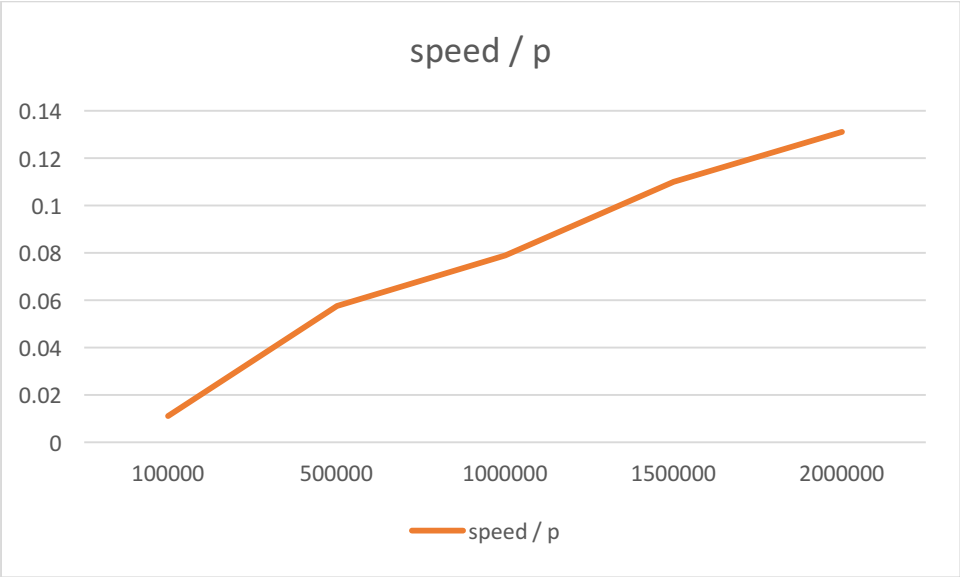
Comp 428 Ass 2

Maximilien Malderle

26562906

Q1:




As we can see the speed up vs processors increases as the input size increases. The ordinary quick sort numbers are always smaller than the parallel version, the reason being that the communication overhead eliminates all the parallel benefit.

Q2:

Quick Sort: 0( n log n)

Hyper Quick:

Here we have Reorganization:

O( log p)

Array Reorganization:

O( n/p )

Pivot Broad Cast:

O( n/p)

Local Quick Sort:

n/p log n/p

Speed up:
Precise:

$$\frac{n\log(n)}{\frac{n}{p}\log\left(\frac{n}{p}\right)+\frac{n}{p}+\log(p)}$$

equals:

$$\frac{n\log(n)}{\frac{n}{p}+\frac{n\log\left(\frac{n}{p}\right)}{p}+\log(p)}$$

test with n = 4 and p = 2

Speed up ~ 1.36

Which is greater than one.

Regular Big O:

(n log n) / (n/p)

equals:

p log n

Efficiency:
Precise:

$$\frac{\frac{n\log(n)}{\frac{n}{p}\log\left(\frac{n}{p}\right)+\frac{n}{p}+\log(p)}}{p}$$

Equals:

$$\frac{n\log(n)}{p\left(\frac{n}{p}+\frac{n\log\left(\frac{n}{p}\right)}{p}+\log(p)\right)}$$

test with n = 4 and p = 2

0.68

Which is sub optimal.

Actual:

(p log n) / p

equals:

log n

Q 3:

In terms of load balancing the Nodes in a Bitonic search are idle less frequently. They are constantly being used for compare and switch, whereas in Quicksort the Nodes separate big from small at every dimension, meaning that sometimes nodes may have more or less than others.

In terms of efficiency Bitonic needs a bitonic sequence, whereas Quick sort requires a pivot. A good pivot allows for a nice even tree that does not run too deeply in one direction. With limited input data the speed of quicksort can be O (n). Whereas no matter the input type Bitonic search yields O(log^2 n) time.

Q4:

| | | | | | | |
|---|---|---|---|---|---|---|
| 2,13 | "0000" | 2,5 | "000" | 2,3 | "000" | 1,2 |
| 1,11 | "0001" | 1,6 | "001" | 1,4 | "001" | 3,4 |
| 3,9 | "0010" | 3,8 | "010" | 5,8 | "010" | 5,6 |
| 4,10 | "0011" | 4,7 | "011" | 6,7 | "011" | 7,8 |
| 5,12 | "0100" | 12,13 | "100" | 9,12 | "100" | 9,10 |
| 6,15 | "0101" | 11,15 | "101" | 10,11 | "101" | 11,12 |
| 8,16 | "0110" | 9,16 | "110" | 13,16 | "110" | 13,14 |
| 7,14 | "0111" | 10,14 | "111" | 14,15 | "111" | 15,16 |