Parallel Programming course. Parallelism theory

Obolenskiy Arseniy, Nesterov Alexander

Nizhny Novgorod State University

November 9, 2024

Contents

Parallelism efficiency metrics

Let us introduce a number of terms used in parallel programming theory:

- **Speedup (S)**: Ratio of the execution time of the best sequential algorithm to the execution time of the parallel algorithm.

$$S = \frac{T_1}{T_p}$$

- **Efficiency (E)**: Measure of how effectively the computational resources are being utilized.

$$E = \frac{S}{p} = \frac{T_1}{p \times T_p}$$

- **Scalability**: Ability of a system to maintain performance when the number of processors increases.

Amdahl's Law

Amdahl's Law addresses the maximum improvement to an overall system when only part of the system is improved.

- Formula:

$$S_{\text{max}} = \frac{1}{(1-P) + \frac{P}{N}}$$

- Where:
  - P is the proportion of the program that can be made parallel.
  - N is the number of processors.
- Implications:
  - Diminishing returns as N increases.
  - Emphasizes optimizing the sequential portion.

Amdahl's Law example

- Assuming 90% of a task can be parallelized (P = 0.9) and we have 4 processors (N = 4):
- Formula:

$$S_{\max} = \frac{1}{(1 - P) + \frac{P}{N}}$$

- Calculating for this particular example:

$$S = \frac{1}{(1 - 0.9) + \frac{0.9}{4}} = \frac{1}{0.1 + 0.225} = \frac{1}{0.325} \approx 3.08$$

- So, even though we have 4 processors, the speedup is only about 3.08 times faster than a single processor due to the non-parallelizable portion.

Amdahl's Law assumptions and limitations

Note that:

- Amdahl's Law assumes that the overhead of splitting and managing parallel tasks is (infinitely) small, which may not always be true.
- It doesn't account for other practical factors like memory access contention, communication delays between processors, or the complexity of load balancing.

Gustafson's Law (Gustafson-Barsis's Law)

Gustafson's Law, also known as Gustafson-Barsis's Law is a principle in parallel computing that addresses the scalability of parallel systems.

- Key note: Gustafson's Law suggests that the overall speedup in a parallel system is determined not only by the fraction of the task that can be parallelized but also by the size of the problem being solved. As the problem size increases, the potential speedup from parallelism also increases.
- Formula:

$$S(p) = p - \alpha(p - 1)$$

- Where:
  - $S(p)$ is the speedup with p processors.
  - α is the fraction of the workload that must be executed serially (i.e., non-parallelizable).
  - p is the number of processors.

Gustafson's Law notes

Note that:

- Unlike Amdahl's Law, Gustafson's Law argues that by increasing the size of the problem, the parallel portion can dominate, allowing for more effective use of additional processors.
- Gustafson's Law is more realistic in situations where the problem size increases with the number of processors.
- As the problem size grows, the portion that can be parallelized becomes larger, thus maximizing the benefit of adding more processors.

Parallelism overhead

Basically, parallelism overhead is the extra time that is required to manage parallel tasks.
Sources of overhead:

- Communication between processors.
- Synchronization delays.
- Data sharing and contention.

Best Practices for Efficient Parallelism

- Minimize synchronization and communication.
- Balance load among processors.
- Optimize data locality.
- Use appropriate parallel programming constructs.

Flynn's Classification

Categorizes computer architectures based on instruction and data streams.

	Single Instruction (SI)	**Multiple Instructions (MI)**
Single Data (SD)	Traditional single-core processors	Fault-tolerant systems (e.g. space shuttle control systems)
Multiple Data (MD)	Vector instruction in CPU, Graphics Processing Units (GPUs)	Multi-core processors, Distributed computing systems

Thank You!

References