Task parallel sensitivity analysis and parameter estimation of groundwater simulations through the SALSSA framework

Karen Schuchardt\(^1\), Khushbu Agarwal\(^1\), Jared Chase\(^1\), Mark Rockhold\(^1\), Vicky Freedman\(^1\), Todd Elsethagen\(^1\), Tim Scheibe\(^1\), George Chin\(^1\), Chandrika Sivaramakrishnan\(^1\)

\(^1\)Pacific Northwest National Laboratory, PO Box 999 Richland WA 99352

E-mail: karen.schuchardt@pnl.gov

Abstract. The Support Architecture for Large-Scale Subsurface Analysis (SALSSA) provides an extensible framework, sophisticated graphical user interface, and underlying data management system that simplifies the process of running subsurface models, tracking provenance information, and analyzing the model results. Initially, SALSSA supported two styles of job control: user directed execution and monitoring of individual jobs, and load balancing of jobs across multiple machines taking advantage of many available workstations. Recent efforts in subsurface modelling have been directed at scaling simulators to take advantage of leadership class supercomputers. We describe two approaches, current progress, and plans toward enabling efficient application of the subsurface simulator codes via the SALSSA framework: automating sensitivity analysis problems through task parallelism, and task parallel parameter estimation using the PEST framework.

1. Introduction

Groundwater modeling is inherently uncertain due to our limited knowledge of the subsurface, being hidden from direct observation and for which only very small samples can be taken from a small number of locations, usually at significant cost. In addition, the subsurface contains a high degree of complexity formed over a very broad range of time scales with diverse geophysical and geochemical processes acting on a wide range of length scales [1]. Because of the uncertainty associated with developing a conceptual model of the subsurface, large numbers of simulations are often used to ascertain the sensitivity of the model to certain parameters, gain insights into how a model is behaving or to validate the model against a benchmark problem. These simulations may be conducted as a sensitivity analysis where a parameter space that is known \textit{a priori} is evaluated or through parameter estimation techniques that algorithmically optimize a parameter space under user specified conditions or constraints.

Sensitivity analysis and parameter estimation are popular techniques in groundwater modeling. However these can be time consuming to conduct and manage when applying sophisticated three-dimensional models that require large numbers of processors per simulation. We are developing high-level tools to enable users to fully leverage task parallelism on supercomputers with thousands to tens of thousands of processors. In this paper, we describe ongoing extensions to our groundwater simulation software environment that enable users to easily generate and run sensitivity analyses and our plans for supporting parameter estimation while maintaining full simulation provenance records.
The remainder of this paper is organized as follows: section 2 provides brief background information on our software environment and recent extensions; section 3 describes current developments to support sensitivity analysis applied to the Smoothed Particle Hydrodynamics (SPH) code [2], and section 4 describes our approach to supporting parameter estimation using Parameter Estimation Toolkit (PEST) [3] and the parallel flow and transport code Subsurface Transport Over Multiple Phases (STOMP) [4].

2. Background
The Support Architecture for Large-scale Subsurface Simulation and Analysis (SALSSA) software is a user environment for running simulations, tracking inputs and outputs of each simulation, recording provenance information, and accessing analysis and visualization tools. It can be thought of as both an activity tracking system and a dashboard to provide summary and detailed information about individual activities. While the framework was created for groundwater modeling, its design is generic and can be applied to simulators from any number of scientific domains. The key components are a content store for tracking artifacts associated with running and analyzing simulations (input, outputs, parameters), a provenance recording and display capability, a job launching component for launching simulations from the desktop to virtually any computer, a tool registry for adding new tools to the framework, and a baseline set of tools constructed specifically for the SPH and STOMP codes [5-6]. A number of capabilities have recently been added to the SALSSA framework and are summarized below.

Archive: Because simulations can generate data set sizes in the hundreds of gigabytes and beyond, we have added the capability for the user to selectively archive simulations. The content store maintains metadata about the outputs including references to their physical location. An “archive” is any machine that accepts files over one of the standard file transfer protocols. Users can add new archives to the system through a simple registration mechanism.

Reconnect: A “reconnect” function within the workflow enables users to reconnect to a running (or completed) simulation that was initiated and then disconnected from SALSSA due to network errors, computer shutdown (e.g. while on travel), etc.

Generic workflow Components: Generic workflow components have been developed to simplify building scientific workflows in Kepler. Two such components are the Data Transfer Actor which can be used to move files between systems over a variety of protocols and the Job Launching Actor which integrates a series of steps in the job execution process [7]. These actors have been incorporated into our workflows resulting in less complex workflows that are adaptive to more operating environments.

Command line tools: Command line tools have been developed for selected steps, such as job submission, that were previously accessed only through the user interface.

3. Sensitivity Analysis
Sensitivity analysis is typically conducted by running a suite of similar problems in which there are no dependencies between the simulations and the full set of desired runs is known a priori. These embarrassingly parallel simulations are frequently run on workstations or clusters but there is an increasing recognition that supercomputer task parallel techniques can be applied to run multiple simulations in parallel. In SALSSA, our objective is to make this operation transparent across a variety of systems.

For our representative use case, we selected a series of SPH simulations where the parameters of viscosity, speed of sound, and gravity are varied. Similar runs had previously been completed individually as part of a model validation activity [8]. To add this capability to SALSSA, a number of extensions were developed and are briefly described below.
3.1. Generating Many Simulations

A new capability, designed to support the current practice of planning and documenting sensitivity analyses using spreadsheets, reads a standard Excel spreadsheet and a template input file and generates a series of simulations (Figure 1) which can then be run using the new task parallel job launcher. This provides the user with full control over which simulations should be generated. The column labels in the spreadsheet control value substitution. For example, the viscosity column values will be substituted into the viscosity field in the input template. This capability is available as both a command line and interactive tool.

![Excel spreadsheet](image)

**Figure 1.** Excel spreadsheet used to define a set of task parallel SPH simulations and the resulting process objects within SALSSA. This spreadsheet contains a combination of settings where gravity and viscosity are varied. Units are dimensionless and specified in terms of the cutoff radius, particle mass, and speed of sound as defined by the SPH code.

3.2. Task Parallel Job Submission and Monitoring

Job submission, monitoring and all data transfer is implemented by a workflow developed within the Kepler Workflow system. The existing job submission components within Kepler have been modified to support multi-task job submissions and to record provenance information associated with each task within the job. The workflow (Figure 2) is responsible for generating temporary directories for each task and staging files to each task’s directory (2a), submitting the job (2b), monitoring the job and individual tasks (where possible) (2c), and recording metadata about the job/tasks (2d) and associated output files (2e).

![Kepler workflow](image)

**Figure 2.** Kepler workflow for executing multiple parallel tasks within a single job submission. Each of the components shown is itself a composite component containing its own sub-workflow. This workflow relies on the Kepler Data Transfer Actor, which is capable of staging files from multiple machines (e.g. desktop and archive).

3.3. Job Scripts

Scheduling on modern supercomputers is generally managed by a combination of workload manager (e.g. MOAB) and resource manager (SLURM, PBS) software. Most of these support the capability of scheduling multiple tasks in a single allocation. The job script for running a task parallel job will
depend on the scheduler and resource manager being used by a particular system. We present here the job scripts corresponding to a MOAB-SLURM system running at the Environmental and Molecular Sciences Laboratory (EMSL) (Figure 3b), and for PBS jobs running at the National Energy Research Scientific Computing Center (NERSC) (Figure 3a). MOAB-SLURM supports task parallel job execution, with built-in commands to monitor and control execution of each task independently. PBS does not have inherent support for task control and monitoring.

```
#PBS -q debug
#PBS -l mwidth=42
#PBS -l walltime=0:30:00
#PBS -j oe
cd $PBS_O_WORKDIR
for dir in dir0 dir1 dir2 dir3
do
cd $dir
  aprun -n 8 Scca/local/bin/cca-batch --cca-rc sph & > sph.out &
cd ..
done
wait
```

Figure 3a

```
## Nodes to use for all tasks
#PBS -l "nodes=4:ppn=8"
... 
nodesTask=1
nodeId=0
for dir in dir0 dir1 dir2 dir3
do
cd $dir
  mpirun -srun -n 8 -N 1 -r $nodeId -o job.out -e job.err $cca/local/bin/
    cca-batch --cca-rc sph & > sph.out &
  let "nodeId += $nodesTask"
cd ..
done
wait
```

Figure 3b

The general structure of task-parallel jobs involves allocating enough nodes to ensure tasks do not interfere with each other. This also ensures jobs are scheduled on independent sets of nodes. Since job wall-clock time is governed by the longest running task, the user should make certain that each task in the job requires a similar amount of time to run, so as not to waste resources. For simplicity reasons we can assume tasks in the job are square, executing the same code and if run on same number of nodes, will finish in nearly equal time. The job scripts are generated dynamically based on a user-defined set of simulations such as those specified in the spreadsheet of Figure 1 and a target machine and job options are given at the command line or through a user interface.

4. Parameter Estimation

For parameter estimation, our target use-case was a Hanford 300 Area modeling problem which, when run using a serial version of STOMP and the PEST framework, required a total of 364 forward runs to optimize 7 parameters using 7 PEST slaves. This problem took six weeks of continuous computation to complete. Our objective is to apply PEST to a parallel version of STOMP while also modifying PEST so that the slaves are subtasks within a job on the supercomputer. Based on some initial benchmarking of parallel STOMP on 32 processors, we estimate that by using on the order of 200 processors (32 per slave), the same problem could be completed in approximately 30 hours.

PEST may dynamically adapt its approach to computing derivatives. For example, depending on the configuration, it may use a forward difference method requiring one run per parameter and then switch to a central difference method, requiring two forward runs per parameter. PEST may also interrupt its normal processing to focus on a single parameter. Because of these types of adaptive behavior, there may be idle processors and time estimates are approximate. We are currently investigating environments.

5. Summary

We have extended the capabilities of both the SALSSA environment and the Kepler workflow system to enable the parallel execution of multiple, simultaneous parallel simulations on two different supercomputer platforms. This capability enables users to run sensitivity analyses more quickly than can be done in cluster environments. The components developed are general in nature ranging from a spreadsheet based user interface to a generic job launching and job script generation capability along
with workflow components for multi-task job submissions. In addition, we have begun to investigate
the adaptation of the PEST framework to support efficient parameter estimation within a single job on
a supercomputer. Efficient use of cycles is a key challenge requiring further investigation, although we
believe this approach should be effective for a broad class of problems and is being driven by the need
to reduce simulation time from months to days for complex 3-dimensional models.

Acknowledgments
This research is supported by the U. S. Department of Energy's Office of Science under the Scientific
Discovery through Advanced Computing (SciDAC) program. A portion of this research was
performed using the Molecular Science Computing Facility at EMSL, a national scientific user facility
sponsored by the Department of Energy's Office of Biological and Environmental Research and
located at Pacific Northwest National Laboratory. The work was conducted at the Pacific Northwest
National Laboratory, a multiprogram national laboratory operated by Battelle for the U.S. Department
of Energy under contract DE-AC05-76RLO1830.

References
framework for smoothed particle hydrodynamics simulations of reactive fluid flow in porous
media,” International Journal of High Performance Computing Applications, 24(2):228-239,
2010.
in numerical simulations,” In proceedings of IEEE - The Second International Workshop on
Scientific Workflows (SWF '08), Hawaii, July 8, 2008.
integration framework for subsurface simulations, Journal of Physics: Conference Series 78:012047
Context-aware scientific workflow systems using Kepler. Int. J. Business Process Integration and
“Application of SALSSA Framework to the Validation of Smoothed Particle Hydrodynamics
Simulations of Low Reynolds Number Flows.” Journal of Physics: Conference Series 180
(2009), 012065.