Big Data Mining 52002 - Home Exam
Section 2: Parallel Computing (30 points)

Name : Netanel Azran , ID: 314992595

1 Question 1: Amdahl's Law with Communication (15 points)

We analyze parallel execution with communication overhead in a ring topology.

Given:

- Single processor time: $g(n)$
- Parallel fraction: p
- Processors: m in ring
- Communication cost: cmn

1.1 Part (a): Formulas

Total Time:

Three components contribute to runtime:

- Parallel portion: $pg(n)/m$
- Sequential portion: $(1-p)g(n)$
- Communication: cmn

$$T(m,n) = \frac{pg(n)}{m} + (1-p)g(n) + cmn$$

Speedup:

$$S = \frac{g(n)}{T(m,n)} = \frac{1}{\frac{p}{m} + (1-p) + \frac{cmn}{g(n)}}$$

Note: In the classical formulation (without communication), speedup is independent of n. Here the $cmn/g(n)$ term introduces an explicit dependence on the problem size.

1.2 Part (b): Optimal Processor Count

Minimize $T(m,n)$ by setting $\frac{dT}{dm} = 0$:

$$\frac{dT}{dm} = -\frac{pg(n)}{m^2} + cn = 0$$

Solving for m:

$$m^2 = \frac{pg(n)}{cn}$$

$$m^* = \sqrt{\frac{pg(n)}{cn}}$$

Second derivative $\frac{d^2T}{dm^2} = \frac{2pg(n)}{m^3} > 0$ confirms minimum.

1.3 Part (c): Numerical Results

Parameters: $n = 10^{12}$, $c = 1$, $p = 0.25$

Case 1: $g(n) = n^2$

$$m^* = \sqrt{\frac{0.25 \cdot 10^{24}}{10^{12}}} = \sqrt{0.25 \cdot 10^{12}} = 5 \times 10^5$$

Answer: **500,000 processors**

Case 2: $g(n) = n \ln n$

With $\ln(10^{12}) \approx 27.6$:

$$g(n) \approx 2.76 \times 10^{13}$$

$$m^* = \sqrt{\frac{0.25 \cdot 2.76 \times 10^{13}}{10^{12}}} \approx \sqrt{6.9} \approx 2.6$$

Answer: **Since m* must be an integer, we take :[2.6] = 3 processors**

Discussion: With $g(n) = n^2$ the computation-to-communication ratio $g(n)/n = n$ is enormous, so many processors pay off. With $g(n) = n \ln n$ the ratio is only $\ln n \approx 28$, so the ring overhead quickly exceeds any speedup from extra processors.

2 Question 2: MapReduce Matrix Multiplication (15 points)

Goal: Compute $C = AB$ where $C_{ik} = \sum_j A_{ij} B_{jk}$

2.1 Part (a): Algorithm Design

Round 1 - Mapping:

For A: $(i, j, A_{ij}) \to (j, (A, i, A_{ij}))$

For B: $(j, k, B_{jk}) \to (j, (B, k, B_{jk}))$

Round 1 - Reducing (key j):

Collect entries from both matrices with same j. For each combination of (i, A_{ij}) from A and (k, B_{jk}) from B:

Output: $((i, k), A_{ij} \cdot B_{jk})$

Round 2 - Mapping:

Identity mapper — pass each partial product through unchanged:

$((i, k),\ v) \ \to \ ((i, k),\ v)$

Round 2 - Reducing (key (i, k)):

Sum all partial products received for the same output cell:

$$C_{ik} = \sum_j A_{ij} B_{jk}$$

Output: $((i, k), C_{ik})$

2.2 Part (b): Sparse A, Dense B

Given: A has s nonzeros, B is dense (n^2 entries)

Round 1 Map phase: Emits s pairs from A and n^2 pairs from B: cost $\Theta(s + n^2)$.

Round 1 Reduce phase: For each j, column j of A has a_j entries and row j of B has n entries (dense), producing $a_j \cdot n$ partial products.

Total intermediate products emitted: $\sum_j a_j \cdot n = n \cdot s$

Round 2 Reduce phase: Sums the sn partial products to produce the output entries of C: cost $\Theta(sn)$.

Total complexity: $\Theta(n^2 + sn)$

- $s \ll n$: the n^2 term from reading the dense matrix B dominates
- $s = \Theta(n^2)$: the sn term gives $\Theta(n^3)$, recovering naive matrix multiply

2.3 Part (c): Both Sparse, Random Positions

Given: Both matrices have s nonzeros, uniformly random positions

Let X_j = nonzeros in column j of A, Y_j = nonzeros in row j of B

Distribution:

$$X_j,\ Y_j \sim \text{Binomial}\left(s, \frac{1}{n}\right)$$

(Expectation note): The Binomial assumption is used only for convenience; under the exact model (sampling without replacement), the expectation is unchanged:

Expected values:

$$\mathbb{E}[X_j] = \mathbb{E}[Y_j] = \frac{s}{n}.$$

Products for index j: $X_j \cdot Y_j$

Total products: $P = \sum_{j=1}^{n} X_j Y_j$

Expected value (using independence of A and B):

$$E[P] = \sum_{j=1}^{n} E[X_j]E[Y_j] = n \cdot \frac{s}{n} \cdot \frac{s}{n} = \frac{s^2}{n}$$

Total complexity:

$$\Theta\left(s + \frac{s^2}{n}\right)$$

Regime analysis:

- $s \ll \sqrt{n}$: $\Theta(s)$ - input reading dominates
- $s = \Theta(\sqrt{n})$: $\Theta(\sqrt{n})$
- $s = \Theta(n^2)$: $\Theta(n^3)$ - standard multiplication

Intuition: Random placement means nonzeros rarely share the same index j, reducing expected work by factor n compared to adversarial alignment.