CS6230 Coursework 2 Due February 21st 2014

Introduction

This exercise concerns the idea of partitioning a problem in different ways, when iteratively solving equations and then the accuracy that is obtained with different schemes for the amount of work that is done on a parallel machine. You are provided with serial and parallel codes that implement a solution to the Poisson problem,

$$\frac{\partial^2 a}{\partial x^2} + \frac{\partial^2 a}{\partial y^2} = 0$$

using a Jacobi iteration, see pp.357-360 of Wilkinson and Allen, where (x_i, y_j) are the grid points in the domain $[0,1] \times [0,1]$, when $a(x,y)$ is specified on the edges of the domain using the exact solution. The exact solution is given by

$$a_{ij} = sin(\pi x_i) * exp(-\pi y_j)$$

These codes are the basis for all work in this exercise so you should read through them carefully and understand them.

Tasks

Task 1 - 10 percent

Test the network performance of the CHPC Telluride machine using the provided program for measuring communictions performance. Use timings to create a performance model and derive an expression for the speedup possible with the supplied code.

Task 2 - 10 percent

Test the scalability of the existing code on the cluster. Look at cluster sizes up to 64 cores and graph the results to show the trend. Vary the mesh sizes used. Compare the actual speedup with the model predictions.

Task 3 - 25 percent

Modify the code so to implement the Red-Black iteration with over relaxation. where the relaxation factor is chosen by you using experiments so that the iteration converges. Comment on the difference in parallel performance and scalability between this version, and the original code. The optimum parameter is suggested in a recent publication as $s = \frac{2}{1+sinh(\pi h)}$. Compare the performance of the two methods. Vary the mesh sizes and the number of processors and produce tables of the time to reach a certain accuracy errors as defined by.

$$Error = maxi, j|(a_{ij}^{computed} - a_{ij}^{exact})|$$

Note - you will have to run the iteration until it converges for different values of p and grid sizes. It is also OK (at least initially) to start with the exact analytical solution as the initial guess.

Task 4 - 25 percent

Consider the extension of the method to problems in a cube in 3D where the main cube is decomposed into smaller cubes in exactly the same way as the squares is decomposed into smaller squares. In this case the ghost cells are six *nxn* planes around a cube of size *nxnxn*. The equations are given by:

$$\frac{\partial^2 a}{\partial x^2} + \frac{\partial^2 a}{\partial y^2} + \frac{\partial^2 a}{\partial z^2} = 0$$

with analytical solution

$$a_{ij} = sin(\pi x_i) sin(\pi y_j) \frac{sinh(\sqrt{2}\pi z_k)}{sinh(\sqrt{2}\pi)}$$

where (x_i, y_j, z_k) are the grid points in the domain $[-1,1] \times [-1,1] \times [-1,1]$. Again compare the performance of the original method and the Red-Black SOR Method. You will need to implement a 3D version of the "Jacobi" algorithm (that solves the Poisson equation), and then will do the same type of comparisons/analysis with this code as you did with the "Red Black" code of Task 3.

Task 5 - 15 percent

Derive expressions for the speedup and efficiency of the new method.

Task 6 - 15 percent

For a calculation such as those you have done above it is possible to define stencils of width $2m+1$ components in each of d dimensions. Evaluating the stencil then costs $2md$ operations. For example the case $d = 3$ and $m = 1$ is the case you have just coded. Obviously in a d dimensional case it is necessary to transmit a face of the cube and to also transmit data m deep. Using a modification of the scalability analysis in the lecture slides derive an expression for constant efficiency in terms of the parameters d, m, t_s, t_d and n where each processor has subdomain is with n^d elements.

Deliverables

Please submit a report on paper or electronically:

1. Evidence to show the communications performnce of Telluride.
2. Evidence to show the scalability of the original code on the the cluster.
3. Evidence to show the scalability of the new code on the the cluster.
4. A comparision of the performance of the two methods.
5. Example results of the performance of the two methods and a discussion of these results.
6. An efficiency analysis of the new code(Task 5).
7. An efficiency analysis of the generalization of the new code (Task 6)

Please submit electronically your code for Tasks 2 and 3 and your report (pdf form) in a zip file.

Deadlines

The work should be submitted by 5pm February 11th. Electronic submissions should be emailed to me and the paper submissions should be handed in to me at my office, or in the lecture.