#DOC429 - Study Notes for Parallel Algorithms

Paul Gribelyuk (pg1312, a5)

May 14, 2013

1 Metrics (64 pages)

1.1 Architectures

- *Message passing* used in the case of many machines having only local memory.
- *Shared address space* used when multiple processors in same computer access same memory; *UMA* has equal access times for all, otherwise *NUMA*, so address space is distributed among processors
- *Interconnect Network (IN)* provides hardware to pass messages; topology defines performance: ring, mesh, hypercube

PRAM is an idealization of shared memory MIMD with UMA. Different access modes:

- EREW - exclusive read exclusive write; minimizes concurrency
- CREW - concurrent read exclusive write
- CRCW - concurrent read, concurrent write; to write concurrently, semantics can be *common*, *arbitrary*, *priority*, *reduce* (sum or max or some other reduce operation).
- ERCW - dumb

1.2 Embedding

Binary grey codes used to convert ring network to hypercube:

$$G(0,1) = 0 \quad (1)$$

$$G(1,1) = 1 \quad (2)$$

$$G(i,n+1) = \begin{cases} G(i,n) & i < 2^n \\ 2^n + G(2^{n+1} - 1 - i, n) & i \geq 2^n \end{cases} \quad (3)$$

Mesh to hypercube can be done by concatenating RGC of each dimension. For node i in $2^{r_1} \times \ldots \times 2^{r_m}$ mesh, mapping is $G(i_1, r_1)G(i_2, r_2)\ldots G(i_m, r_m)$. Can also map a tree to a hypercube. At each level k of the tree, assign the k-th bit either 0 or 1.

1.3 Communication patterns

- Simple message
- One-to-All broadcast; dual is single node accumulation
- All-to-All broadcast; dual is multi-node accumulation
- One-to-One personalized; dual is single node gather
- All-to-All personalized scatter, dual is multi-node gather
- Other (?)

1.4 Performance

- Run Time: T_p
- Speedup: $S_p = \frac{\text{best serial } T}{T_p}$
- Efficiency: $E_p = \frac{S_p}{p}$ is speedup per processor, usually less than 1
- Cost: $C_p = p \cdot T_p$, total amount of computation done on p processors. Cost optimal if $E_p = \Theta(1)$
-

Cost optimality means costs are quivalent to best serial runtime!
Example: to add n numbers on hypercube with $p = 2^d$ nodes each nodes doing $k = n/p$ serial steps, then partial sums reduced via single node accumulation:

$$\begin{aligned} \text{best serial } T &= n & (4)\\ T_p &= \frac{n}{p} + \log p & (5)\\ S_p &= \frac{p}{\frac{n}{p} + \log(p)} & (6)\\ E_p &= \Theta(1/(n/p + \log(p))) & (7)\\ C_p &= n + p\log(p) & (8) \end{aligned}$$

So cost-optimal if $n = \Theta(p\log(p))$.
To understand how algorithm scales, we look at *isoefficiency*. $O_p = C_p - Work$ is a measure of communication latency.

1.5 Communication Costs

- *startup time* t_s incurred once per message
- *per-hop time* t_h
- *transfer time* t_w

Store and forward messages:

$$t_{comm} = t_s + (mt_w + t_h)l$$

Cut-through routing:

$$t_{comm} = t_s + mt_w + lt_h$$

- *Diameter* is maximum length between any two nodes
- *Arc connectivity* is how many links must be broken to fragment network
- *Bisection width* is how many links must be broken to split network into 2 equal halves
- *cost* is usually the number of links in a network

topology	diameter	bisection	arc con	cost
completely connected	1	$p^2/4$	p-1	$p(p-1)/2$
star	2	1	1	p-1
binary tree	$2\log((p+1)/2$	1	1	p-1
linear array	$p-1$	1	1	$p-1$
2-D mesh w/o wrap	$2(\sqrt{(p)}-1)$	$\sqrt{(p)}$	2	$2p\sqrt{(p-1)}$
2-D wraparound	$2\lfloor\sqrt{(p)}/2\rfloor$	$2\sqrt{p}$	4	$2p$
hypercube	$logp$	$p/2$	$logp$	$p\log p/2$
wrap k-ary d cube	$d\lfloor k/2\rfloor$	$2k^{d-1}$	2d	dp

Different costs associated with these topologies:

operation	hypercube	mesh	ring
one-to-all bcast	$\min\{(t_s + t_w m)\log(p), 2(t_s\log(p) + t_w m)\}$	$2(t_s + t_w m\log(p)$	$t_s + t_w m\log(p)$
all-to-all bcast	$t_s\log(p) + t_w m(p-1)$	$(t_s + t_w m\sqrt{p})(\sqrt{p}-1)$	$(t_s + t_w m)(p-1)$
all-reduce	$\min\{(t_s + t_w m)\log(p), 2(t_s\log(p) + t_w m)\}$		
scatter, gather	$t_s logp + t_w(p-1)$		
ATA personalized	$(t_s + t_w m)(p-1)$		
circular shift	$t_s + t_w m$		

Isoefficiency is how E scales with amount of work W and with p. Goal is to find a way to scale W as a function of p.

$$T_p = \frac{W + O(W,p)}{p} \quad \Longrightarrow S_p = \frac{W}{T_p} = \frac{W\cdot p}{W + O(W,p)} \quad \Longrightarrow E = \frac{S}{p} = \frac{1}{1 + O(W,p)/W}$$

Another way to formulate cost-optimality:

$$pT_p = \Theta(W) \quad \Longrightarrow W + O(W,p) = \Theta(W) \quad \Longrightarrow W = \Omega(O(W,p))$$