School of Computer Science and Applied Mathematics
UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

COMS3008A: Parallel Computing
Exercise 1 & 2

2021-8-17

Objectives

This exercise covers the topics we learnt during Week 1 (Lec1 slides) and Week 2 (Lec2 slides), as well as self-study additional material presented in Section 2. The objectives are to understand and apply the concepts and discussions on

- Classification of parallel computers
- Understand the simple metrics used to evaluate the performance of a parallel program
- Understand the similarities and differences between Amdahl's Law and Gustafson's Law
- Apply the basic performance metrics, Amdahl's Law, and Gustafson's Law for simple problems.
- Theoretical modelling of parallel computation
- The basic properties and topologies of common interconnection networks
- Limitations of memory system performance

1 Problems

1. In the discussion of parallel hardware, we used Flynn's taxonomy to identify parallel systems: SISD, SIMD, MIMD, SPMD, and MISD. Explain how would each of these parallel computer works, and give some examples of such systems.

2. Of the three PRAM models, EREW-PRAM, CREW-PRAM, and CRCW-PRAM, which one has the minimum concurrency, and which one has the maximum power? Why?

3. Assume p processing elements (PEs) share a bus to the memory, each PE accesses k data items, and each data access takes time t_{cycle}.

(a) What is the maximum execution time for each PE to access the k data items?

(b) Assume each PE now has its own cache, and on average, 50% of the data accesses are made to the local memory (or cache in this case). Further, let's assume the access time to local memory is the same as to the global memory, i.e., t_{cycle}. In such case, what is the maximum execution time for each PE to access k data items?

4. Suppose that 70% of a program execution can be sped up if the program is parallelized and run on 16 processing units instead of one. By Amdahl's Law, what is the maximum speedup we can achieve for this problem if we increase the number of processing units to 32, then to 64, and then to 128?

5. For a problem size of interest, 6% of the operations of a parallel program are inside I/O functions that are executed on a single processor. What is the minimum number of processors needed in order for the parallel program to exhibit a speedup of 10?

6. An application executing on 64 processors requires 220 seconds to run. Benchmarking reveals that 5% of the time is spent executing serial portions of the computation on a single processor. What is the scaled speedup of the application? (Answer: 60.85.)

7. A company recently purchased a powerful machine with 128 processors. You are tasked to demonstrate the performance of this machine by solving a problem in parallel. In order to do that, you aim at achieving a speedup of 100 in this application. What is the maximum fraction of the parallel execution time that can be devoted to inherently sequential operations if your application is to achieve this goal?

8. Both Amdahl's Law and Gustafson's Law are derived from the same general speedup formula. However, when increasing the number of processors p, the maximum speedup predicted by Amdahl's Law converges to a certain limit, while the speedup predicted by Gustafson's Law increases without bound. Explain why this is so.

9. How do you label the nodes in a d-dimensional hypercube? Construct a 4-dimensional hypercube with labels, where there is a communication link between two nodes when their labels differ in only one binary bit.

10. (a) Partition a d-dimensional hypercube into two equal partitions such that

    i. each partition is still a hypercube;

    ii. neither partition is a hypercube.

    (b) Derive the bisection width for each of the two cases in Item 10a.

11. Derive the diameter, bisection width, and cost for each of the following interconnection network. Show your derivation. Assume there are p number of nodes where applicable.

    • Fully connected network

- Bus
- Ring
- 2D mesh
- 2D torus
- 3D mesh
- Hypercube
- Fully connected crossbar network

12. Using the basic interconnection network topologies we studied in the class, design, as many as you can, network topologies that can connect p number of PEs to m number of memory modules, so that each topology should provide a path between any pair of a PE and a memory module.

13. Complete the following exercises after reading Section 2.

    (a) Consider a memory system with a DRAM of 512MB and L1 cache of 32KB with the CPU operating at 1GHz. The $l_{DRAM} = 100$ns and $l_{L1} = 1$ns (l represents the latency). In each memory cycle, the processor fetches 4 words. What is the peak achievable performance of a dot product of two vectors? (Answer: 40MFLOPS.)

    (b) Consider an SMP with a distributed shared-address-space. Consider a simple cost model in which it takes 1 ns to access local cache, 100 ns to access local memory, and 400ns to access remote memory. A parallel program is running on this machine. The program is perfectly load balanced with 80% of all accesses going to local cache, 10% to local memory, and 10% to remote memory. What is the effective memory access time of one processor for this computation (Answer: 50.8 ns)? If the computation is memory bound, what is the peak computation rate?

2 Limitations of memory system performance

The effective performance of a program on a computer relies on the speed of the processor but also on the ability of the memory system to feed data to the processor.

1. **Latency:** A memory system, possibly with multiple levels of cache, takes in a request for a memory word and returns a block of data of size b containing the requested word after $l$ns. Here, l is referred to as the latency of the memory.

2. Effect of memory latency on performance

   Example 2.1 *Consider a processor operating at 1GHz (1ns clock) connected to a DRAM with a latency of 100ns (no caches). Assume that the processor has two multiply-add units* *and is capable of executing four instructions in each cycle of 1ns. The peak performance rating is therefore 4GFLOPS. However, since the memory latency is 100ns, and if the block size is only one word (4 bytes), then every time a memory request is made, the processor must wait 100ns (latency) before it can process a word. In the case of each floating point operation requires one data fetch, the peak speed of this computation is limited to one floating point operation every 100ns, or a speed of 10MFLOPS, which is only a small fraction of the peak processor performance.*

3. Improving effective memory latency using caches: Caches are used to address the speed mismatch between the processor and memory. Caches are a smaller and faster memory that is placed between the processor and the DRAM. The data needed by the processor is first fetched into cache. All subsequent accesses to data items residing in the cache are serviced by the cache. Thus, in principle, if a piece of data is repeatedly used, the effective latency of this memory system can be reduced by the cache.

4. **Cache hit ratio:** The fraction of data references satisfied by the cache is called the cache hit ratio of the computation on the system.

5. **Memory bound computation:** The effective computation rate of many applications is bounded not by the processing rate of the CPU, but by the rate at which data can be pumped into the CPU. Such computations are referred to as being memory bound. The performance of memory bound program is critically impacted by the cache hit ratio.

   Example 2.2 *We still consider the 1GHz processor with a 100ns latency DRAM as in Example 2.1. In this case, let's consider adding a cache of size 32KB* ($1K = 2^{10}$) *with a latency of 1ns. We use this setup to multiply two matrices A and B of dimensions* 32×32 *(for this, the input matrices and output matrix can all fit in the cache). Fetching the two input matrices (2K words) into the cache takes approximately 200*μs*. The total floating point operations in multiplying the two matrix is 64K (multiplying two* $n \times n$ *matrices takes* $2n^3$ *multiply and add operations), which takes approximately 16K cycles, i.e., 16Kns (or 16*μs*) at 4 instructions per cycle. So, the total time for the computation is* $200\mu s + 16\mu s = 216\mu s$*. This corresponds to a peak computation rate of* $64K/216\mu s \approx 303$*MFLOPS. Compared to Example 2.1, this is a much better rate (due to cache), but is still a small fraction of the peak processor performance (4GFLOPS).*

6. **Temporal locality:** The notion of repeated reference to a data item in a small time window is called temporal locality of reference. Data reuse is critical for cache performance because if each data item is used only once, it would still have to be fetched once per use from the DRAM, and therefore the DRAM latency would be paid for each operation.

7. **Bandwidth:** The rate at which data can be moved between the processor and the memory. It is determined by the bandwidth of the memory bus as well as the memory units.

Example 2.3 *We continue with Examples 2.1 and 2.2. Now consider increasing the block size to 4 words (memory bandwidth in this case; also, if a memory request returns a contiguous block of 4 words, the 4-word unit is also considered as a cache line). Now fetching the two input matrices into the cache takes* $200/4 = 50\mu s$*. Consequently the total time for the computation is* $50\mu s + 16\mu s = 66\mu s$*, which corresponds to a peak computation rate of* $64K/66\mu s \approx 993MFLOPS$*.*

8. Increasing the memory bandwidth, or building wide data bus connected to multiple memory banks, is expensive to construct. In a more practical system, consecutive words are sent on the memory bus on subsequent bus cycles after the first word is retrieved. For example, with a 32-bit data bus, the first word is put on the bus after 100ns (the latency) and one word is put on each subsequent bus cycle.Then the 4 words will become available after $100 + 3 \times (\text{memory bus cycle})$ns. Assuming a data bus operating at 200MHz, this adds 15ns to the cache line access time.
9. **Spatial locality:** Successive computations require contiguous data. If the computation (or data access pattern) in a program does not have spatial locality, then effective bandwidth can be much smaller than the peak bandwidth.

Bibliography

1. Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-Art Platforms, by Roman Trobec, Boštjan Slivnik, Patricio Bulić, Borut Robič, Springer, 2018,
2. Introduction to Parallel Computing, second edition, by Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar. Addison Wesley Publisher, 2003. (https://www-users.cs.umn.edu/~karypis/parbook/)