PARM: Physics Aware Runtime Manager for Large-scale Scientific and Engineering Applications

Yeliang Zhang, Salim Hariri
Electrical and Computer Engineering Department
The University of Arizona
1230 E. Speedway Blvd.
Tucson AZ 85721
U. S. A.
{zhang,hariri}@ece.arizona.edu

Jianwei Xiang, Jim Yeh
Hydrology and Water Resources
The University of Arizona
1133 E. James E. Rogers Way
Tucson AZ 85721
U. S. A.
{xjw,yeh}@hwr.arizona.edu

Abstract

Choosing the ideal algorithms and solutions for a scientific application is difficult because of the heterogeneity and dynamism of the application execution phases at runtime. In this paper we present an autonomic programming framework that is capable of self-configuring and self-composing the application solution methods in order to exploit the heterogeneity and the dynamism of the application execution states. We focus our approach on Partial Differential Equation (PDE) problems involving multiple computational phases that are defined in terms of their spatial and temporal characteristics. We have implemented a Physics Aware Runtime Manager (PARM) that periodically monitors and analyzes the spatial and temporal characteristics of the application to identify its current execution phase (state). Then PARM will determine an appropriate numerical schemes and algorithms that will most efficiently exploit the current state. Our preliminary results show a significant speedup can be achieved by using PARM.

1. Introduction

We propose a new paradigm to solve large-scale scientific applications. We monitor the application execution states, identify the application phase changes by exploiting the knowledge about the application physics and how it behaves at runtime, and then use the appropriate numerical solution/algorithm for each detected phase during the application execution. We evaluated our approach using a Finite Element Method (FEM) application (e.g., Variable Saturated Aquifer Flow and Transport (VSAFT2D)) [1]. Our preliminary results show a considerable speedup.

2. Physics Aware Runtime Optimization: Motivation

Most scientific transient problems could be divided into several computational phases, some of which are best represented by different physical domains where each domain is homogeneous with respect to the simulation's temporal and spatial characteristics. For example, VSAFT2D, a 2D finite element hydrodynamics application developed by the Hydrology and Water Resources Department at The University of Arizona [1]. It is designed for the simulation of water flow and chemical transport through variably saturated porous media.

Considering a computational domain composed by two different materials: clay and sand. According to the mass conservation law [2], the volume of water flow through different media in a unit time will be a constant in the steady state case [2],

$$K_{sand}J_{sand} = K_{clay}J_{clay} \tag{1}$$

where K is the hydraulic conductivity and J is the gradient of water head.

During the application execution, the application can be either in a saturated or unsaturated phase. However, each phase requires different configuration parameters to achieve the desired performance and accuracy requirements as explained below:

- Saturated Execution Phase: K is a constant and $K_{clay} < K_{sand}$, according to (1) $J_{clay} > J_{sand}$, J represents how fast the variation is. The greater the J is, the faster the variable changes, then the finer grid for fast changing region to achieve desired accuracy required. Hence, at runtime, the PARM approach will exploit the physics properties of the application by making the grid finer in the clay domain

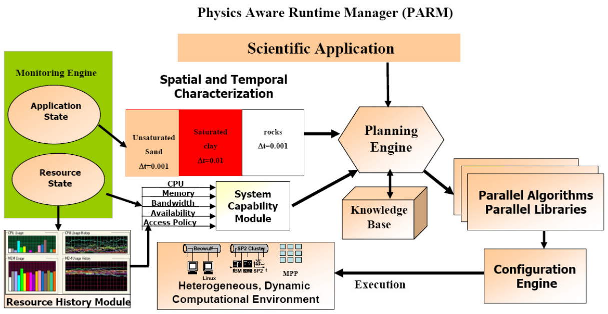

Figure 1. PARM Architecture

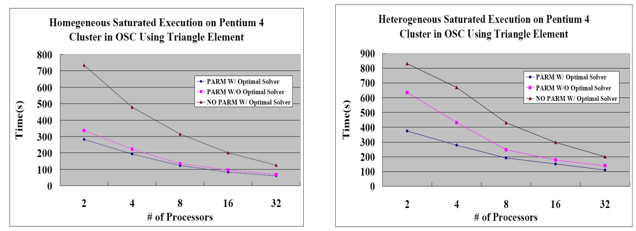

Figure 2. Execution Time with 1,000,000 triangle shape elements on Saturated Homogeneous and Heterogeneous Cases

($J_{clay} > J_{sand}$, we need to use a smaller Δx, Δy and Δz). In a similar way, the grip points will coarser for the sand region.

- Unsaturated Execution Phase: The hydraulic conductivity K is a function of waterhead h. The curves of $K_{sand}(h)$ and $K_{clay}(h)$ intersect at h' where $K_{sand} = K_{clay}$. When $h < h'$, $K_{clay} < K_{sand}$, so the $J_{clay} > J_{sand}$, we need a finer grid, i.e., reduce Δx, Δy and Δz for the clay domain and a coarser grid for the sand domain. However, when $h > h'$, $K_{clay} > K_{sand}$, so the $J_{clay} < J_{sand}$, we need to apply a coarser grid for the clay region but a finer grid for the sand region.

3. Overview of the Physics Aware Runtime Manager (PARM) Architecture

The architecture of PARM consists of five major modules: A Monitoring Engine, a Planning Engine, a knowledge base, numerical libraries and a Configuration Engine as shown in Figure 1.

The monitoring engine continuously measures the application operations (e.g., hydraulic conductivity and waterhead in VSAFT2D) in order to accurately characterize the application spatial and temporal properties. Then the monitored information is analyzed by the planning engine that determines if the current application configuration needs to be changed and if so, it will select the numerical schemes and execution strategies that optimize the application performance such as resize the mesh, change the time step or switch a linear system solver. The planning engine utilizes the knowledge stored in its knowledge base that has been acquired from empirical application execution and legacy numerical research [3]. Configuration Engines compiles all the numerical information together and generates the files needed by the application before it restarts the computation with new configuration.

4. Experimental Results

We integrated the PETSc [4] with our runtime manager (PARM) to provide linear solver algorithms. We have evaluated the performance of the PARM prototype to optimize VSAFT2D on The Ohio Supercomputing Center Pentium 4 Cluster. Each node of this cluster has 4G RAM and interconnected by an infiniband 10Gb interface.

Figure 2 shows the comparison of 3 scenarios: 1. PARM with optimal solver, 2. PARM without optimal solver, 3. Not using PARM on Saturated homogeneous and heterogeneous cases. Both cases shows that by using PARM, a considerable speedup can be obtained.

5. Conclusions and Future Work

In this paper we presented an autonomic programming framework that can dynamically self-configure the application execution environment to exploit the heterogeneity and the dynamism of the application execution states. We are currently applying PARM to applications like eigen value problems and also considering load balancing issues.

References

[1] Yeh. T. -C., J. R. Srivastava, A. Guzman and T. Harter, *A numerical model for water flow and chemical transport in variably saturated porous media*, Ground Water, 31(4), p 634-644, 1993.

[2] J. D. Istok, *Groundwater Modeling by the Finite Element Method (Water Resources Monograph),* Amer Geophysical Union, 1989.

[3] Y. Zhang, X. S. Li and O. Marques, *Towards an Automatic and Application-Based Eigensolver Selection*, LACSI Symposium 2005.

[4] S. Balay et. al. *PETSC Users Manual*, ANL-95/11-Revision 2.1.5, Argonne National Laboratory, 2004.