2 Summary: Dynamic Modeling

The second major focus area addressed by the 2013 MODSIM workshop was that of Dynamic Modeling. The essence of Dynamic Modeling is the applied use of MODSIM technologies during the production execution of extreme scale systems; that can be used to guide and optimize system and application execution for performance, for power, and/or for reliability. Dynamic Modeling, in comparison to the MODSIM methodology focus area, was represented by far fewer presentations at the 2013 workshop reflecting the lower number of actual position papers submitted in advance. This also reflects the lower degree of research that is currently in progress and its lack of maturity but does not diminish the importance of this area as we see from the presentations and comments that were made during the workshop. There were only two sessions dedicated to dynamic modeling containing a total of seven position papers. Each session provoked active discussion and interaction amongst the participants.

There was a general recognition that Dynamic Modeling is a key area that will become more prominent as the complexities of extreme-scale computing increase, in terms of both systems and applications. Both software and hardware aspects are expected to significantly increase in their dynamic behavior as resources dynamically change during the execution of an application which itself will have temporally varying requirements. Thus, the application of Dynamic Modeling, that is able to provide predictive information on the mapping between the instantaneous application requirements and the resources available within a certain future time-window will enable a whole range of possible optimization opportunities. But at the same time pose many significant challenges that need to be addressed.

We detail below the discussion points made by the speakers in both sessions using the questions that were supplied in advance to the presenters: the current major research contributions made to date; the existing gaps in the research; the need and potential of cross pollination; and our recommendations and path forward for Dynamic Modeling. In this, we also supplement the points made by the presenters with points that were raised during active discussions during the sessions.

2.1 Major Research Contributions

The contributions of current research as presented in the two sessions were focused on the following five aspects:

Methods and Techniques: It was recognized by many that Dynamic Modeling requires rapid prediction methods if they are to be useful within a dynamic system, and that they need to minimize the perturbation of system execution. As such, methods that had analytical models at their core were deemed more appropriate than those requiring detailed simulation based approaches. White-box based approaches, using integrated application and software information that was passed through the software stack, as well as black-box based approaches, based on models formed from empirical data, were discussed. An example black-bock approach that was presented was centered on the use of Machine Learning techniques, which has the potential of providing an adjustable prediction mechanism to assist in a runtime system as well as for auto-tuning. It was claimed that Machine Learning can improve overall prediction accuracy compared with other black-box approaches based on heuristics or simple regression.

Model generation: An important aspect presented was in the formation of Dynamic Models. An example of this was with the early work on a tool based approach (PALM) to semi-automate analytical model generation. PALM is based on the use of annotations at application level to construct a static description (model) that is then populated using the dynamic behavior of the application from run-time measurements. The current implementation of the tool outputs python scripts to describe a model that is focused on performance aspects but could be put into an executable form for rapid execution.

Interfaces to Dynamic Models: The definition of interfaces was also discussed, specifically interfaces that are able to express domain and software-specific knowledge to the Run-time-system. In a current research prototype, layers in the execution stack can export state information, and allow for a wealth of information to be collected. This can be then be used to dynamically reconfigure and adapt systems towards a better execution state.

Integration into run-time systems: The use of dynamic models was discussed in the context of an experimental self-aware/self-adaptive system that currently uses Dynamic Models based on a black-box approach using empirical data (counters) at node level. This also is able to analyze deviations from past history to identify possible performance issues.

Integration into programing and execution models: In one presentation, current research on the over-decomposition and object migration for driving dynamic load-balancing and implementation of fault tolerance techniques within experimental as well as production level programming models was discussed. Such approaches currently lack predictive knowledge of the expected application behaviors that could be provided using Dynamic Models that could be actionable in terms of guiding performance and power optimizations during execution.

2.2 Gaps

The discussion of current gaps in Dynamic Modeling focused on the following aspects:

Methods and Techniques: The appropriateness of current MODSIM techniques for Dynamic Modeling is not particularly well explored though the need for rapid predictive capabilities was generally recognized. Addition aspects including the degree of accuracy, the longevity of any prediction in terms of time (since the system and application are temporally changing), and the spatial scope for any prediction across system resources, is not well understood. Specific points raised included:

- the need for a common way in which a Dynamic Model could be implemented, so as to cover multiple MODSIM methodologies
- the need for rapid predictive capabilities
- the need to explore appropriateness of different MODSIM techniques for Dynamic Modeling
- the potential and mapping of Machine Learning techniques to the complexities and intricacies of advanced architectures.

Model generation: There is a need to generate Dynamic Models with as little user intervention as possible. In essence such model generation will elevate a plethora of possible empirical observations to predictive capabilities encapsulated within a dynamic model. This is also coupled with the need to identify and represent the critical path of an application, as well as being able to automatically distinguish between representative and unexpected measurements. Models also need to include coverage of applications that vary temporally in terms of their resource requirements.

Integration of concerns: There is the need to incorporate all of performance, power, and reliability concerns within Dynamic Models. Current exemplars of Dynamic Modeling are almost all performance based but the inclusion of other factors will considerably increase their utility, and assist in not just performance optimization activities but also with optimization of power use and to assist in system fault tolerance aspects. But their inclusion pose significant research challenges.

Integration of Dynamic modeling: The incorporation of Dynamic Models into run-time systems need to be explored. This includes by what means models may become actionable and assist in the dynamic decision making mechanisms within run-time systems. In addition, hardware and software layers are mostly compartmentalized in terms of what information is shared, and what reconfiguration knobs are available. This necessitates the need for information exposition interfaces, as well as mechanisms that need to be supported by the hardware and run-time that allow for optimization strategies that simultaneously consider multiple competing objectives.

Use of actionable models: Actionable models can be used to guide the dynamic optimization of task-based applications for performance, such as in determining load-balancing across system resources, minimization of data movement, and minimizing power use. Models could be expressed through programming model constructs, fed by application information, and used dynamically within an execution model and run-time system. Mechanisms for determining Quality of

Service requirements in terms of power/energy, performance, and/or resilience need to be explored so that they can be expressed by the programmer within suitable programming models.

2.3 Cross Pollination

Until recently, the availability of funding advanced run-time systems has been difficult, and has caused the development of, and interface to, Dynamic Modeling techniques very difficult. Recent DOE FOAs have resulted in funding for advanced run-time systems but Dynamic Modeling is seen as a key enabling and cross-cutting technology which has wide-spread applicability. Dynamic Modeling methodologies can be thought of as an extension of static modeling and simulation. The techniques in Dynamic Modeling are iterations of previous “lessons learned” from these areas with research focused on trading calculation speed with accuracy. Each area has the potential to cross-pollinate the other.

To feed this cross-pollination a need for standardization of input/outputs that enable Dynamic Modeling is essential. This standardization will also promote robust model development in all areas. To aid in this cross-pollination, techniques for Dynamic Modeling should be not be application or domain specific and thus should be designed be applicable across all applications and systems.

Dynamic Modeling was identified as being a necessity in the area of self-aware and/or self-adaptive system. These key concepts were recognized as critical as we move towards Exascale computations. Self-aware and self-adaptive system designs deal with energy efficiency, performance, and reliability issues in concert in order to adapt to a complex environment in order to complete the computation. This requires research projects in multiple disciplines (processor and memory design, Operating Systems, compiler and so on), all aimed at developing self-timing or self-adapting systems.

In a similar area of self-adapting is in the area of system software (operation and runtime systems). This software is often designed to make decisions about power and performance to adapt the system. These decisions would benefit greatly from Dynamic Modeling techniques to feed these decisions. Thus the interaction and integration of Dynamic Modeling with research in system software (operating and runtime systems), is a large area ripe for cross pollination.

Some of these cross-pollinating ideas already exist in ASCR funded projects. There are interactions in projects that examine the identification of performance faults in large-scale systems, those that are developing tools and techniques for performance and power modeling, and those that are developing new techniques for OS and runtime systems.

2.4 Recommendations and Path Forward

Dynamic Modeling is a key enabling technology that is cross cutting across multiple research areas and activities. It is expected to become the heart of self-adaptive/self-aware systems, a key feature expected to be critical for future Exascale systems. Current research in Dynamic Modeling is at its infancy but in demand as the challenges of performance, power, and resiliency merge into a complex multi-objective optimization essential to future extreme-scale computing.

The gaps identified above would benefit from targeted funding from all agencies concerned. Dynamic Modeling should be considered as a research topic in its own right given its potential in guiding the optimization of future systems for all of performance, power and reliability. However, interaction among other research areas will be critical if Dynamic Modeling is to live up to its potential.