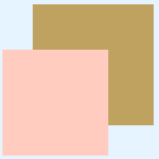
GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Assignment 2, Task 2
Parallelization So2011

Pattreeya Tanisaro

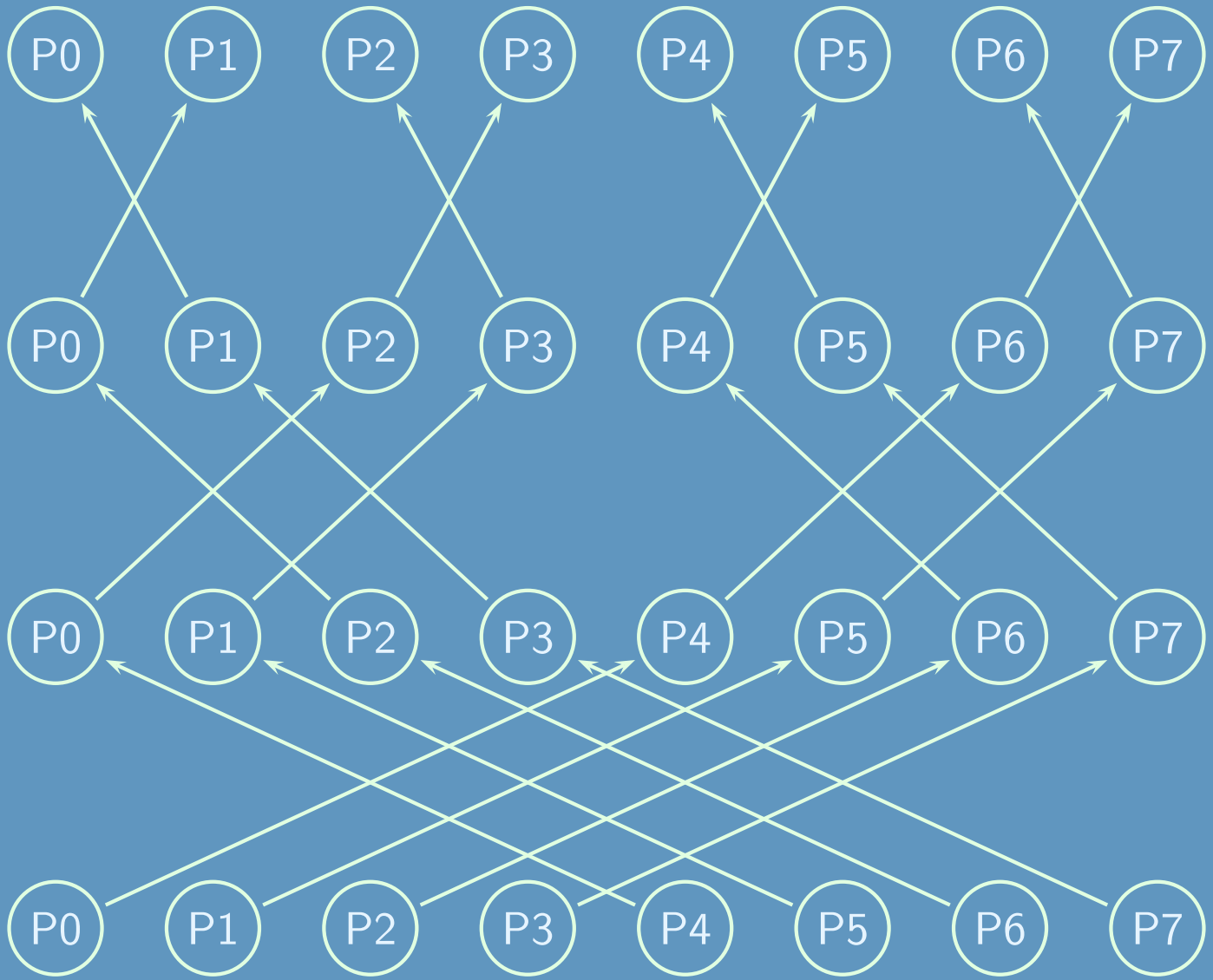
May 23, 2011

Tasks

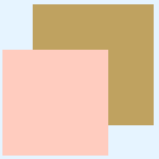


- Deploy butterfly algorithm for communication and use: (1) Standard Blocking (2) Synchronous Blocking (3) Buffered Blocking (4) Standard Non-Blocking (5) Synchronous Non-Blocking (6) Buffered Blocking
- Modify program of Task 1 to compute the following integrals $\int_{0.1}^{10000} \frac{1}{x} dx = 11.512925$ and $\int_{10}^{2000} \sin^2 x dx = 995.399112$
- Measure computation and communication time for sequential and parallel processing and compare the results

Butterfly Network



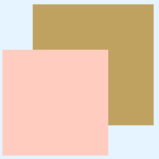
Executable - butterfly



```
$ mpiexec -np 16 -f hosts ./butterfly a 1000000 0.1 10000 Bsend
```

- The first 4 parameters are similar to Task 1. 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 e.g. Send, Bsend, Ssend, Isend, Ibsend and Issend

Executable - butterfly



```
$ mpiexec -np 16 -f hosts ./butterfly a 1000000 0.1 10000 Bsend
```

- The first 4 parameters are similar to Task 1. 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 e.g. Send, Bsend, Ssend, Isend, Ibsend and Issend

Choosing N=100000000, each (function) calculation has 6 communication types and takes 4 various number of processes, resulting in 24 calls! We calculate 2 functions, therefore 48 calls for just choosing 1 integral step size! run_butterfly.sh will perform all the works!

Performance Analysis (1)

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

	P=1			P=8			P=16		
	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec
Send	14	2043409	2043423	375	278565	278940	508	152446	152954
Ssend	13	2039038	2039051	357	277866	278223	714	153909	154623
Bsend	19	2032323	2032342	176	278253	278429	256	156379	156635
Isend	12	2026387	2026400	197	282802	282999	232	152852	153084
Issend	20	2068016	2068036	267	285806	286073	484	155812	156296
Ibsend	19	2054497	2054516	266	275318	275584	323	152932	153255

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

	P=32			P=128			P=256		
	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec
Send	4945	111271	116216	26009	65563	91572	48852	72294	121146
Ssend	2992	106199	109191	14275	63085	77360	31966	62297	94263
Bsend	4666	110285	114951	36344	59207	95551	53760	78823	132583
Isend	4600	110596	115196	36289	58295	94584	63619	66235	129854
Issend	4850	112046	116896	26173	63838	90016	51009	61748	112757
Ibsend	4385	112794	117179	26423	58583	85006	43233	78413	121646

Performance Analysis (2)

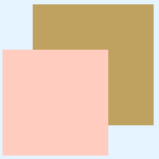
Table 3:
$f(x) = \sin^2 x$ with $N = 100000000$

	P=1			P=8			P=16		
	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec
Send	11	8672892	8672904	361	1093044	1093405	395	560006	560401
Ssend	11	8749362	8749373	307	1097052	1097359	437	561495	561932
Bsend	19	8598751	8598770	177	1096495	1096672	278	564192	564470
Isend	11	8621713	8621724	188	1098148	1098336	325	561217	561542
Issend	17	8662789	8662806	331	1106089	1106420	426	563402	563829
Ibsend	17	8856396	8856413	225	1096390	1096615	301	589187	589488

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

	P=32			P=128			P=256		
	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec	comm t μsec	calc t μsec	total t μsec
Send	10050	319285	329335	13965	157776	171741	44020	188096	232116
Ssend	2830	320564	323394	15702	158074	173776	39514	191335	230849
Bsend	4186	318293	322479	25445	211861	237307	36768	245274	282042
Isend	4318	314399	318717	26290	174015	200305	38057	189468	227525
Issend	4067	314279	318346	36241	157171	193412	45606	189429	235035
Ibsend	4137	320733	324871	35326	154281	189607	49095	191706	240801

Conclusion



- Parallel processing yields a better overall performance than sequential processing
- Total execution time is clearly decreasing as the number of processors increasing but at one point although increasing number of processors, the execution time starts to increase. Since parallel overheads start to dominate the execution time. Therefore, optimum number of processors which minimizes overall execution time should be made for an optimistic assumption
- Non-blocking operations should perform best, however in this exercise, the results do not obviously meet the expectation in all tests. Probably because the message size using in the butterfly network is quite small (only 8 bytes) so the performance gain does not standout. Besides, we may need to modfiy the way we place the algorithm. In the tests, we do not make use of the processors after finishing initialization.