Early Prototyping and Evaluation of Grid-Oriented PDE Solver – Generation of Simulation Script from the Domain Specific Template

Zhou Jun [1)], Yukio Umetani [2)]

1) Graduate School of Science and Engineering, Shizuoka University, Japan (zhoujun@cs.inf.shizuoka.ac.jp)
2) Computer Science Dept., Faculty of Informatics, Shizuoka University, Japan (umetani@cs.inf.shizuoka.ac.jp)

1 Introduction

The potential of Grid computing [1] has greatly influenced the scientific computation and engineering fields. Yet, due to the intricacy of partial differential equation based problems and complexity of network issues, very few researchers and engineers except for the experts can harness the power of today's computational Grid. Development of Grid-enabled and user-friendly PDE based problem solving environment still remains a challenging task for PSE developers.

In the past decades, many commercial software packages or research PSEs have been developed for rapid prototyping of ideas and improving productivity. However, for some sophisticated PSEs, the learning curve is long and sometimes cumbersome, especially for users who are often computer literate but not necessarily trained computer scientists. On the other hand, the design on communication between human and PSE is still weak, especially on the issue what work the user needs to do and what the PSE needs to handle. Furthermore, friendly graphical user interfaces still need to be carefully designed since the user-GUI interactions are not only for simple inputs but also for assisting users to perform their simulations in an intuitive and straightforward way.

Consequently, we argue that a user-friendly PSE on which researchers and engineers can perform PDE based physical simulations by utilizing glHFpI3bA4-wWFeOiIq-bP-W'OFW] ';,FtI''bw'PF IqP'wbP-3FrIqwbO'4HFeI-iIW'OFsI-b-4'H4FnIq3bWO3WOFuIq3b

Platform Independence

where σ_x, σ_y and σ_z are the non-zero axis stress components, τ_{xy}, τ_{yz} and τ_{xz} are the shear stress components, F_x, F_y and F_z are body forces, per unit volume; u, v and w are small displacements along axes, ρ is density, R is damping constant

3.2.4 Applying Boundary Conditions

Three kinds of boundary conditions, namely Dirichlet Condition, Neumann Condition and Mixed Condition can be specified in each PDE wizard. The user only needs to input the right side fields with values or expressions. Besides, same boundary condition applied on different boundary regions can be specified at the same time by multiply selecting items from boundary region list.

In our example, the boundary condition applied on the four sides except the top and bottom of the piano soundboard is the same. That is, fixed displacement equals to zero. So they can be defined by first input right side values, u = 0, v = 0, w = 0, and press "add" button after these boundary items have been multiply selected from the boundary region list.

As for Neumann Condition applied on the top and bottom of the piano soundboard, the specification is essentially alike. The user also only needs to provide the right side values or expressions. The PDE wizard will generate corresponding boundary condition statements that the PDE solver can understand. Take the Neumann Condition applied on the top for an example, the user only needs to input zero for the stresses to X and Y axes, and select "Sz" from the defined parameter list (which has been passed to this step for selection) for the stress to the Z axis.

3.2.5 Applying Initial Conditions

Initial condition is solely for the time-dependent simulations. If the temporal flag described in 3.3.1 is time independent, the menu for specifying initial condition will be disabled when loading a PDE wizard main framework.
Since this project is time dependent, the user first needs to input the beginning time, ending time and time interval for this project. After that, specifies the initial condition for it. Here corresponding to displacement equals to zero along each axis at initial stage.

3.2.6 Specifying Solution Scheme

In this step, the user can specify iteration steps and numerical method (such as "Gauss" or "PCG"), call built-in integration subroutines, select listed unknowns for obtaining solution, print monitoring results on screen and save results. In this example, we ignore body forces, select "PCG" as solution method with iterations of 1000 for testing purpose and print the value of Wkoma at each iteration step on the screen.

3.2.7 Connecting Server for Execution

The generated simulation script then is sent to the solver together with the mesh file for this project for execution. Presently, the simulation can only be performed in a remote computer through local networks in our laboratory. Since this paper is mainly demonstrating the process of prototyping and evaluation of the process of generation of simulation script, we do not list the solved results

3.3 Generated Simulation Script

The generated script in this example is in a format for PSILAB solver, which translates it into high-level language code for execution. See Figure3 for the simplified simulation script generated for this soundboard project.

4 Conclusions and Future Work

```
PROG     SOUND;
METHOD   FEM;
DOMAIN   X=[-710.0:710],Y=[-590.0:590.0],Z=[-4.0:4.0];
Time     T=[0:1]; …
BOUND    S8+S30+S17+S21+S25+S29;
MESH     FILE;
CONST    Ex = 1.15640D+07,  …,CCz=Ez*(1-xy*PRxy*Ex/Ey)/Ch;
CTENS    A11=(CAx,0,0,0,Gxy,0,0,0,Gxz),  …
          A33=(Gxz,0,0,0,Gyz,0,0,0,CCz);
VAR      Sz ,Wk, U, V, W, EPSx, EPSy, EPSz, GAMxy, …;
SVAR     Wkoma;
BCOND    U=0  at S17+S21+S25+S29,  …
        N(Cu..grad(U))+N(Cv..grad(V))+N(Cw..grad(W))= -Sz at 30;
ICOND     U0= 0, U1= 0,   …
SCHEME;
      iter CT until CT EQ 1000; …
       solve U of
       (Rho/dlt**2 +R/(2*dlt))*U =
          2*Rho/dlt**2*U1 + (R/(2*dlt) - Rho/dlt**2)*U0 +…
       by 'PCG' under BGRP01; …
       PRINT CT, Wkoma; …
END SCHEME;
END;
```

Fig.3. Simplified Simulation Script

No PDE based problem solving system so far has targeted the specification of PDE problems in a way that we have presented although some of them also provide GUIs as well. With GridPSi, a given simulation becomes more specific and fairly straightforward since the beginning. Still, the simulation script based client/server model service provided by our system can greatly reduce the network consumption and delay. In contrast, most current existing PSE providers let users access their remote servers via a Java Applet based web portals to describe and define their PDE problems, such as WebPellpack [6] for an example, the user needs to log onto the remote server and describe their problem page. Such kind of online description is usually inefficient and incapable for dealing with complicated PDE-based simulations.

The GridPSi system is capable of providing computational researchers a user-friendly environment for conducting PDE-based simulations in an intuitive and straightforward way, hence eases their prototyping processes and increase productivity. It also enables the users to run powerful applications on the computational Grid, taking advantages of global resources such as supercomputers.

The GridPSi system is still being under design and implementation. The next step is to construct a Unicore server in our lab. to test the real grid accessibility, finish the post-analysis editor construction and extend the system by building other functional PDE wizards and solvers.

References

[1] Ian Foster, The Grid: Computing Without Bounds, Scientific American Digital, April, 2003.

[2] Zhou Jun and Y.Umetani, *GridPSi:* A Grid-Enabled Problem Solving Environment for Physical Simulation Applications, Proceedings in the 7th PSE Workshop&Grid Seminar, Japan, 2003.

[3] http://www.unicore.org

[4] http://www.hitachi-ul.co.jp/PROD/PSILAB

[5] http://www.geuz.org/gmsh

[6] E. N. Houstis, et al. PELLPACK: A Problem-Solving Environment for PDE-Based Applications on Multi-Computer Platforms. ACM Trans. Math. Softw., Vol. 24, No. 1, pp. 30--73, March 1998.