CMSC416 Instructor Notes Spring 2024
Topics: Performance Analysis and Tools

1 Review of Key Terms

Strong Scaling: Measures how the runtime decreases as more processes are used to solve a fixed-size problem.

Weak Scaling: Measures how the runtime changes as the problem size increases proportionally with the number of processes.

Amdahl's Law: Highlights the impact of the sequential part of the program on the overall speedup achievable with parallelization.

2 Analytical Techniques and Time Complexity Analysis

The lecture began with a discussion on analytical techniques for basic time complexity analysis.

Parallel Prefix Sum was revisited as an example to calculate time complexities, including the time complexity for communication. The model used involved log(p) phases, with each phase entailing message sending between neighbors, approximating communication costs.

2.1 Models for Communication Time Complexity

Two models were introduced to add complexity to the initial analysis.

LogP model: A model considering the startup latency per message and the processor's busy time before it can proceed to the next operation.

The alpha plus n beta model: where alpha represents the startup latency, and beta represents the cost per unit of message size.

The complexity of communication modeling can increase when considering factors like startup overheads for each message, bandwidth costs per message, and network conditions.

3 Isoefficiency Analysis

A method to understand how the problem size needs to increase with the number of processes to maintain a certain efficiency level in parallel programs. It involves calculating the ratio of overhead (like communication time) to useful computation to determine scalability.

3.1 1D and 2D Game of Life/Stencil

The lecture applied these concepts to analyze the parallelization scalability of the Game of Life, comparing 1D and 2D data decompositions. For 1D decomposition, the amount of computation per process is proportional to the number of cells assigned to it (n/p), and the communication overhead is influenced by the boundary cells needing data exchange.

The lecture explored 1D and 2D decomposition strategies using the Game of Life as an example. The impact of these strategies on communication overhead and efficiency was analyzed, highlighting the quadratic vs. linear relationship between problem size growth and the number of processors.

4 Empirical Methods for Performance Analysis

Moving from analytical to empirical methods, the lecture outlined the use of performance tools for collecting data and analyzing program performance. This includes manual timers within the code and more sophisticated tools for tracing and profiling.

Tracing Tools record detailed execution events and timings, while profiling tools aggregate time spent in functions or code regions. The lecture highlighted various tools, like GPROF for profiling sequential programs and more advanced tools for parallel programs, such as TAU and HPCToolkit.