

HPC Fall 2010 – Homework 1

Robert van Engelen

Due date: September 1, 2010

1. Consider a sequential algorithm A_s and parallel version A_P . The sequential time of A_s is $t_s = 18$ seconds and the total parallel execution time t_P of A_P for $P = 1, \dots, 6$ was measured in seconds as follows:

$P =$	1	2	3	4	5	6
$t_P =$	24.00	13.50	10.00	8.25	7.20	6.50
$S_P =$						
$E_P =$						

Determine the speedup S_P and efficiency E_P of the parallel algorithm compared to the sequential algorithm.

2. Consider a parallel algorithm A_P . The parallel execution time of the algorithm was measured in seconds for $P = 1, \dots, 6$ as follows:

$P =$	1	2	3	4	5	6
$t_P =$	20.0	11.0	8.0	6.5	5.6	5.0
$S_P^1 =$						
$\alpha_P =$	n/a					

For this example we determine the relative speedup S_P^1 . With the relative speedup, determine α_P from Gustafson's law:

$$S_P^1 = P + (1 - P)\alpha$$

Recall that α_P ($\alpha_P + \beta_P = 1$) measures the sequential part, but here we ignored the effect of the data size on execution time.

3. Assuming that the parallel algorithm of question #2 perfectly follows Amdahl's law

$$t_P = f t_s + (1 - f)t_s/P$$

with fixed sequential part t_s , the Karp-Flatt metric determines f by rewriting Amdahl's law:

$$f = \frac{\frac{1}{S_P} - \frac{1}{P}}{1 - \frac{1}{P}}$$

Compute f and t_s for the algorithm of question #2.

4. The idealized formula for energy consumption by a processor core is

$$E = ct f V^2$$

where c is a CPU-dependent constant c , t is total execution time, f is the processor's clock frequency, and V is the supply voltage. The frequency and voltage are correlated

$$f = \alpha V$$

with $\alpha = 0.2 \cdot 10^9 \text{HzV}^{-1}$. Suppose our algorithm must complete in $t = 10$ seconds and needs a total of 10^{10} clock cycles to execute. What is the energy saving ratio when we run this perfectly parallelizable algorithm on two CPU cores in parallel instead of on one core, assuming both take t seconds to complete?

5. Consider the following loop

```
k = 0;
for (i = 0; i < n; i++)
{ k = k + 8;
  a[k] = 0;
}
```

- (a) Does this loop exhibit any cross-iteration dependence (flow or anti dependence)? Explain and indicate any dependences between statements that you can find.
- (b) Assume we want to eliminate variable k all together from this loop. Rewrite the loop by removing k and changing the code so that the code performs the same operations on $a[]$ (sets certain $a[]$ values to zero).