Whitepaper for Extreme Scale Science Panel – Jeff Candy, Phil Snyder, Vincent Chan, General Atomics

Summary
The high integrated cost (hardware, energy consumption, manpower) of exascale systems obviates the establishment of a quality standard, or a "suitability metric", for targeted scientific application software. In the suitability metric, we believe application performance (floating-point and parallel scalability) should be subordinate to verified scientific credibility and capability. At present, there are only weak mechanisms in place for guaranteeing the correctness of exascale applications.

Background
The ostensible goal of an extreme-scale science application is to yield a level of physical realism exceeding that achievable on mainstream – so-called capacity – computing platforms. While exascale computing certainly represents a new potential for scientific advance, certain serious pitfalls also emerge. In plain terms, applications may be selected with little or no regard to real scientific value. Our experience is that, too often, a review panel simply assumes that the chief scientist and his team are capable of objectively assessing the true validity and merits of their application code. With this assumption unchallenged, readiness for exascale may well be established based, essentially, on computing performance metrics alone.

Current approach and pitfalls
We emphasize again that a typical motivation for exascale code development is to improve upon the realism of existing HPC codes in a given domain. To achieve this end, code developers may add new features or introduce novel approaches to gain added realism. The danger here is that applications may be catapulted directly into an exascale vacuum. By this, we mean that there may be no traditional HPC codes with the same mathematical formulation or physical behaviour. The current INCITE program approach, for example, is to solicit a proposal outlining the scientific capabilities and merits of a candidate code. This proposal will typically argue for the computational readiness of the code by presenting extensive scaling and floating-point performance studies. A weakness of this approach is the lack of a formal method for establishing whether or not these massive, scalable simulations are correct or even sensible. This argues strongly for a suitability metric based on code verification in order to evaluate readiness of a code for allocation on exascale systems.

Suggestion for licensing
To prevent huge misappropriation of compute time that does not lead to true scientific knowledge advancement, we believe new, more stringent standards for verification should be required for codes positioned for execution on exascale systems. We emphasize, too, that this process should be decoupled from the

application to run on a particular exascale platform. The rationale for this decoupling is similar to that for vehicle licensing. For obvious safety reasons, one requires a driver's license in order to operate an automobile. On the other hand, there is no need to write one proposal to drive a Mustang and another to drive Porsche. All you only need is a single license that guarantees some acceptable standard of competence, i.e. a track record of rigorous verification guarantees a developer will follow the same in all future platforms. In the same way, the proposal process for exascale systems could be streamlined, since "code competence" will have been established through the licensing process. The requirements for licensing would be to set guidelines for the type and amount of verification (i.e, code benchmarking) required, but not to make the process overly rigid. When applying for the license, code authors would submit supporting documents to a panel of referees who would make an objective assessment. The referees would need to have some level of domain-specific expertise. In many ways, the process could be patterned largely after peer review of a scientific paper.

Barring this licensing, codes could be given access to systems on a kind of probationary basis to carry out optimizations or development. Application for probationary accounts would be streamlined and require very little scientific or HPC justification. But access to large amounts of computer time would only be allowed after the licensing process.

Validation

Once a code has been licensed, the next phase in its life-cycle – the validation phase -- can begin. This phase introduces new complexities, not necessarily encountered at the verification stage, related to handling of experimental profile and other diagnostic data. It also necessitates the need for suitable synthetic diagnostics, and for operation with enough physical realism to make validation meaningful. In this sense, it is possible for a code that is well-verified to perform poorly in the validation phase. However, successful validation is required before predictive modeling -- the final phase in the life-cycle -- can occur. While we do not propose a formal licensing protocol for these post-verification phases, we do emphasize that prediction before validation, or in particular before verification, are scientifically unsound practices and should be officially discouraged.