ECE 669

Parallel Computer Architecture

Lecture 12

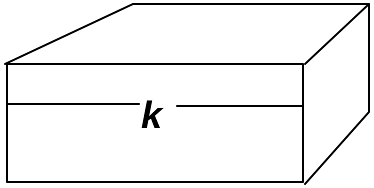
Interconnection Network Performance

Performance Analysis of Interconnection Networks

- Bandwidth
- Latency
  - Proportional to diameter
- Latency with contention
- Processor utilization
- + Physical constraints

Bandwidth & Latency

Direct



Latency

- **Average latency k_d:**

nk – worst case

$n\frac{(k-1)}{2}$ - *torus* - 1 dir channels

$\sim n\frac{k}{4}$ – *torus* – 2 dir channels

$\sim n\frac{k}{3}$ – no end conns – 2 dir channels

- **Bandwidth per node - more complex**
  - $\frac{1}{B}$ **if all near neighbor messages**
  - **If average travel dist then?**
  - **Let**

$$\text{Avg DIST} = \frac{nk}{3}$$

$$\text{Msg size} = B$$

Analogy

- – If each student takes 8 years to graduate
- – And if a Prof. can support 10 students max at any time

How many new students can Prof. take on in a year? $8x = 10$

Each year take on $\frac{10}{8}$

° Similarly:

- Network has Nn channels
- # of flits it can sustain = Nn
- # of msgs it can concurrently sustain= $\frac{Nn}{B}$
- Each msg flit uses $\frac{nk}{3}$ channels to dest
- So

$$N\ x\ \frac{nk}{3} = \frac{Nn}{B}$$

or

$$BW \text{ per node } = x = \frac{3}{kB}$$

Another way of getting *BW* is:

- Max # msgs in net at any time $= \dfrac{Nn}{B}$
- These take $= \dfrac{kn}{3}$ cycles to get delivered, during which time no new mgs can get in
- I.e. we can inject $\dfrac{Nn}{B}$ msgs every $\dfrac{kn}{3}$ cycles

  or # injected per node per cycle

$$= \frac{Nn}{B} \cdot \frac{3}{kn} \cdot \frac{1}{N}$$

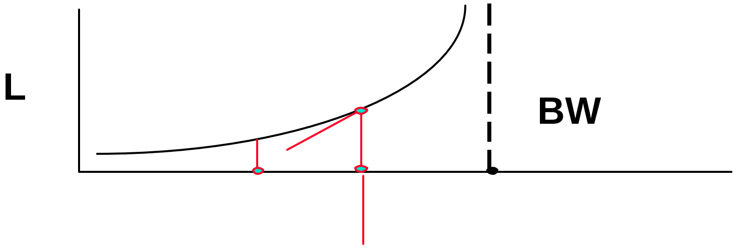
$$= \frac{3}{Bk}$$

- Note: We have not considered contention thus far.
- In practice, latency shoots up much before we achieve the theoretical due to contention.

What’s a good performance metric to analyze networks

- Possibilities:
  - Bandwidth
  - Latency
- Discuss
  - ***Bandwidth*** - good when latency not an issue. I.e., if we can overlap all communication with computation
    - Estimate of how much info we can transport
  - ***Latency*** - good when not much parallelism exists (critical sections, e.g.), or when communication cannot be overlapped with computation.

$$U = \frac{1}{1 + \text{req rate} * L\,(U)}$$



  - Effective processor utilization best metric - but always not easy to derive.

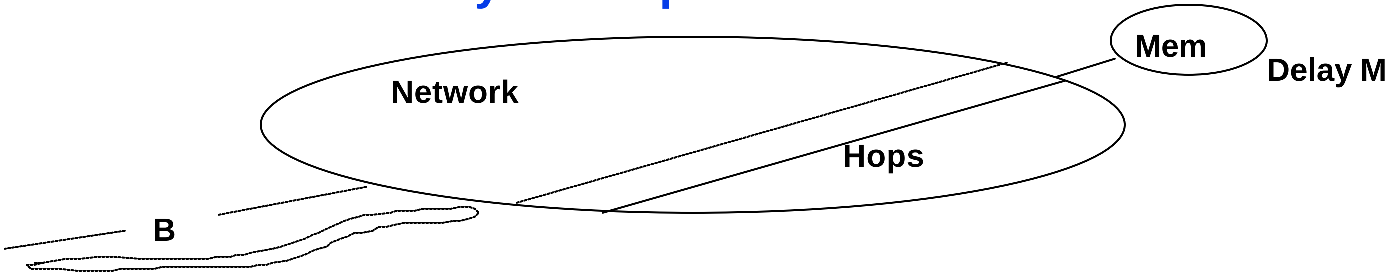
Performance Analysis

- First, analysis based on latency alone
  - Taking contention into account, but ignoring technological constraints

Such analysis is o.k., as long as we do not compare workloads with different traffic rates

With no contention,

$$\text{Latency } T = \text{hops} + M + B$$



With contention,

$$\text{Latency } T = (1{+}c)\ \text{hops} + M + B$$

Average contention delay cycles at each switch

Direct k-ary n-cube (Agarwal paper)

$$c = \frac{\rho B}{(1-\rho)} \frac{(k_d - 1)}{k_d^2} \left(1 + \frac{1}{n}\right)$$

$$\text{hops} = n k_d$$

ρ = Fraction of bandwidth consumed,

$= mBk_d$

Distance traveled in a dim. in a unidirectional torus

$$k_d = \frac{k-1}{2}$$

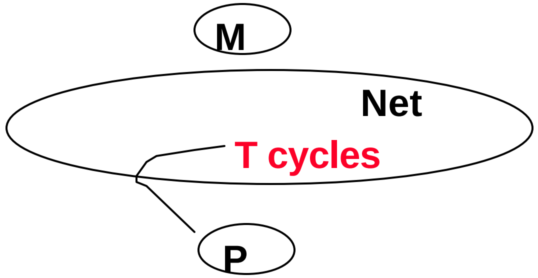
m = traffic rate, [probability of request]

k-ary, n-cube latency

$$\therefore T = \left[1 + \frac{\rho B}{(1-\rho)} \frac{(k_d - 1)}{k_d^2} \left(1 + \frac{1}{n}\right)\right] n k_d + M + B$$

A better metric

- Processor utilization *U*
- Or, how network delay impacts overall performance
- *U* = Fraction of time spent doing useful work



- Ideally,
  - Each instruction takes 1 cycle
  - If probability of message (remote mem req) on each useful cycle is m and corresponding latency is T

Each instruction takes 1+mT cycles

or

$$U = \frac{1}{1 + mT}$$

Deriving *U* is not easy

- **Notice**

  m = **Probability of a message on a** useful **processor cycle**

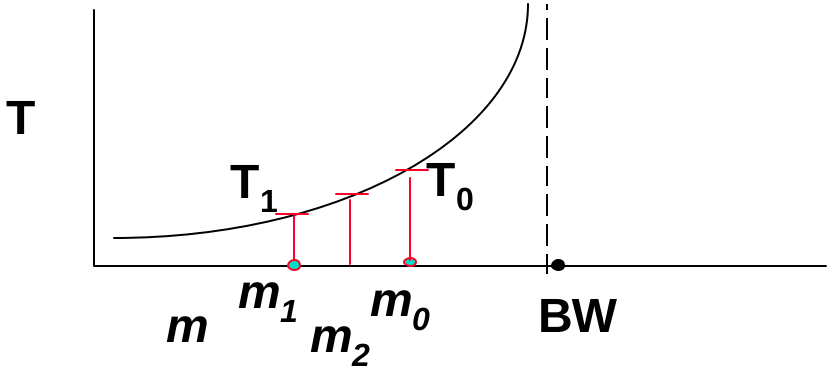
  $\mathbf{m_{eff}} = \boldsymbol{m} \bullet \boldsymbol{U}$ **= probability of msg on any cycle**

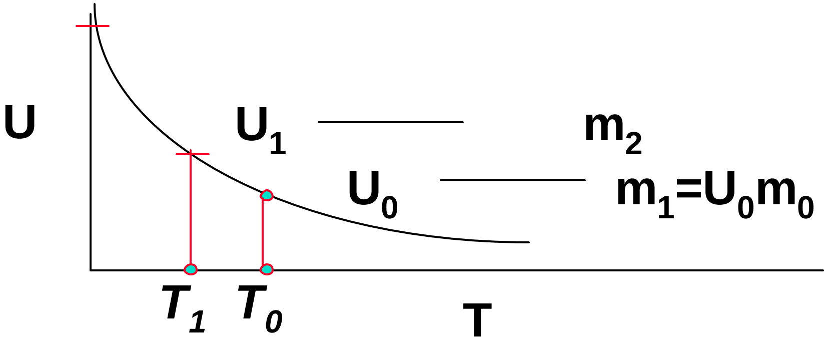
  $\mathbf{T} = \boldsymbol{f}\,(\boldsymbol{m_{eff}})$ **= network delay as a function of** *m*

$$U = \frac{1}{1 + mT}$$

  or # of useful processor cycles depends on ***T*** **and** $\mathrm{m_{eff}}$

- **Cyclic dependence!**





Processor utilization

- **k-ary n-cube**

  m **= Probability of msg on useful cycle**

- **Channel utilization** $\rho = mBk_dU$
- **Latency** $T = \left[1 + \frac{\rho B}{(1-\rho)} \frac{(k_d - 1)}{k_d^2} \left(1 + \frac{1}{n}\right)\right] nk_d + M + B$ **[1]**
- **Processor utilization** $U = \frac{1}{1 + mT}$ **[2]**
- **Two equations [1] and [2]**
- **Two unknowns: *T, U***
- **Solving,**

$$T = \frac{T_0}{2} + \frac{Bk_d}{4} - \frac{1}{2m} + \frac{1}{2}\sqrt{\left(T_0 - \frac{Bk_d}{2} + \frac{1}{m}\right)^2 + 2B^2(k_d - 1)(n + 1)}$$

- **Where** $T_0 = nk_d + M + B$

  $=$ unloaded network latency

- **and** $U = \frac{1}{1 + mT}$

Technology Constraints

How do technological constraints impact network design, performance?

- Constraints
    - Wire lengths limit maximum speed of clock
    - # pins limit size of nodes
    - Bisection width limits # of wires per channel
- Software
    - Can we compile to the architecture?
    - Can users specify programs?
    - Is it scalable?

The performance equation becomes

- **Latency**
- **Recall,** T = [(1 + c) hops + B]
- **Now,**

$$T = \overbrace{[T_{switch} + T_{wire}(n)]}^{\text{cycle time}} \left[(1 + c)\ hops + \frac{B}{W(n)}\right]$$

switch delay

f(n)

wire delay

$$E.g \ \mu \ k^{\frac{n}{2}-1}$$

$$\mu \ N^{\frac{1}{2}-\frac{1}{n}}$$

msg length

channel width: function of bisection width

$$E.g \ \mu \ k$$

$$\mu \ N^{\frac{1}{n}}$$

- **See paper for details.**
- **Summary: Constraints favor small n.**