of Research Direction

One goal of this research would be to devise analysis techniques to be able to provide suggestions of possible transformations or implementations of the application with different performance/power profiles. In order to do application power analysis effectively the research community should investigate "*prediction mechanisms*" to help users understand the implications of power utilization based on selection of algorithms. The ultimate goal is to develop power-aware tools and techniques that can aid the user to properly direct the system's use of power, using transparent automation where possible, and transparently adopting environmental settings to achieve a desired power reduction and energy savings with a predetermined impact to performance. These prediction mechanisms would allow for application developers to examine lower power modes for some of the important kernels of the applications, such that they could get an idea of the resulting performance and power consumption. These prediction mechanisms would also make suggestions of other possible transformations or implementations of the code with different performance/power profiles.

In addition, the research community should investigate approaches to help the programmer make the necessary decisions, using a combination of expected results (based on previous simulations or actual runs of the program), analysis of power consumption and delivered performance, and a consideration of the various ways of mapping the application to the available components. Ultimately these power prediction tools would be able to answer questions such as "What performance for power, in units such as flops/joule, will my program (or function) achieve if I use this combination of resources?"

Another goal of this research would be the implementation of adaptive power management at the application level. Software support for this adaptive power management should be present at many levels of the software stack. In the compiler, hints should be generated to indicate which processing, storage, and communication components were

about to be utilized more or less heavily. While some of these hints could be generated automatically, ultimately the programmer will have the best knowledge of what resources their program flow requires and when, and what the options are for how to perform the various required computations. The runtime would be responsible for collecting various pieces of performance and power consumption data. It would also act on the hints and directions given by the compiler and programmer, and using the knowledge obtained in the power analysis it would notify the operating system, and consequently the hardware, about the intended uses of hardware components and any expected changes in component utilization during the run, allowing power to be directed proactively to where it will be needed.

Challenges Addressed

Several challenges should be addressed to achieve effective application power analysis and adaptive power management. One of the main challenges would be to identify and correctly attribute the power weights to the individual "operation" components and map back to the application. Automatic adaptation or the ability to effectively change the application during runtime is another major challenge to be addressed. This research would address these challenges with prediction mechanisms based on the combination of micro-benchmarks and simulation, as well as the use of adaptation and analysis techniques based on the knowledge acquired with auto-tuning and automatic performance tools research.

Maturity

The use of micro-benchmarks to create a "power profile" of the different "operation" components, and the use of a simulator to break down the kernels into the multiple components appears to be a reasonable direction to define the weights required for power analysis. A significant amount of research has been done to advance the state of the art in automatic performance analysis. Some of the results of this research could be applied for power analysis.

Similarly, the use of compiler technology to provide input to the runtime, as well as adaptation and auto-tuning technology, together with power analysis can certainly be applied for adaptive power management.

Uniqueness

Power aware computing is not necessarily unique, as it has been in use in several other areas, as for example mobile computing. However, the tradeoffs between lower energy consumption and maximum performance are certainly unique to HPC.

Novelty

The ultimate goal of this research would be to develop power analysis and power management techniques to aid the application developers and user to properly direct the system's use of power to achieve a desired power reduction and energy savings with a limited impact to performance. This is certainly a novel approach in HPC.

Applicability

This research activity has common grounds to other areas, such as automatic performance analysis, performance prediction, auto-tuning, and application adaptation targeting performance. Thus, some of the results from this research could be applied to those areas, in the same way that some of the results from research on those areas would be investigated, targeting application on power analysis and power management.

Effort

The work described here requires substantial research and development in simulation infrastructure, analysis tools, and runtime. The effort would require approximately 5 FTEs for at least three years. This estimate does not include compiler and programming models work, which the scope is outside of this call.