**CECS 570 – Assignment 1**

1. A sequential application with a 20% part that must be executed sequentially, is required to be accelerated three-fold. How many CPUs are required for this task? If the required speedup was 5, what would be the number of CPUs required?

**Solution:**

S(p) = 3; (1-α) = 20% = 0.2; therefore, α = 0.8;

By Amdahl’s law, S(p) = 1/ (1-α(1-1/p));

Therefore, 3 = 1/ (1-0.8(1-1/p));

1-0.8(1-1/p) = 1/3 = 0.333;

Therefore 0.8(1-1/p) = 0.667;

1-1/p = 0.667/0.8 = 0.833;

1/p = 1-0.833 = 0.167

P = 1/0.167 = 5.99

**Therefore, to be accelerated three-fold, approximately 6 CPU’s are required.**

S(p) = 5; (1-α) = 20% = 0.2; therefore, α = 0.8;

By Amdahl’s law, S(p) = 1/ (1-α(1-1/p));

Therefore, 5 = 1/ (1-0.8(1-1/p));

1-0.8(1-1/p) = 1/5 = 0.2;

Therefore 0.8(1-1/p) = 0.8;

1-1/p = 0.8/0.8 = 1;

1/p = 1-1 = 0

P = ∞

**Therefore, to be accelerated three-fold, approximately ∞ CPU’s are required.**

1. An application with a 5% non-parallelizable part, is to be modified for parallel execution. Currently on the market there are two parallel machines available: machine X with 4 CPUs, each CPU capable of executing the application in 1 hour on its own, and, machine Y with 16 CPUs, with each CPU capable of executing the application in 2 hours on its own. Which is the machine you should buy, if the objective is to minimize the execution time?

**Solution:**

(1-α) = 5% = 0.05; therefore, α = 0.95;

For X,

ts = 60 mins.; p = 4

By Amdahl’s law, S(p) = 1/ (1-α(1-1/p)) = 1/ (1-0.95(1-1/4)) = 1/ (1-0.95(0.75)) = 1/ (1-0.713) = 1/0.288 = 3.478

S(p) = ts/ tp ;

Therefore, tp = ts /S(p) = 60/3.478 = **17.25 minutes … (1)**

For Y,

ts = 120 mins.; p = 16

By Amdahl’s law, S(p) = 1/ (1-α(1-1/p)) = 1/ (1-0.95(1-1/16)) = 1/ (1-0.95(0.938)) = 1/ (1-0.891) = 1/0.109 = 9.143

S(p) = ts/ tp ;

Therefore, tp = ts /S(p) = 60/9.143 = **13.125 minutes … (2)**

**Therefore, by looking at (1) and (2), I would buy machine Y.**

1. A climate model requires 1016 floating point operations for a ten-year simulation. How long would this computation take at 1Gflops?

**Solution:**

Flops = 1016; CPU = 1Gflops = 109 flops/sec;

Therefore, Time = Flops/CPU = 1016 /109 = 107 seconds = 115.741 days.

**At 1Gflops this would take 115.741 days for computation.**

1. A climate model generates 1011 bytes of data in a ten-day simulation. What transfer rate is required if we are to search this data in ten minutes?

**Solution:**

Data size = 1011; Time to search = 10 mins. = 600 secs.;

Therefore, Transfer rate = Data/Time = 1011 / 600 = 1.667 \* 108 bytes/sec = 166.667 Mbps.

**Therefore, transfer rate of 166.667 Mbps is required if we are to search this data in ten minutes.**

1. Suppose a sequential widget-assembly machine can assemble one widget every 3 seconds, and a 3-way data parallel widget-assembly machine is capable of producing three widgets every three seconds.
   1. How much time does it take to produce 100 widgets using (i) the sequential machine, and (ii) the parallel machine? What is the speedup?
   2. How many widgets must the parallel machine assemble in order to achieve a speedup of 3 over the sequential machine? Justify your answer.

**Solution:**

Sequential machine requires (3 sec/widget) \* 100 widgets = **300 seconds**.

Parallel machine requires (1 sec/widget) \* 100 widgets = **102 seconds**.

Therefore, Speedup S(p) = ts/ tp = 300/102 = **2.941 … (a)**

**From (a) it is clear that widgets of multiples of three can be assembled by the parallel machine in order to achieve a speedup of 3 over the sequential machine.**

1. A parallel application running on 5 identical processors, has a 10% sequential part. What is the speedup relative to a sequential execution on a single processor? If we would like to double that speedup, how many CPUs would be required?

**Solution:**

p = 5; (1-α) = 0.1; therefore, α = 0.9;

According to Gustafson-Barsis’s Rebuttal of Amdahl’s Law: S(p) = (1-α) + p\*α

Therefore, S(p) = 0.1 + 5\*0.9 = 0.1 + 4.5 = **4.6 … (1)**

For S(p) = 2\*4.6 = 9.2,

9.2 = 0.1 + p\*0.9

Therefore, p\*0.9 = 9.1

Therefore, p = 9.1/0.9 = **10.111 … (2)**

**From (1) and (2) we know that, the speedup relative to a sequential execution on a single processor is 4.6 and if we would like to double that speedup, approximately 11 CPUs would be required.**

1. A parallel application running on 10 CPUs, spends 15% of its total time, in sequential execution. What kind of CPU (i.e., how much faster) would we need to run this application completely sequentially, while keeping the same total time?

**Solution:**

p = 10; (1-α) = 0.15; therefore, α = 0.85;

According to Gustafson-Barsis’s Rebuttal of Amdahl’s Law: S(p) = (1-α) + p\*α

Therefore, S(p) = 0.15 + 10\*0.85 = 0.15 + 8.5 = **8.65**

**The speedup factor indicates that the sequential CPU must be 8.65 times faster than the CPU’s in parallel application.**

1. Is it possible to have a system efficiency (E) of greater than 100%? Discuss

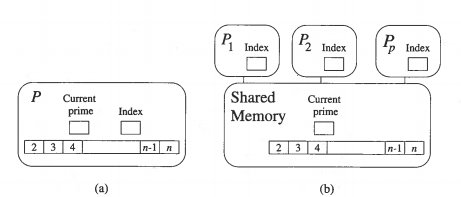
**Solution:**

* Yes, it is possible to have a system efficiency (E) of greater than 100%.
* i.e. S(p)/p > 1 i.e. S(p) > p.
* It is called *Super Linear Speedup.*
* **Example 1:** Super linear speedup in low level computations can be achieved through the “Cache effect”. A single processor will have the computation saved on the secondary storage due to its size, but if we implement enough parallel processors that the whole computations can be split into the corresponding portions and can be fit into the caches of the parallel processors. The difference in the memory hierarchy i.e. secondary storage to cache results in super linear behavior.
* **Example 2:** When searching large datasets such as genomic data searched by BLAST implementations. Hierarchical difference from Secondary memory to RAM of individual nodes results in a drastic reduction of time required for mpiBLAST to search it.
* **Example 3:** If backtracking is performed in parallel, an exception in one thread can cause several other threads to backtrack early, before they reach the exception themselves.
* **Example 4:** In parallel implementations of branch and bound for optimization. The processing done by a processor for one node can affect the processing that needs to be done for the other nodes.

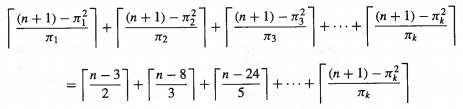
**Reference:** <https://en.wikipedia.org/wiki/Speedup>

1. Determine the maximum speedup that can be achieved with the control parallel approach (i.e., functional decomposition) for the prime finding solution for n = 1000.

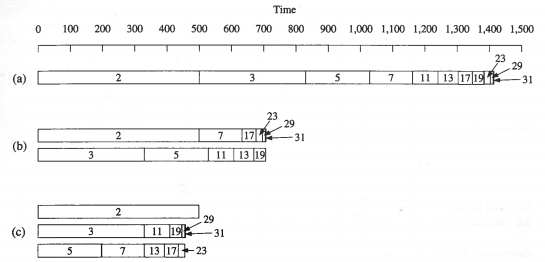
**Solution:**

****

* The above figure shows Shared memory model for parallel implementation of Sieve of Eratosthenes algorithm. Fig(a) shows sequential algorithm that maintains array of prime numbers, variable storing current prime number and variable to store index of loop iterating through the array. Fig(b) shows parallel implementation that stores private loop index for each processor to work independently. It also shares access to other variables in other processors.
* The formula for sequentially calculating the time for processing is a follows. Assume it takes 1 unit of time for a processor to mark a cell. The total number of prime numbers for 1000 numbers will be √1000 = 31. Therefor the total time for a single processor striking out all composite numbers in 1000 numbers will be 1411.



* The below diagram shows how adding processors reduces the execution time of the parallel algorithm when n = 1000. The number in the bar shows the prime number whose multiples have been marked. The length of the bar shows the time to complete the marking for the given prime number.



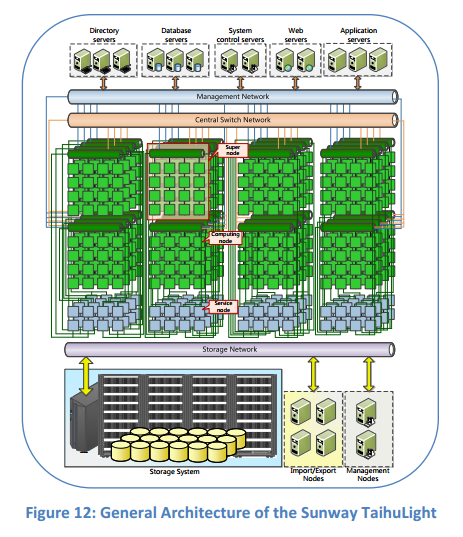
* As the diagram shown, single processor strikes all composites in 1411 units. With two processors the time drops to 706 units. With three processors the time drops to 499 units which is the time to process striking down composites for prime number 2.
* Any further addition of processors would not be lower than that as anyhow 1 processor will always take 499 units.
* **Therefore, maximum S(p) = 1411/499 = 2.828.**

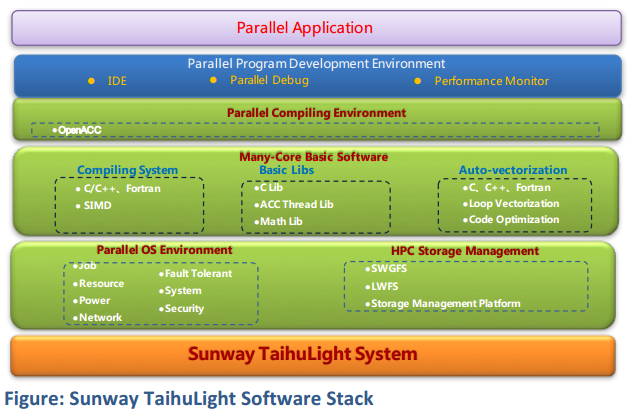
**Reference:** Parallel computing Book by Michael J. Quinn

1. Look up for one of the top 10 most powerful supercomputers in the world, and write what you can find about it.

**Solution:**

* **Name**: Sunway TaihuLight System
* **Developed by**: National Research Center of Parallel Computer Engineering & Technology (NRCPC)
* **Location**: National Supercomputing Centre in Wuxi.
* **CPU**: Shenwei-64
* **Processor**: The SW26010 processor, that was designed by the Shanghai High Performance IC Design Center.
* **Theoretical peak performance**: 125.4 Pflop/s. Performs 93,000 trillion calculations per second.
* **Application domain**:
  + Advanced manufacturing: CFD, CAE applications
  + Earth system modeling and weather forecasting
  + Life Science
  + Big data analytics.
* **Number of cores**: 10,649,600.
* **Primary Memory**: 1.31 PB.
* **Number of nodes**: 40,960
* **Operating System**: Linux-based
* **The peak power consumption under load**: 15.371 MW or 6 Gflops/W.
* **Cooling system**: Closed-coupled chilled water cooling with a customized liquid water-cooling unit.
* **Cabinets**:40.
* **Nodes per cabinet**: 1024 nodes
* **Cores per node**:260 cores

****

****

**Reference:** Report on the Sunway TaihuLight System, Jack Dongarra, University of Tennessee, Oak Ridge National Laboratory