Instituto Geofísico-EPN

The Jacobi Iterator

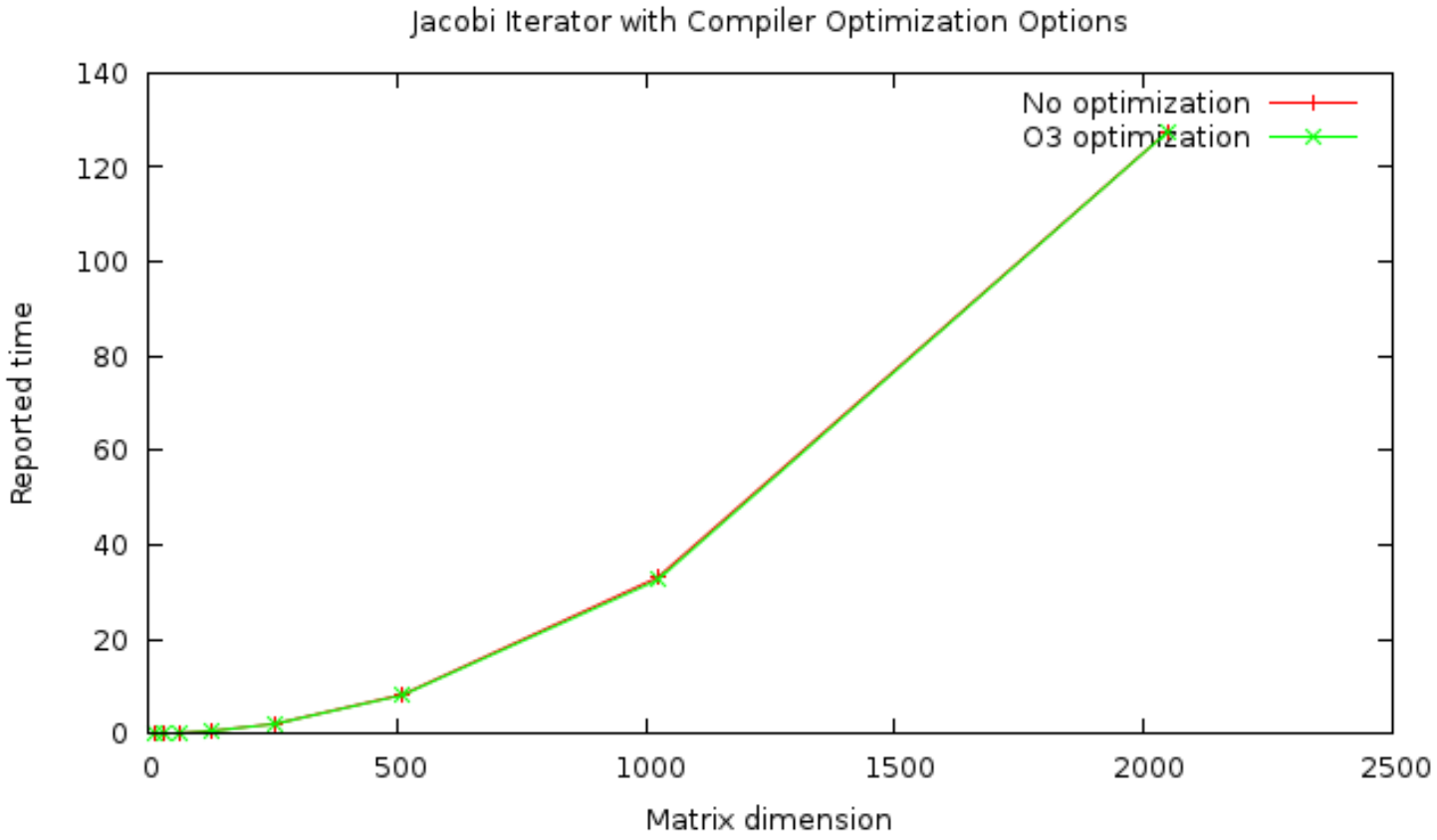
(to solve Laplace's equation)

Freddy Vásconez

Ecuador

Instituto Geofísico-EPN

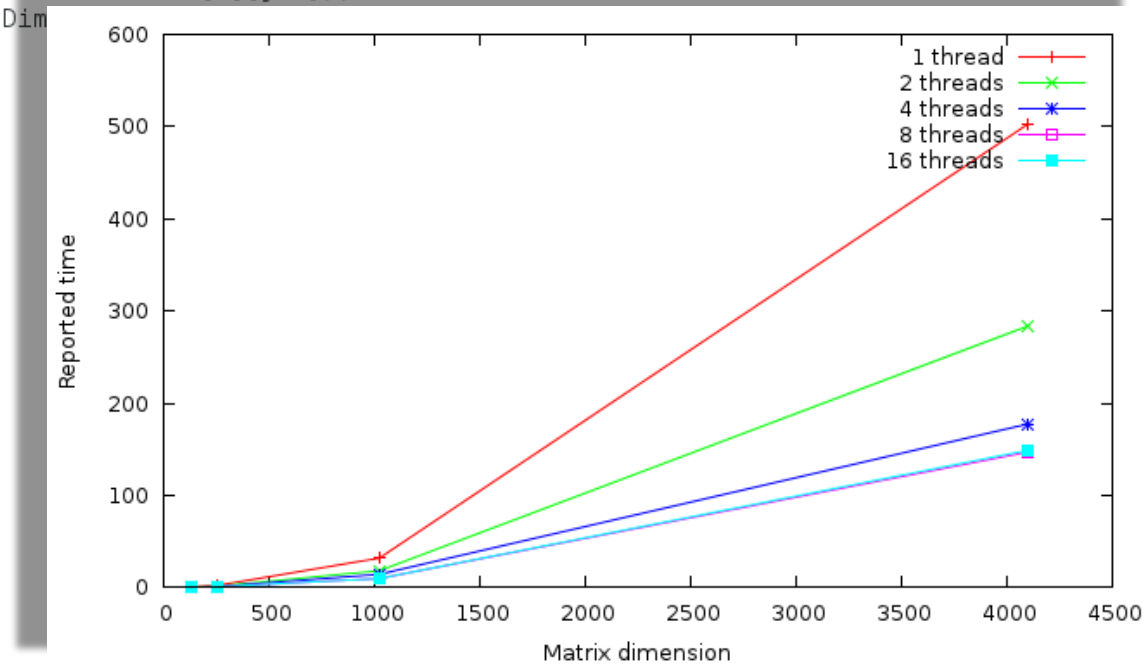
Compiler Optimization



Instituto Geofísico-EPN

Open MP

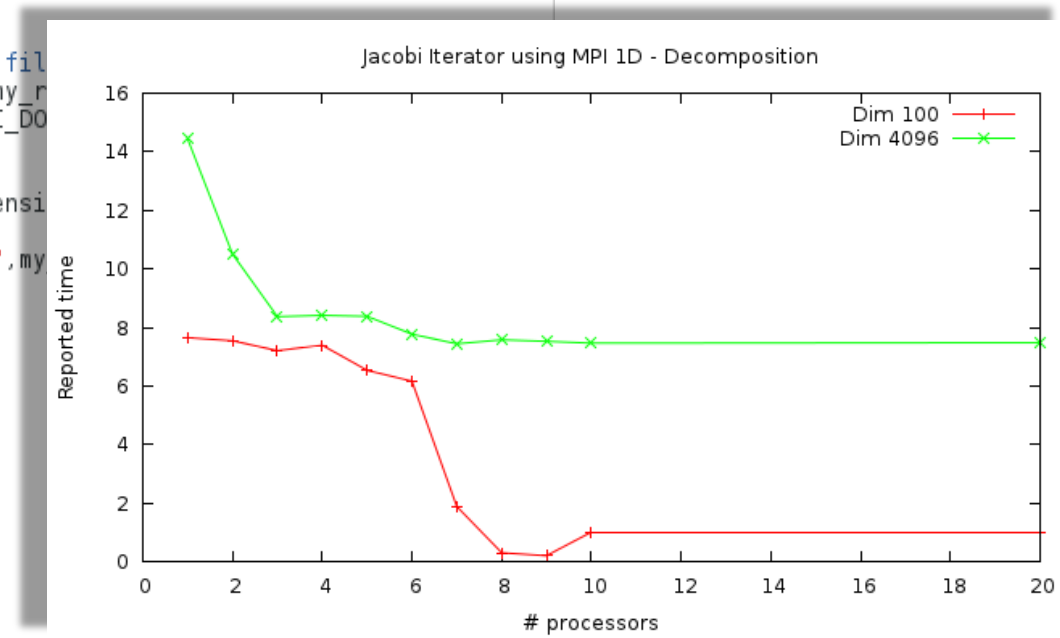
```
73   // Itertate
74   double TimeStart = WallTime();
75
76   #pragma omp parallel for
77   for(int iCount = 1; iCount <=Iterations; iCount++){
78
79     //This will be a row dominant program.
80
81     for(int i=1;i<=Dimension;i++)
82       for(int j=1;j<=Dimension;j++)
83         SurfaceMatrix_t[i][j] = (0.25)*(SurfaceMatrix[i-1][j] +
84                                         SurfaceMatrix[i][j+1] +
85                                         SurfaceMatrix[i+1][j] +
86                                         SurfaceMatrix[i][j-1]);
87     PrintSurfaceMatrix(SurfaceMatrix_t,Dim
88     double ** tmp;
89     tmp = SurfaceMatrix;
90     SurfaceMatrix = SurfaceMatrix_t;
91     SurfaceMatrix_t = tmp;
92   }
93   double TimeEnd = WallTime();
```



Instituto Geofísico-EPN

MPI: 1D - Decomposition

```
 92
 93    // Once each process has finished, sends a message to the next with the corresponden
    t ghost cell
 94    int next, prev;
 95    next = (my_rank+1)%size;
 96    prev = (my_rank-1+size)%size;
 97 if(DEBUG)printf("P%d: Sending row %d to P%d\n",my_rank,load[my_rank]+1,next);
 98    MPI_Isend(&SurfaceMatrix[load[my_rank]*(Dimension+2)+1], Dimension+2, MPI_DOUBLE, ne
    xt, 0, MPI_COMM_WORLD, &request);
 99
100    // Receive the message sent by the previous process
101    MPI_Recv(&SurfaceMatrix_t[0], Dimension+2, MPI_DOUBLE, prev, 0, MPI_COMM_WORLD, MPI_
    STATUS_IGNORE);
102 if(DEBUG)printf("P%d: Received row %d from P%d\n",my_rank,load[my_rank]+1,prev);
103    // Wait until the message has been delivered and continue to calculate
104    MPI_Wait(&request, MPI_STATUS_IGNORE);
105
106
107    // Send a message to the previous process to fil
108 if(DEBUG)printf("P%d: Sending row %d to P%d\n",my_r
109    MPI_Isend(&SurfaceMatrix[0], Dimension+2, MPI_DO
    est);
110
111    MPI_Recv(&SurfaceMatrix_t[load[my_rank]*(Dimensi
    ext, 0,MPI_COMM_WORLD, MPI_STATUS_IGNORE);
112 if(DEBUG)printf("P%d: Received row 0 from P%d\n",my
113
114    MPI_Wait(&request, MPI_STATUS_IGNORE);
115
```



Instituto Geofísico-EPN

MPI: 1D - Decomposition

```
143    // ******************************
144    // *** Master process' section ***
145    if(!my_rank){
146
147       double *CompleteMatrix;
148       CompleteMatrix = (double *) malloc((Dimension+2)*Dimension*sizeof(double));
149
150       for(i=0;i<load[0];i++){
151          for(j=0;j<Dimension+2;j++)
152             CompleteMatrix[i*(Dimension+2)+j]=SurfaceMatrix[(i+1)*(Dimension+2)+j];
153       }
154
155       // Collect the information from everybody but myself
156       for(i=1;i<size;i++){
157          MPI_Recv(&CompleteMatrix[i*load[i]*(Dimension+2)], load[i]*(Dimension+2),
    MPI_DOUBLE, i, 1, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
158       }
159
160
161       FILE *outfile;
162       outfile = fopen("jacOut.dat","w");
163
164       fprintf(outfile,"\t");
165       for(i=0;i<Dimension+2;i++){
166        fprintf(outfile,"%d\t",i);
167       }
168       fprintf(outfile,"\n");
169
```