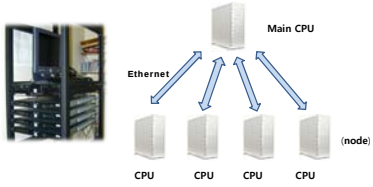
Parallel Computing on Linux cluster

Global KU Frontier Spirit
이동선(dosnen2@hotmail.com)
김준석(junseok_kim@yahoo.com)

클러스터의 이해

클러스터는 각 각 노드(node)들이 메인 컴퓨터를 중심으로 네트워크를 이루어 마치 슈퍼 컴퓨터의 병렬화된 프로그램과 같은 효과를 내는 장치를 의미한다.



클러스터의 장점

비교적 저렴한 가격으로 슈퍼 컴퓨터의 성능을 낼 수 있으며 또한 클러스터에 맞게 프로그램을 개발하고 운영체제를 개조할 수가 있다.

클러스터 이용하기

유한 차분법을 이용하여 열방정식을 풀고 그 계산을 클러스터에서 병렬 계산으로 풀어보자.

C를 이용한 열방적식 풀기

유한 차분법으로 푼 열방정식

우선, 열방정식을 유한 차분법으로 풀어본다.
다음과 같은 경계조건을 만족시키는 열방정식이 있다고 하자.

$$u(x,0) = \sin(\pi x) \quad u(0,t) = u(1,t) = 0$$

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t) \quad 0 < x < 1, \quad t > 0,$$

위 열방정식의 $\partial u/\partial t$와 $\partial^2 u/\partial x^2$를 유한 차분법을 써서 u_i^{n+1}에 대해 나타낸다.

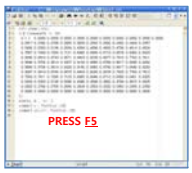
$$u_i^{n+1} = \alpha u_{i+1}^n + (1-2\alpha)u_i^n + \alpha u_{i-1}^n \quad \text{where} \quad \alpha = \frac{\Delta t}{h^2}$$

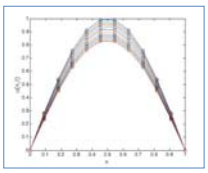
위 식을 C로 풀어본다.

```
#include <stdio.h>
 #include <stdlib.h>
 #include <string.h>
 #include <math.h>

int main()
 {
int i ,j, N, Nt, L;
N=12;Nt=10;L=1;

double x, k, h, pi;
k=0.002;h=0.0909;pi=4*atan(1);
double U[N][Nt];

FILE *fp;                              /*ⓐ*/
char filename[256], buffer[10];
for ( i = 0; i <= N-1 ; i++ ) {
 x = h * i;
U[i][0] = sin(( pi * x ));             /* initial condition */
}

for (j=0; j< Nt ; j++)
{
U[N-1][j]=0;
U[0][j]=0;
}

for ( j = 1; j < Nt ; j++)             /* Finite-Difference method */
{
for ( i = 1; i < N -1 ; i++)
U[i][j]=(k/(h*h))*U[i-1][j-1] + (1-2 * (k/(h*h)) )*U[i][j-1] + (k/(h*h))*U[i+1][j-1];
}

fp = fopen("test.m", "w");             /*ⓑ, create test.m file*/
fprintf(fp, "clf; clear; clc; \n");
fprintf(fp, "x = linspace(0, %d, %d)\n ", L, N);
fprintf(fp, "U = [ ");

for (i = 0; i <N ; i++)
for (j = 0; j < Nt; j++)
{
fprintf(fp, " %0.4f", U[i][j]);
if (j == Nt-1 )
fprintf(fp, "\n");
}
fprintf(fp, "]; \n");
fprintf(fp, "plot(x, U , '-o-' ) \n");
fprintf(fp, "xlabel('x','FontSize',13) \n");
fprintf(fp, "ylabel('u(x,t)','FontSize',13) \n");
fclose(fp);
return 0;
 }
```

ⓐ FILE* fp을 이용하여 matlab파일을 만들자.
위 파일을 컴파일하고 실행을 하면

ⓑ test.m이라는 파일이 나온다.
matlab파일을 열고 실행(F5)하면 아래와 같은 그래프가 나온다.





병렬 계산하기

이제, 각 core별로 다른 초기조건을 가지는 열방적식을 풀어보자.
먼저 C코드를 보자.

```
/*heateqn.c*/
#include <stdio.h>
 #include <stdlib.h>
 #include <string.h>
 #include <math.h>
 #include "mpi.h"                      ⓐ

 int main(int argc, char **argv)       ⓑ
 {
int p, rank;                           ⓒ
int i, j, N, Nt, L;
N=12;Nt=10;L=1;

double x, k, h, pi;
k=0.002;h=0.0909;pi=4*atan(1);
double U[N][Nt+1];

FILE *fp;
MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &p);     ⓓ
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
char filename[256];
char buffer[10];

for ( i = 0; i <= N-1 ; i++)
{
x = h * i;
U[i][0] = sin(( pi * x ))* rank ;      ⓔ
U[i][10]=0;
}
for (j=0; j<= Nt ; j++)
{
U[11][j]=0;
U[0][j]=0;
}
for ( j = 1; j < Nt ; j++)
for ( i = 1; i <= N -1 ; i++)
U[i][j]=(k/(h*h))*U[i-1][j-1] + (1-2 * (k/(h*h)) )*U[i][j-1] + (k/(h*h))*U[i+1
][j-1];
strcpy(filename, "heateqn");           ⓕ
        sprintf(buffer, "%d", rank);
        strcat(filename, buffer);      ⓖ
        strcat(filename, ".m");
        fp = fopen(filename, "wt");

printf("%d¥n",rank);
fprintf(fp, "clf; clear; clc; ¥n");
fprintf(fp, "x = linspace(0, %d, %d)¥n ", L, N);
fprintf(fp, "U = [ ");
for (i = 0; i <= (N - 1) ; i++)
for (j = 0; j <=Nt; j++)
{
fprintf(fp, " %0.5f", U[i][j]);
if (j == Nt)
fprintf(fp, "¥n");
}
fprintf(fp, "]; ¥n");
fprintf(fp, "plot(x, U , '-o-' ) ¥n");
fprintf(fp, "xlabel('x','FontSize',13) ¥n");
fprintf(fp, "ylabel('u(x,t)','FontSize',13) ¥n");
fprintf(fp, "axis([0 1 0 3]) ¥n");
fclose(fp);
return 0;
 }
```

병렬 계산을 위해서

ⓐ #include "mpi.h"를 추가해준다.

ⓑ int main()대신에 int main(int argc, char **argv)을 쓴다.

ⓒ int p는 프로세서의 수 변수 p
int rank 프로세서의 고유번호 rank를 의미한다.

ⓓ 다음 명령문을 추가해준다.
MPI_Init(&argc, &argv);
MPI_Comm_size(MPI_COMM_WORLD, &p);
MPI_Comm_rank(MPI_COMM_WORLD, &rank);

ⓔ 경계 조건 중 초기값에 rank를 곱해준다.

ⓕ 출력되는 파일의 이름을 정해준다.

ⓖ strcpy와 strcat로 출력될 matlab파일의 확장자명을 정한다.

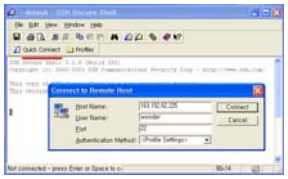
앞의 C코드 heateqn.c를 클러스터에서 실행해보자.

준비하기

-프로그램 설치하기
SSH Secure Shell 프로그램을 설치한다. (http://www.ssh.com/downloads/)

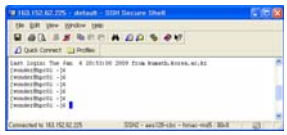
-클러스터에 접속하기

SSH Secure Shell를 설치하였다면 Client을 클릭하여 클러스터에 접속해야 한다.
Quick Connect를 선택하면 Connect to Remote Host가 나온다.



Host Name:에는 클러스터의 IP와 User Name:에는 사용자 이름을 쓴다.
예) 163.152.62.225
User Name: wonder

입력 후에는 Connect을 클릭하고.
PASSWORD: **** 입력한다.



SSH을 통해 Cluster에 접속했다.

-클러스터에 파일 보내기

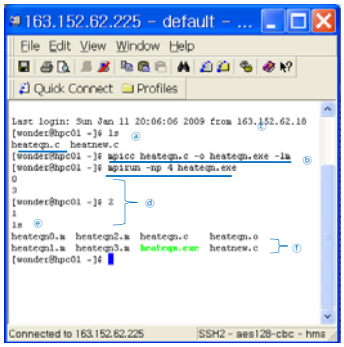
SSH Secure File Transfer Client에서
Quick Connect을 클릭한 후
Host Name에는 클러스터 IP를
User Name에는 사용자 이름을 쓴다.

비밀번호를 입력하고
Remote Name에 보내고 싶은 파일을 끌어다 놓으면 클러스터에 파일을 보내게 된다.
예)
Remote Name아래에 보면
Heatnew.c 라는 파일이 클러스터에 전송되었다.

컴파일하기

SSH Secure Shell에 접속해서 C코드를 실행해 본다.

SSH Secure Shell에 접속한다



ⓐ: ls 명령문으로 컴파일할 파일 heateqn.c가 클러스터에 있는 것을 확인하자.
만일 없다면
SSH Secure File Transfer Client을 이용해서 파일을 클러스터로 옮긴다.

ⓑ: 클러스터에서 컴파일하기
명령문: mpicc [컴파일될 파일.c] –o [컴파일된 파일.exe] –lm
-o 뒤에는 원하는 파일 이름을 쓴다.
-lm C코드에 수학 기호가 있으면 쓴다.
예)mpicc heateqn.c –o heateqn.exe -lm

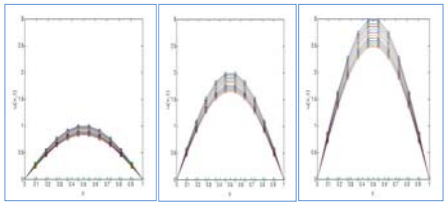
ⓒ: 실행하기
명령문 :mpirun –np [process의 수] [컴파일된 파일]
예)mpirun –np 4 heatnew.exe

ⓓ: 클러스터의 rank 0, 1, 2, 3이 반응하였다.

ⓔ: ls 명령문으로 생성된 파일을 확인한다.
heateqn0.m, heateqn1.m, heateqn2.m, heateqn3.m이 생성되었다.

결과 그래프

heateqn.exe를 실행시킨 결과



heateqn1.m heateqn2.m heateqn3.m

Acknowledgements

This work was supported by the research fund (R0600842) of Seoul Research and Business Development Program.

Reference

-Richard L. Burden, J. Douglas Faires, Numerical Analysis,Thomson
-Barry Wilkinson, Michael Allen, Parallel Programming, Prentice Hall
-Peter S. Pacheco, Parallel Programming with MPI, Morgan Kaufmann