Multiprocessing

Architectures and Performance Measures Problems

Dr. Raul Ramirez-Velarde

Student: Gerardo Mauricio Gutierrez Quintana A00815174

**ALGORITHM COMPLEXITY**

1. In order to balance load for parallel bucket sort of n elements, uniformly spaced splitters need to be selected (distribution functions). This can be done by first dividing the list into B lists and choosing B equi-spaced samples from each. The final B splitters are chosen uniformly spaced from these samples. How balanced are the buckets if these splitters are used? This is to say, that if a bucket will contain elements xi to xj, the sample will be any of those elements.

Example, if numbers can have values from 1 to 100 and there are 4 lists, there will be a sample between 1 to 24, one between 25 to 49, one from 50 to 74 and one fro 74 to 100. For example, 10, 31, 58, and 80 were taken as samples. These numbers will be used as splitters (distribution functions):

Numbers between 1 and 10, bucket 1

Numbers between 11 and 31, bucket 2

Numbers between 32 and 58, bucket 3

Numbers between 59 and 80, bucket 4

Numbers between 81 and 100, bucket 5

**Answer:** All the buckets will be equally balanced but, there is an exception in which the N elements are not divisible. On this scenario, the buckets can’t be balanced.

1. This is the pseudo code for merging two ordered lists:

**algorithm** merge(A, B) **is**

**inputs** A, B : list

**returns** list

C := new empty list

**while** A is not empty and B is not empty **do**

**if** head(A) ≤ head(B) **then**

append head(A) to C

drop the head of A

**else**

append head(B) to C

drop the head of B

*// By now, either A or B is empty. It remains to empty the other input list.*

**while** A is not empty **do**

append head(A) to C

drop the head of A

**while** B is not empty **do**

append head(B) to C

drop the head of B

**return** C

How fast can **two** sorted lists of size n each be merged into one using p processors?

**Answer:** Two sorted lists of size n that are using p processors will be as fast as n/p times.

**SPEED UP**

1. Suppose one wants to determine if it is advantageous to develop a parallel version of a certain sequential application. Through experimentation, it was verified that 90% of the execution time is spent in procedures that may be parallelizable.
2. What is the maximum speedup that can be achieved with a parallel version of the problem executing on 8 processors?

**Answer:**

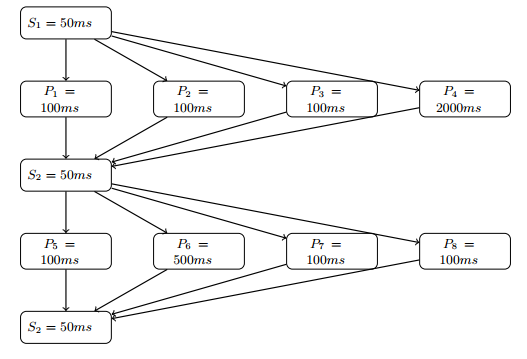
1. What is the limit of the maximum speedup that can be attained?

**Answer:** 10

1. Suppose the best sequential algorithm for a problem has complexity for n data items.
2. What is the minimum fraction of parallelizable code required to have a speed up of at least 5 using n2 processors?

**Answer:**

1. Examine the task graph shown below. Each task, which you can think of as an asynchronous task, is labelled with its runtime. Answer the following four questions about the program’s runtime. In all cases you may ignore any work scheduling or task spawning overheads



1. Assuming a single worker thread what is the runtime of this program?

**Answer:** As we summarize all the times, we get **3250 ms**

1. What is the speedup when there are 8 processors?

**Answer:** As the stages are now running in parallel, we take the slowest task as reference, getting the next:

And the speedup is:

1. If each parallel task were parallelized further to become two parallel tasks (exactly halving the execution time), and again run with 8 processors, what would the runtime be? What is the speedup be relative to the previous run?

**Answer:** As the stages are now running in parallel, we take the slowest and divided by two as they are twice times faster, getting the next:

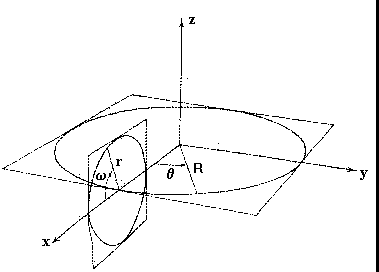
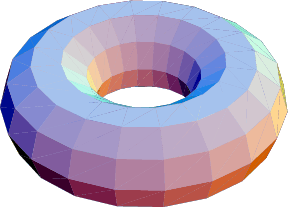
And the speedup is:

1. Why is the speedup not greater than 2?

**Answer:** It is because only a part of the program suffers from a speedup and not the entire program.

**ARCHITECTURE**

1. Assume a computational torus is built using k nodes for both circles (see images). What would be the number of nodes, the diameter, bisection width and edges per node?



**Answer:**

**# of nodes: (k)(k)**

**Diameter:**

**Bisection width: 2k**

**Edges per node: 4**

1. What is the diameter of a tree network with k levels and S children?

**Answer:** **2(k-1)**