Self Introduction

Dr Tay Seng Chuan
Tel: 65168752 Email: scitaysc@nus.edu.sg
Office: S16-02, Dean's Office at Level 2
URL: http://www.physics.nus.edu.sg/~phytaysc

I have been working in NUS since 1990, and I teach mainly computer and computing subjects. (Deputy Director at Centre for Remote Imaging Sensing and Processing: 2001 to 2004)

I am (was) a hands-on person on High Performance Computing. Todate I have made my hands dirty on PC-transputer System, network of workstations, and GRID System. Participated in IBM-iHPC's Blue Challenge in 2002.

Current Duties in NUS:

Day: SM2/SM3 Programme (MOE), Faculty IT Unit (Science), Physics Dept (Senior Lecturer). I am also a mentor of MOE Gifted Education Programme.

Night: Assistant Master of Temasek Hall, Resident Fellow (Block E).

1

- Consultation for CZ4102
  - Day: S16-02, Dean's Office. By email appointment or Tel at any time. It is better to give a telephone call first.
  - Night: Temasek Hall, Block E, Room E100. Give a call first.

2

- Continual Assessments
  - 2 sets of individual assignment to be graded (15% x2) – hand written form but you can also type it , and programs.
  - 1 set of group project to be graded (30%) – report form (to be typed) with programs and presentation is needed.
  - We practise honour system in the award of scores, ie, you cannot copy.

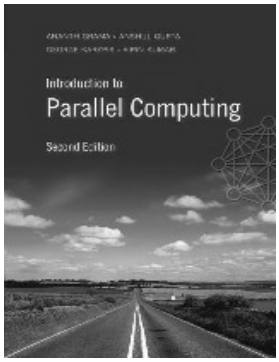
3

- Exam
  - 40% from final exam
  - Closed book.

4

Textbooks

- **Introduction to Parallel Computing (2nd Edition)**
  by Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, Addison Wesley, Second Edition, 2003, ISBN 0-201-64865-2
  (has been ordered in NUS COOP)



- MPI Manual

5

References

- **A few scientific papers**

- **The Art of Computer Programming, Volumes 1-3**
  by Donald E. Knuth, Addison-Wesley Publishing Co., October, 1998

6

- Topics
  - Introduction – From Hands-on Experience
  - Hardware Platforms – Vector/Distributed
  - S/W Platforms (threading versus message passing)
  - MPI (Message Passing Interface)
  - Scientific Computation Examples (Parallel algorithms for matrix multiplication, linear systems, sorting and merging)
  - Parallel Discrete Systems

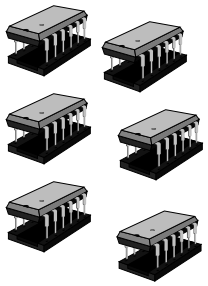
7

CZ4102 High Performance Computing

Lecture 1: Introduction – From Hands-on Experience

Dr Tay Seng Chuan

Reference: experience !!!



8

Objectives

- to appreciate the organization of a parallel processing system and the communication models used in parallel computing
- to understand the notions of speedup and efficiency and their implications

9

Objectives *(cont'd)*

- to appreciate the interacting effect of process granularity, communication overhead, load balancing and parallelization penalty
- to challenge common senses and common beliefs with regards to the fairness of workload distribution

10

Let's do some calculation…

Suppose a computation time of 1 nanosecond (supercomputer range) is expected. What is the distance traveled by an electromagnetic signal on ICs?

- An electromagnetic signal at free space travels at 299792458m/s (~ 3 x 10^8 m/s).
- The speed of a computer is inversely proportional to the transmission delay of electrical signals on the ICs.

11

Answer :

Since distance = speed x time,

the distance traveled is

3 x10^8 x 10^{-9} = 0.3 m or 30 cm.

12

High Performance Computing : *The Motivation*

The distance calculated is reduced by a factor of 2 to 3 in many materials used to build computer. As such the distance traveled is reduced to 15 cm or less (dimension of PDA). Can we find a PDA (Personal Digital Assistant) of this size and yet is able to achieve the supercomputer performance?

13

Parallel Computing : *The Motivation (cont'd)*

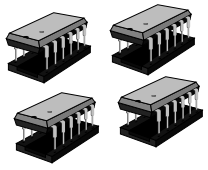
Answer : Not optimistic.

How to resolve the cooling problem as the ICs are so lose to each other?

The marginal improvement will be progressively expensive.

Alternative :

High Performance Computing.



14

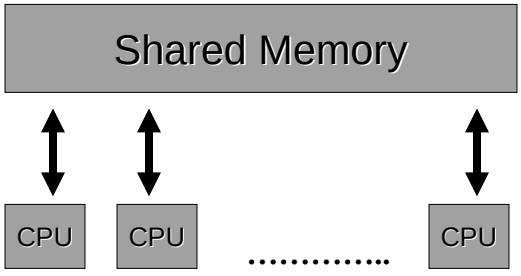
Parallel Computing System

A parallel computing system is a platform that contains a collection of processing elements which can communicate and cooperate to solve large problems fast.

15

Communication Models of Parallel Computing System

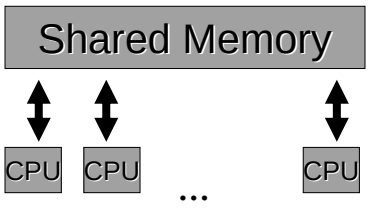
1. Shared Memory



16

Shared Memory

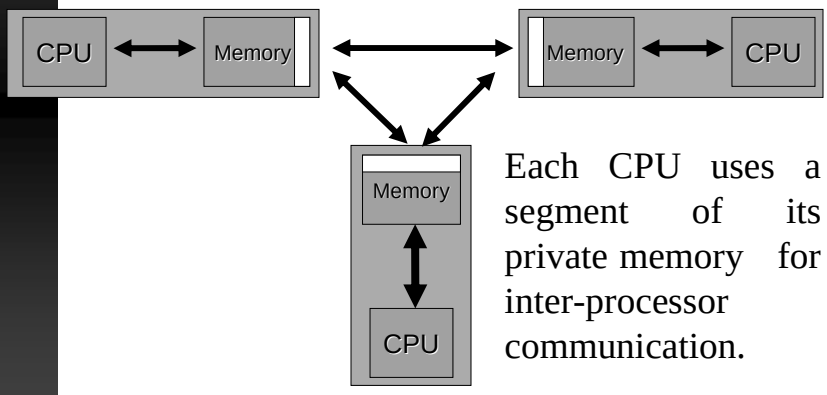
The shared memory configuration is not scalable as the access to the same memory location may need to be sequentialized so that data integrity can be assured. Sequentialization is equivalent to the program execution on a uniprocessor.



17

Communication Models of Parallel Computing System *(cont'd)*

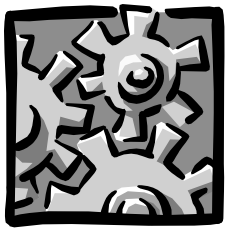
2. Private Memory



Each CPU uses a segment of its private memory for inter-processor communication.

18

Performance Metrics



Speedup

Speedup *(S)* is the ratio between the time needed for the most efficient sequential algorithm *(T_1)* to perform a computation, and the time needed to perform the computation on a parallel machine incorporating parallelism *(T_p)* where *p* processing elements, *p >1*, are used.

19

Performance Metrics *(cont'd)*

Speedup

$$S(p) = \frac{T_1}{T_p}$$

S(p) = *p* means a linear speedup

S(p) < *p* means a sub-linear speedup

S(p) > *p* means a super-linear speedup

S(p) < *1* means a slow down

20

Performance Metrics *(cont'd)*

Efficiency (E)

$$E(p) = \frac{S(p)}{p} \times 100\%$$

21

Example: A Search Problem

In the following grid cells there is one negative number. What is the time required to find the number?

4	8	7	5
2	12	10	6
1	16	15	9
3	11	13	-7

22

Search Problem *(cont'd)*

4	8	7	5
2	12	10	6
1	16	15	9
3	11	13	-7

Let c be the time required to process a cell, and a top down search on the columns is adopted.

When 1 processor is used, $T_1 = 16c$.

When 4 processors are used and assume no overhead, $T_4 = 4c$.

23

Search Problem *(cont'd)*

Speedup and Efficiency

$$S(4) = \frac{T_1}{T_4} = \frac{16c}{4c} = 4$$

$$E(4) = \frac{S(4)}{4} \times 100\% = 100\%$$

24

Speculation :

Can speedup be greater than *p* (or can efficiency be greater than 100%)?

Given a sequential algorithm (G1) that incurs the least runtime (T_1) for a problem. Suppose its parallel version is able to achieve a super-linear speedup. We have

$$S(p) = \frac{T_1}{T_p} > p$$

$$T_p < \frac{T_1}{p}$$

25

Speculation *(cont'd)*

We now construct a sequential algorithm *(G2)* by sequentializing the parallel algorithm. The runtime of *G2* on one processor is

$$\underbrace{T_p + T_p + ... + T_p}_{\text{p terms}} < \frac{T_1}{p} \times p$$

26

Speculation *(cont'd)*

This is an anomaly since *G1* is the best sequential algorithm (of the least runtime) but now the runtime of *G2* is less than T_1.

Therefore, a super-linear speedup cannot be guaranteed. But it can happen if the condition is right.

27

Instance for Super-linear Speedup

Consider the grid search again. What is the speedup if the grid contents are:

4	8	7	5
2	12	10	-7
1	16	15	9
3	11	13	6

$$T_4 = 2c$$

$$S(4) = \frac{14c}{2c} = 7 > 4$$

28

Overheads incurred in Parallel Computing

Communication Overhead and Computation Granularity

Communication Overhead = $\alpha + n \times \beta$

α = channel setup time

$n \times \beta$ = transmission time, where

n is the number of bits, and

β is time required to transmit one unit of data

29

Overheads incurred in Parallel Computing *(cont'd)*

Consider a square grid with heap property

1	5	7	8
2	6	10	12
3	9	13	14
4	11	15	16

father → son

↓

son

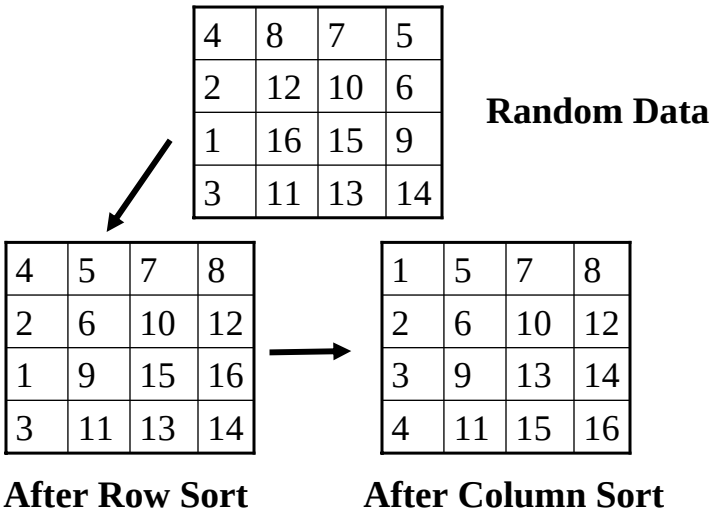
The usefulness of heap property is that the smallest data on the square grid or sub-grids can be located at O(1) time.

30

Overheads incurred in Parallel Computing *(cont'd)*

Give a square grid filled with random data. The heap property can be established by applying row sort and column sort.

31

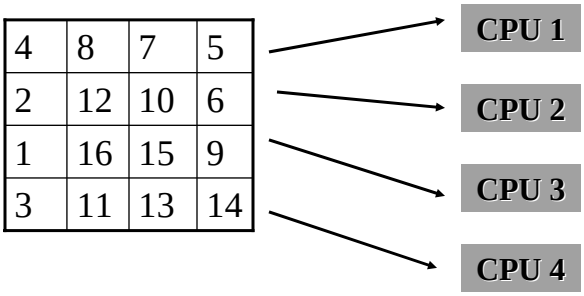


Suppose the time required for each row sort or column sort is *s*. We have $T_1 = 4s + 4s = 8s$.

32

Overheads incurred in Parallel Computing *(cont'd)*

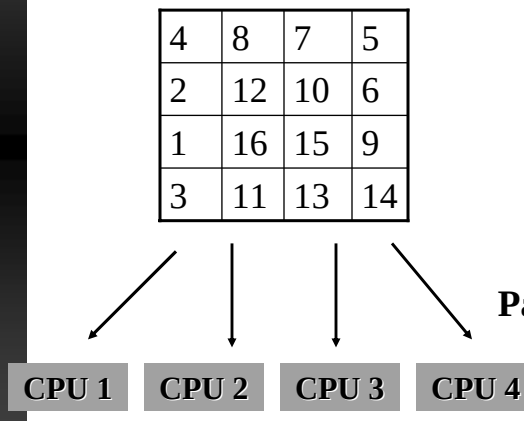
What if 4 processors are used?



Parallel Row Sort

33

Overheads incurred in Parallel Computing *(cont'd)*



Parallel Column Sort

34

Overheads incurred in Parallel Computing *(cont'd)*

$$T_p = (\alpha + n\beta) + s + (\alpha + n\beta) + (\alpha + n\beta) + s + (\alpha + n\beta)$$
$$= 2s + 4(\alpha + n\beta)$$

35

Overheads incurred in Parallel Computing *(cont'd)*

The communication overhead should be treated as a relative term with respect to computation granularity. Why?

$$S(p) = \frac{8s}{2s + 4(\alpha + n\beta)}$$

If $s >> \alpha + n\beta$, $S(p) \approx 4$.

(What is $2000 if you already have $1,000,000!!)

Otherwise *S(p)* is sub-linear, and in the worst case it can become a slowdown.

36

Load Balancing

Should workload be evenly (equally or fairly) distributed?

- Common belief : Yes.
- My answer : Not always yes if our objective is to minimize the runtime of a parallel program.
- It has been analytically and experimentally established that an even workload distribution may aggravate communication overhead, resulting in a longer program execution time.

37

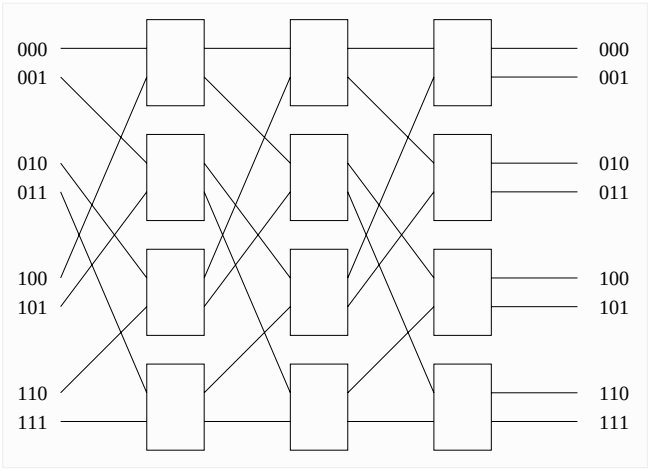
Load Balancing *(cont'd)*

- It has also been shown (analytically and experimentally) that an unbalanced workload distribution may outperform an even workload distribution scheme. The cause for such an unusual outcome is that the reduction in communication overhead of the unbalance scheme exceeds the increase in computation time.

38

Load Balancing *(cont'd)*

Example: A Parallel Simulation of Multi-State Interconnection Networks (MIN)



39

Load Balancing *(cont'd)*

Why we do this simulation

- MIN can be used to connect the processors in multiprocessor platforms, or to connect processors to memory modules.
- More recently, MIN is used as intercommunication structure for the switch nodes for high-speed communication networks such as broadband Integrated Services Digital Networks (B-ISDN), and Asynchronous Transfer Mode (ATM).

40

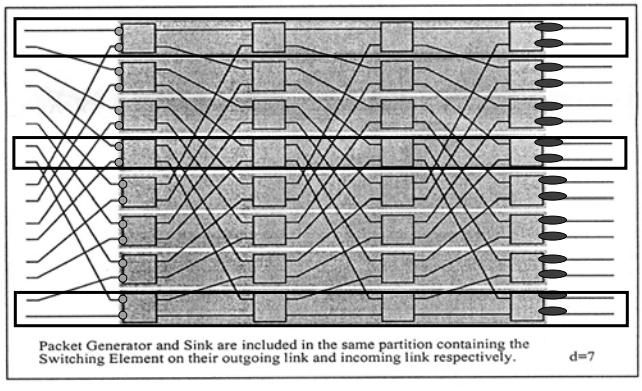
Load Balancing *(cont'd)*

Three load-balancing schemes are used in MIN simulation :

- Horizontal Partitioning Scheme (Balance)
- Vertical Partitioning Scheme (Unbalance)
- Modular Partitioning Scheme (Unbalance)

41

Horizontal Partitioning Scheme

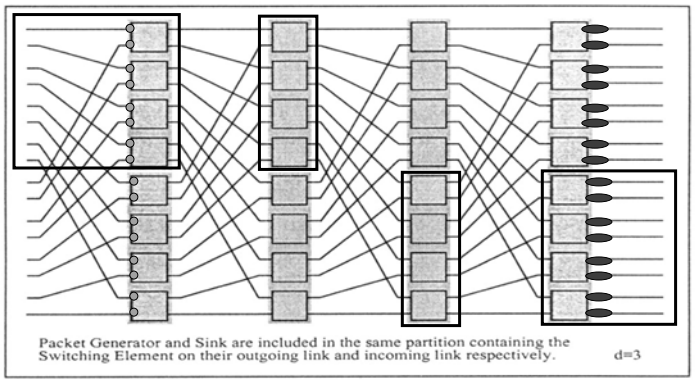


: Packet Generator

: Sink

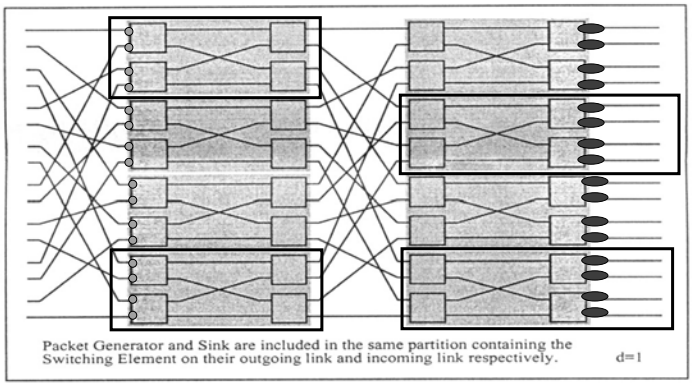
42

Vertical Partitioning Scheme



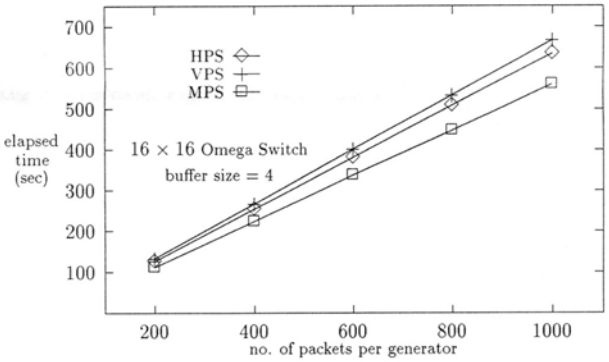
43

Modular Partitioning Scheme



44

Comparison of Runtimes



Run Time for Different Partitioning Schemes

The runtime comparison shows that the modular partitioning scheme incurs the least program runtime.

45

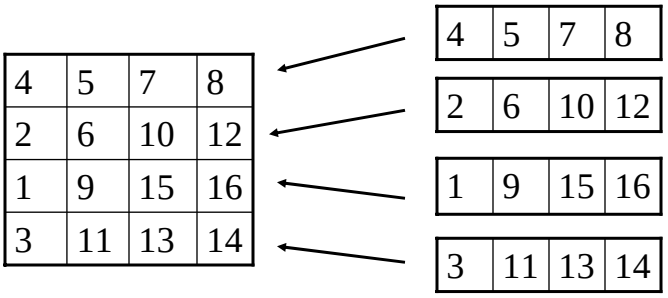
Algorithm Penalty

The penalty refers to the additional processing required due to the parallelization of a sequential algorithm.

E.g., the consolidation of grid contents after row sort is performed is a penalty of the parallel version of the heap construction algorithm.

46

Algorithm Penalty *(cont'd)*



Algorithm penalty is of O(n) for the grid point consolidation.

47

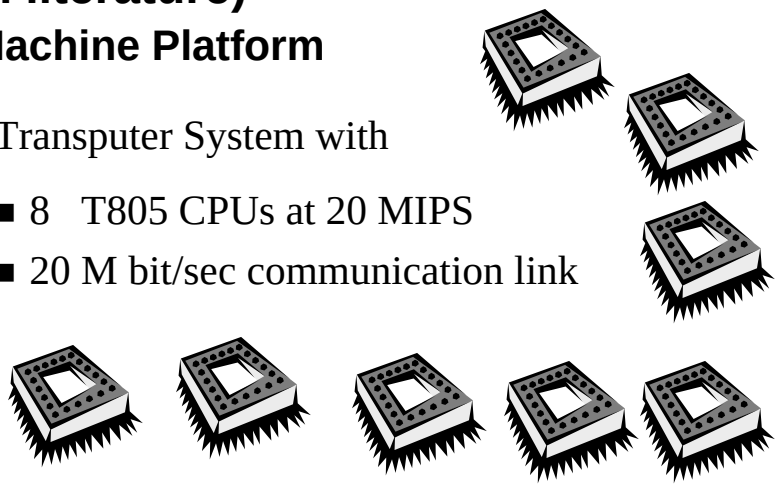
High Performance Computing : To do or not to do?

- High Performance Computing introduces a new set of problems which does not exist in the sequential version. This will be discussed in CZ4102.
- To achieve a good speedup, the interacting effect of process granularity, communication overhead, load balancing, and parallelization penalty must be considered. This will be discussed in CZ4102.
- There are many HPC applications that have achieved a close-to-linear speedup.

48

Example: Parallel Block Sort (not in literature)

Machine Platform



Transputer System with

- 8 T805 CPUs at 20 MIPS
- 20 M bit/sec communication link

49

Parallel Block Sort (cont'd)

6	**26**	**25**	**18**	**3**	**45**	**49**	**5**
32	**2**	**63**	**56**	**36**	**59**	**15**	**7**
41	**29**	**54**	**22**	**46**	**12**	**10**	**60**
61	**52**	**28**	**34**	**62**	**23**	**30**	**51**
33	**42**	**53**	**11**	**8**	**38**	**48**	**9**
17	**57**	**21**	**58**	**47**	**64**	**55**	**35**
16	**27**	**43**	**37**	**1**	**20**	**44**	**4**
13	**39**	**14**	**24**	**19**	**31**	**40**	**50**

Random Data

50

Parallel Block Sort *(cont'd)*

1	2	4	6	10	11	12	14
3	5	13	15	18	20	21	26
7	8	16	22	23	25	27	30
9	17	19	24	28	29	31	40
32	34	35	36	38	39	43	44
33	37	42	45	46	48	50	53
41	47	49	51	52	54	56	59
55	57	58	60	61	62	63	64

Preprocessed Data (Heap property is established)

51

Parallel Block Sort *(cont'd)*

1	2	4	6	10	11	12	14	3	5	13	15	7	8	16	9
18	20	21	26	22	23	25	27	30	17	19	24	28	29	31	32
40	34	35	36	38	39	43	44	33	37	42	45	46	48	41	47
50	53	49	51	52	54	56	59	55	57	58	60	61	62	63	64

Sub-interval = [1,16]

Sub-interval = [17,32]

Sub-interval = [33,48]

Sub-interval = [49,64]

Dividing Preprocessed Data into 4 Blocks

52

Parallel Block Sort *(cont'd)*

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Sorted Data

53

Speedup of Parallel Block Sort vs. Parallel Quick Sort

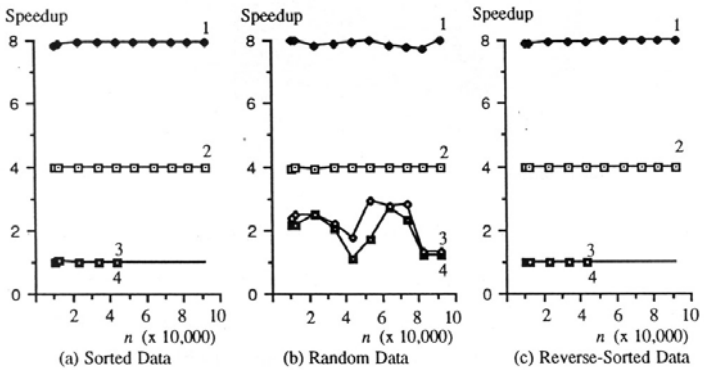


Figure 5 The speedup of PBS and PQS on 4 and 8 processors
Legends : 1:PBS (p=8) 2:PBS (p=4) 3:PQS (p=8) 4:PQS (p=4)

54

Conclusions

- 😐 The speed of the traditional computer cannot be increased indefinitely. A parallel platform offers the potential to reduce program runtime.
- 😐 Parallel computing also incurs communication overhead and parallelization penalty.
- 😐 An even workload distribution scheme may not result in the least runtime.

55

Conclusions *(cont'd)*

- ☺ To achieve a good speedup, the interacting effects of process granularity, communication overhead, load balancing, and parallelization penalty must be considered. You got to know how to do it, and I will teach you in this course.
- ☺ **Good speedup has been achieved in many HPC applications. That's why CZ4102 is not a fairy tale.**

56