1 Summary: Simulation and Modeling Methodologies

The workshop presentations and discussions were organized around a set of questions designed to move the community forward in identifying high impact advancement opportunities in the challenge areas articulated during the MODSIM 2012 workshop. Broadly these questions were categorized into i) major research contributions, ii) gaps, and iii) opportunities for leverage and cross-pollination across MODSIM sub-disciplines and efforts. In particular, item iii) was intended to help focus future efforts by eliding duplication of effort across the community as well as accelerating collective progress. Accordingly this summary is organized according to what we have learned in these categories. The following summary covers sessions on Modeling Methods, Emulation, Empirical Methods, and Simulation. It also includes input from the industry panel.

1.1 Major Research Contributions

The major foci presented in the individual sessions addressed the following major research goals and associated motivations.

Emulation & Empirical Methods: Several presentations stressed the importance of empirical methods as a basis for the construction and validation of models as well as an integral component of the iterative optimization of applications. Applications themselves represent a complex integration of different behaviors (compute, memory, control flow, etc.) and thus their performance is a complex function of many distinct characteristics. Empirical methods can provided focused extraction of behaviors. In particular if the empirical methods were sufficiently general they could be used for cross-validation of models across multiple applications. Such cross-validation will also serve the purpose of better understanding the important properties and behaviors of these complex integrated applications, which in turn should lead to better models. Finally, empirical methods can be used to characterize application behaviors by focusing on properties that cover a span of applications and architectures. Thus, these methods are a prelude and input to model construction processes across a greater span of applications and architectures.

Modeling Abstractions: Several presentations focused closely on the importance of recognizing the multiple levels of abstraction that lay between applications and devices. In developing modeling and simulation techniques it became clear that it is necessary to consider the relationship between layers and how the management of system properties, such as power and reliability, can be mapped to these layers. For example, resilience mechanisms and power management can happen at one or more abstraction layers, e.g., run-time, application, and hardware. To accurately model future system behaviors we must understand and capture how these properties can map to these abstraction layers and how they may interact across layers. Alternatively, it was also recognized that one could advance the need for new models that span multiple layers. For example, the use of computational idioms was advanced as concept for linking application level behaviors to hardware architectures and systems. The key observation in this case was that certain computational patterns have identifiable consequences for the implementations that survived traversal of, and translation across, the multiple layers of abstraction between the application and low-level hardware. Such abstractions are powerful basis for the constructions of models.

Modeling Accuracy: More than one session addressed the issue of model and simulator accuracy and emphasized the need to develop metrics and methodologies to measure accuracy (more in Section 1.2). In particular, discussions advanced the notion that predicting trends and relative merits of alternative approaches was more important than absolute accuracy. This is due in part to the proprietary nature of some types of information (e.g., device properties) and partly due to the speculative nature of many evaluations, e.g., future technology nodes where devices have not been developed yet and whose exact physics may be unknown but for which some models may exist.

1.2 Gaps

Much of the discussion of gaps in current research areas identified in MODSIM 2012 naturally fell in to the following categories.

Modeling Accuracy: A significant concern is the accuracy of simulators. Even if a simulator is validated, the validation is dependent on the particular simulator configuration. This observation raises the following challenges.

- Can we quantify this variation of accuracy across simulator configurations or even across simulators? Can we rely on rigorous methodologies such as uncertainty quantification to assist in the development of characterizations of accuracy?
- In the same vein can we characterize and quantify the accuracy vs. performance variation across simulators? This is poorly done in current simulators (if at all).
- A specific consideration in this regard is that trend behaviors are often more important than absolute values and in fact with regard to evaluation at future technologies reliable, correct, absolute values are simply infeasible. Rigorous statistical techniques may find foundational value here.
- We note that these problems are amplified by the fact that functional needs often go beyond a specific simulator. For example, exploiting locality in speeding up parallel simulation is key to simulator speed. Simulator speed can in turn be improved by sacrificing accuracy. A second example, trace-driven simulators are not useful for studying data dependent (e.g., power) behaviors. This simulator capabilities have to be matched the questions being asked.

Integrated Models: As systems become more diverse in their architecture, their physics (e.g., optics vs. electrical), and their packaging (2D, 2.5D, 3D), systems will have to deal with complex interactions between system components and the physics. This effect is amplified, as power and energy have become first class metrics to be optimized. Such integrated models create new modeling and simulation challenges. Some major challenges that were identified are the following.

- For extreme-scale architectures such as geographically distributed architectures, we need end-to-end models that integrate data flows with models of the architectures and applications at the end points.
- It is critical to be able to integrate physical models for energy, reliability, thermal, and packaging with architecture, system, and application models. The complexity of models is now considerably greater. Individual combinations are also necessary, e.g., application level power and performance, or resilience and power in concert, etc.
- Hardware diversity (e.g., heterogeneity or asymmetry) makes the modeling and simulation problem more complex. Ideally, models should be able to capture fundamental computational aspects without the need for the diversity of hardware to translate into commensurate increase in the complexity of the models.
- Models must enable resilience, power, and performance tradeoffs.
- Model correlation: When including models for distinct physical phenomena we have to make sure that the models are compatible. For example, one cannot arbitrarily couple application level power models with just any hardware-level reliability model.
- Hybrid Models: This relates to the execution of mixed models such as (analytic + discrete-event) over large time scales. For example, one presentation explored the integration of analytic workload models expressed in language form (Aspen) with massive parallel network simulation.
- How can we link physical phenomena to their impact on the application? For example, evaluating application resilience in the presence of transient soft errors. This requires traceability from device level events to algorithmic behaviors. Doing so at scale becomes exceedingly challenging. How can we estimate error in such models?
- Multi-scale Models: When physical behaviors are modeled in conjunction with application behaviors, events can take place at multiple time scales. For example, power can vary at microseconds while temperature can vary at millisecond intervals while transient errors and device degradation can occur at much longer time scales. How do we develop such multi-scale models.
- Modeling of near threshold voltage operation for low power must also capture the increased sensitivity to noise and the occurrence of transient errors. Modeling the impact at the application level requires integrated models that link physical operation to the application.

Application-Driven Modeling: Modern simulators are largely constructed bottom-up – reflecting an architecture or hardware perspective. Future systems will have to reflect recognition of application needs. In particular application developers reflect an important constituency of simulator users. To enable productive application-architecture co-design and co-optimization, simulators must reflect the needs of this constituency. This perspective cedes the following characteristics and challenges for the development of the next generation of simulators.

- Simulators and their interfaces should also be designed from the perspective of the applications. Such designs may be quite different from the structure and operation of modern event driven simulators, for example in the design of the interface and the type and manner of exposure of simulator “knobs” for exercising different configurations and experiments. Today’s simulators are generally not friendly to application developers.
- How does one expose novel architecture features to the application? How can we specify and map applications to simulators? User level tools for mapping?
- How well do proxy applications (or benchmarks for that matter) represent full applications and how can experiments with these proxies be scaled to represent the expected behaviors of full applications?
- How can we compare large applications across multiple and diverse hardware platforms as represented by their respective simulation models?
- There is a need for flexible application descriptions to drive simulations. These descriptions may span multiple levels of fidelity and scale. For example instruction level (micro-scale) to analytic (macro-scale) models.
- There is currently a big gap between application-level needs for resilient operation and resilience mechanisms implemented in hardware. How can we explore the bridging of this gap via models and simulations?

Model Generation: The diversity of hardware, the emergence of new classes of applications (e.g., graph processing), and the increasing impact of device physics is challenging the model construction process. In discussing how to attack this challenge several themes emerged and are summarized below.

- There is a need for tools for measurement and diagnosis of i) opportunities, ii) bottlenecks, and iii) complex interactions between applications.
- There is a need for automatic and semi-automatic model generation methodologies and tools. For example, the near term need is to generate models from measurement data.
- Standardized techniques for storage of data, analysis, and characterizations and access to these via standardized interfaces. This data can be used for generating and validating models.
- We need a set of benchmarks with high coverage over the phenomena of interest.
- Cost models for execution on emerging architectures that exhibit a high degree of hardware diversity
- Resource contention models that scale to extreme scale systems.
- Power models and associated simulators. In particular, there is a need for high-level power models that can be used by application developers to understand the energy and power consequences of algorithm and data structure decisions.

In addressing the technology gaps, a companion question that was explored was how as a community we could leverage each other's research. Such opportunities for cross-pollination are described in the following section.

1.3 Cross Pollination

All presenters were asked to consider sources of leverage and cross-pollination across MODSIM areas and efforts. This section summarizes key points from their recommendations and the end-of-session discussions.

Interoperability: As motivating examples, we should strive to learn from the System-on-Chip (SoC) and climate modeling communities. Both communities support an ecosystem of distinct modules and models that are designed to be interoperable and where interoperability grew out of necessity. The consensus was that the MODSIM community is on the cusp of a similar need.

Introspection: Introspective capability, e.g., instrumentation, is a basic technology with tremendous impact across the MODSIM spectrum. Broadly, this is the process of getting access to detailed hardware and software measurements driven for example, by benchmark applications. While there exist several instrumentation packages that are widely used there is significant replication of experimental data, variance in usage, and relatively little ability to share data or metadata. Consequently, the community would benefit greatly from standardization of techniques for data acquisition, storage, and analysis. In particular, if these techniques could be accessible to tools via standard APIs across all MODSIM areas, this would represent significant leverage and productivity improvements across the community as a whole. One concrete suggestion was the creation of a MODSIM repository that served as a home for data such as traces, input/output data sets, and workload generators. Some of the existing repositories could be coalesced to standardize and initialize a MODSIM repository as a home for storing and retrieving measurement and model-generated data in a standardized format. In particular, empirical methods that deal with diversity of platforms at multiple levels of fidelity can now generate shareable data sets. One class of data sets to include is failure data. Others include detailed profile and measurement data from benchmarks or real applications on new and emerging platforms.

Making cross-pollination easier can reap great dividends with relatively modest investments. For example, supporting inter-site gatherings for 1-2 day technical deep dives on mutually invested topics can promote and accelerate sharing and re-use.

Infrastructure: There is generally a lack of direct investment in engineering. It is most often a side effect of the modeling effort and consequently rarely portable across multiple MODSIM areas. There was a strong consensus on the need for sharing and supporting engineering efforts. Such efforts represent a sizeable investment for any organization and will continue to grow. Such growth is not sustainable if such efforts have to be repeated as the complexity and scale of systems increases. There are many facets of engineering effort that can be shared including the following.

- A common theme was the need for open source tools (most are) with standardized interfaces designed for interoperability (most are not) where the interfaces were independent of, and did not constrain, the models employed within them (most are not). In particular, at what level of modeling abstraction should such interfaces be standardized to minimize constraining modeling functionality and maximizing sharing and re-use?
- The development of a common API for standard simulator functions. How can such a standard be developed while maintaining some level of backward compatibility? The MPI standard was often invoked as an example of a successful standard.
- A MODISM repository for traces, benchmarks, performance data, application data, experiment scripts, etc. For example, trace gathering can be expensive task. While sharing currently does happen in informal groups, standardizing formats will make it easier to generate new contributions and share the results of substantive efforts.
- Modeling frameworks should be separated from the models – one community should not have to be experts in another community in order to exercise their skills. Investments in engineering frameworks to integrate models are a necessity. This should also help with the question of matching the tools to the right questions since the framework naturally provides guidance as the integration of tools of varying functionality.

Resilience: Resilience introduces a special set of needs with regards to modeling and simulation. In particular, there is little data of failure rates and behaviors in large-scale systems. For example, coalescing in a publicly available form failure data for existing 100,000+ core systems would be very valuable. Sharing of this data in some uniform manner will enable beneficial comparisons between failure mitigation techniques. Some agreement on fault models and failure statistics can lead to significant leverage of mitigation techniques and enable useful comparisons across competing approaches.

Frameworks: The community would like to minimize proliferation of tools with significant overlap in functionality and to promote productive re-use. The notion of a modeling stack and a common terminology can enable a clearer articulation of tool functionality, purpose, and usage modes. For example, one presentation pointed out that it was not necessary to use parallelism in all cases – serial models will suffice for many questions and in fact in some instances the difficulty of models of parallelism may nullify the expected gains. For example, software stacks provide a common set of concepts within a general framework (applications, operating systems, run-times, etc.) This makes it easier for distinct communities to pursue developments that can be integrated into overall functional software stacks. The seven-layer OSI stack is another example of generic framework that permits the independent development of interoperable software modules. A similar framework would be invaluable for the MODSIM community. This will also make it easier to match tools with the right predictive capabilities with the research question being asked.

Frameworks can also address a related important question – the need for a common set of metrics to evaluate simulators and compare their performance and functionality (e.g. accuracy) tradeoffs.

Stabilization: How can we transition research products in the form of tools to product grade functionality and stability? Many tools while critical for the scientific HPC market, are less so for the commercial market. Consequently, some important needs are unlikely to be made available commercially. Developing tools with product grade stability will significantly enhance cross-pollination. This will require investments in engineering and a mechanism to provide support over the lifetime of the specific tool in question.

1.4 Recommendations and Path Forward

Methodologies are at the core of all modsim endeavors. Discussions about this critical technology identified in the Modsim Report in 2012 are likely to be continued in future Modsim Workshops. Basically this is an ongoing, continuous process as the challenges of energy efficient, reliable computing pose new challenges to the field.