GOETHE UNIVERSITÄT FRANKFURT AM MAIN

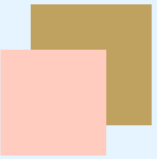
Assignment 2, Task 3
Parallelization So2011

Pattreeya Tanisaro

May 30, 2011

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Tasks



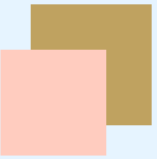
- Compute the following integrals $\int_{0.1}^{10000} \frac{1}{x} dx = 11.512925$ and $\int_{10}^{2000} \sin^2 x dx = 995.399112$ using Composite Simpson's 3/8 Rule
- Process 0 calculates subinterval limits for each process and use MPI_Scatter sends the result to the other processes
- a) Use MPI_Reduce to send the results of the subintervals to process 0
- b) Use MPI_Alltoall to send all results of the subinterval to all processes and calculate the complete integral
- All processes send their time measurement with MPI_Gather to process 0
- Measure computation and communication time for sequential and parallel processing and compare the results

Executable - collectives

```
$ mpiexec -np 16 -f hosts ./collectives a 1000000 0.1 10000 Alltoall y
```

- The first 4 parameters are similar to Task 1 and 2. The first parameter is to select the function we want to integrate where a for $f(x) = \frac{1}{x}$ and b for $f(x) = \sin^2 x$ etc.,
- Second parameter is the number of intervals whereas the third and fourth parameters are the lower and upper limits
- Fifth parameter is the communication methods, in this problem either Reduce or Alltoall
- Sixth parameter is an option to show detail timing

Executable - collectives



```
$ mpiexec -np 16 -f hosts ./collectives a 1000000 0.1 10000 Alltoall y
```

- The first 4 parameters are similar to Task 1 and 2. The first parameter is to select the function we want to integrate where a for $f(x) = \frac{1}{x}$ and b for $f(x) = \sin^2 x$ etc.,
- Second parameter is the number of intervals whereas the third and fourth parameters are the lower and upper limits
- Fifth parameter is the communication methods, in this problem either Reduce or Alltoall
- Sixth parameter is an option to show detail timing

run_collectives.sh will perform all the works with various intervals, number of processes and communication operations

Performance Analysis (1)

Table 1:
$f(x) = \frac{1}{x}$ with $N = 1000000$

Num of Proc	MP_Reduce			MPI_Alltoall		
	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec
1	536	89149	89685	36	90647	90683
4	7438	34533	41971	9951	33445	43397
8	5478	19111	24590	7726	18867	26593
16	5995	9574	15570	15478	9606	25085
64	5962	2449	8411	23237	2468	25706
128	6766	1255	8021	15342	1283	16626

Table 2:
$f(x) = \frac{1}{x}$ with $N = 100000000$

Num of Proc	MP_Reduce			MPI_Alltoall		
	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec
1	515	7699779	7700294	52	7717962	7718014
4	4330	1940359	1944690	31095	1932040	1963135
8	7239	966121	973360	24609	966726	991335
16	8824	491460	500285	15942	496435	512378
64	13663	143822	157486	28085	144960	173045
128	29559	74629	104188	90591	75222	165813

Performance Analysis (2)

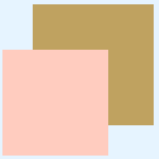


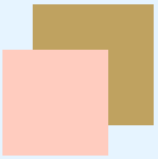
Table 3:
$f(x) = \sin^2 x$ with $N = 1000000$

Num of Proc	MP_Reduce			MPI_Alltoall		
	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec
1	49	358573	358622	45	356621	356666
4	4060	98621	102681	11715	97829	109545
8	6738	57042	63781	20072	61855	81928
16	7085	36175	43261	16488	37522	54011
64	13382	10758	24141	16892	10764	27657
128	13820	5416	19237	23678	5554	29233

Table 4:
$f(x) = \sin^2 x$ with $N = 100000000$

Num of Proc	MP_Reduce			MPI_Alltoall		
	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec
1	37	33625671	33625708	54	33480708	33480762
4	13308	8374577	8387885	14842	8378468	8393311
8	7759	4200477	4208236	22843	4191904	4214748
16	6449	2105906	2112355	25900	2107526	2133426
64	12477	545844	558321	28693	547614	576307
128	53788	273488	327276	294007	276730	570738

Conclusion



- Collective communication is defined as comunication that involves a group or groups of processes. In this task, we applies MPI_Scatter to send calculated subintervals from process 0 to all other processes and MPI_Gather to send communication and calcuation times from all other processes to process 0.
- As expecting, using MPI_Alltoall for communication takes more communication time than using MPI_Reduce. MPI_Alltoall behaves similarly to butterfly network which we implemented in Task2 usings $log_2 p$ dimension hypercube as communication pattern proceeding in $p-1$ phases. $T_{alltoall}$ takes $(o+L+o)(p-1)$ where o is the overhead and L is the latency. Whereas MPI_Reduce computes a sum and assigns it to a single process using a binomial tree.