

TP Quicksort Hypercube

Analyse de performance

Axel Fahy & Jessy Marin & Rudolf Höhn

November 16, 2015

1 Complexité

Pour trouver la complexité totale, on doit séparer le problème et calculer la complexité pour chaque fonction du code. Dans les formules, $\beta = T_{communication}$.

- **Sort** $O(n * \log(n))$
- **Partition** $O(n)$
- **Exchange** $O(2 * \beta)$
- **Diffusion** $O(n * \beta)$
- **Reunion** $O(d * (n + \text{sort})) = O(d * (n + n * \log(n)))$
- **QuickSort** $O(\text{sort} + d * (\text{Diffusion} + \text{Partition} + \text{Exchange} + \text{Reunion})) = O(n * \log(n) + d * (n * \beta + n + 2 * \beta + d * (n + n * \log(n))))$

La complexité totale est donc :

$$C = O(n * \log(n) + d * (n + (n + 2) * \beta + d * (n + n * \log(n))))$$

2 Speedup

Le speedup théorique est : $S = \frac{T_{seq}}{T_{par}}$. Dans notre cas, le speedup est :

$$S = \frac{n * \log(n)}{d * (n + (n + 2) * \beta + d * (n + n * \log(n)))}$$

3 Efficacité

L'efficacité théorique est : $E(p, n) = \frac{T_{seq}(n)}{p * T_{par}(p, n)}$. Dans notre cas, l'efficacité est :

$$E = \frac{n * \log(n)}{p * (d * (n + (n + 2) * \beta + d * (n + n * \log(n))))}$$

4 Isoefficacité

La fonction d'isoeffacité est $n \sim p * \log_2(p)$.