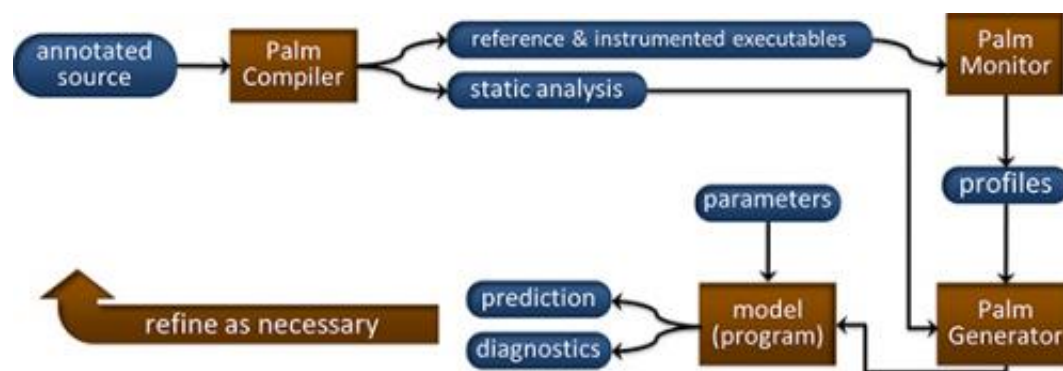# New tool eases the burden of creating and reproducing analytical performance models

May 6 2014

Palm Modeling Tool Workflow

Application performance modeling is an important methodology for diagnosing performance-limiting resources, optimizing application and system performance, and designing large-scale machines. However, because creating analytical models can be difficult and time-consuming, application developers often forgo the insight that these models can provide. To ease the burden of creating models, computer scientists at Pacific Northwest National Laboratory developed the Performance and Architecture Lab Modeling tool, or Palm. Palm simplifies the task of constructing a model by automating common modeling tasks and providing a mechanism for modelers to incorporate human insight. With Palm, reproducing a model—a program that runs on a computer—is straightforward. Given the same input, Palm generates the same model. This is a first step toward enabling the open distribution and cross-team

validation of models.

Application models are hard to generate. Models frequently are expressed in forms that are difficult to distribute and validate. PNNL researchers created Palm as a framework for tackling these problems. The modeler begins with Palm's source code modeling annotation language. Not only does the modeling language divide the modeling task into sub problems, it formally links an application's source code with its model. This link is important because a model's purpose is to capture application behavior. Furthermore, this link makes it possible to define rules for generating models according to source code organization. The model that Palm generates is an executable program whose constituent parts (functions) directly correspond to the modeled application.

"Given an application, a set of annotations, and a representative execution environment, Palm will generate the same model," explained Nathan Tallent, a PNNL research scientist and lead author of the paper describing the Palm tool. "We believe this presents intriguing possibilities for developing, distributing, and validating models." He also observed that the work to annotate an application should pay dividends because applications change relatively slowly.

Palm's source code modeling annotation language is designed to facilitate a divide-and-conquer modeling strategy by composing an application model from a number of sub-models, where sub-models correspond to application code blocks. Palm is designed to make the simple easy and the difficult possible by combining top-down (human-provided) semantic insight with bottom-up static and dynamic analysis. Palm generates models according to well-defined rules based on static and dynamic analysis. It also automatically incorporates measurements to focus attention, represent constant behavior, and validate models.

To demonstrate Palm's abilities, the scientists described the process of

generating models for three applications: Nekbone, a computational fluid dynamics solver; GTC, a gyrokinetic particle code; and Sweep3D, a neutron transport benchmark. They showed that Palm elegantly expressed each model and automated several modeling tasks.

The researchers are working on new techniques for assisting model generation using static source code analysis and critical path analysis. Because machine designs increasingly are driven by power considerations, they also are working on techniques for generating application models for metrics such as power and data movement.

More information: Tallent NR and A Hoisie. 2014. "Palm: Easing the Burden of Analytical Performance." In: *Proceedings of the 28th International Conference on Supercomputing*, ICS 2014. June 10-13, 2014, Munich, Germany. Association for Computing Machinery, New York.

Provided by Pacific Northwest National Laboratory