Let’s Get Real: Amdahl’s Law

Paul Lu
Associate Professor
Dept. of Computing Science
University of Alberta
Edmonton, Alberta, Canada
paullu@cs.ualberta.ca

This is NOT a research talk

Overview

1. Goals for Seminar Series
2. Speedup
3. Amdahl’s Law
4. …in Practice
5. The Good News
6. Concluding Remarks

Goals of Seminar Series

- Coverage of topics NOT usually included in workshops
  - Amdahl's Law
  - Granularity
  - Benchmarking basics
  - Parallel programming tips
  - Trends in parallel systems: multicore
  - Feedback, contributions welcome!

Parallelism is great

- There's often a "feeling" that parallelism will solve all of our computational problems
  - "If our server is not fast enough, let's get a cluster."
  - Many well-known Web sites are highly parallel: Google, Amazon, Olympics/IBM, banks
- But, parallel computers are no more Turing-complete than sequential computers
- There is still a lot of research to be done on algorithms and systems software

Speedup

- Let t_1 be the time to solve the problem sequentially (i.e., wall-clock time)
- Let t_p be the time to solve the problem with *p* processors (i.e., in parallel)
- Then, speedup *S(p)* for problem size *n* is:

$$S(p) = t_1 / t_p$$

  - $S(p) = p$ is (unit-)linear or perfect speedup. Rare.
  - $S(p) < p$ is sublinear speedup. Common.

Amdahl's Law (1)

- In 1967, Gene Amdahl pointed out that *the inherently sequential portions of a parallel program will ultimately determine the maximum speedup*.
- It is "obvious", but so what?
- Let *total* = *seq* + *para* = 1, where *seq* and *para* are the sequential and parallel portions

Amdahl's Law (2)

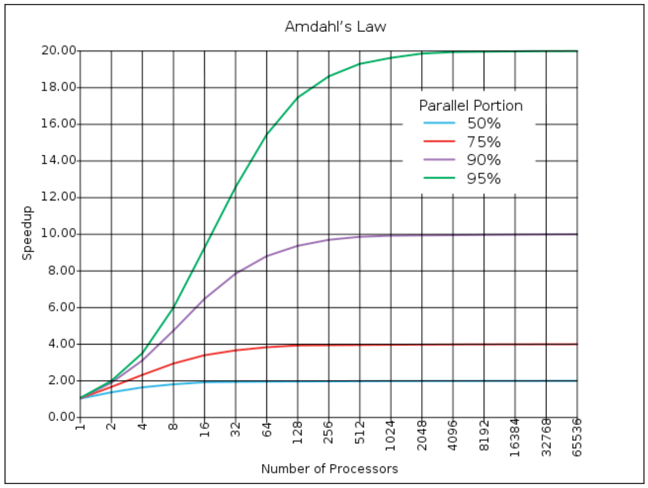
- Let *total* = *seq* + *para* = 1
- The parallel work will be divided among *p* processors as per (*para* / *p*)
- From the speedup equation:
  - *S(p)* = (*seq* + *para*) / (*seq* + *para* / *p*)
    = 1 / (*seq* + *para* / *p*)
- Then:
  - limit(*p*->infinity) *S(p)* = 1 / *seq*
- Yes, maximum speedup is 1 / *seq* !

Amdahl's Law (3)

For example:

Parallel Portion (*para*)	Sequential portion (*seq*)	Max Speedup by Amdahl's Law (1 / *seq*)
0.5	0.5	2
0.75	0.25	4
0.9	0.1	10
0.95	0.05	20

Worse, these are asymptotes



Wikipedia, http://en.wikipedia.org/wiki/File:AmdahlsLaw.svg

Amdahl's Law in Practice (1)

- Capacity (A) vs Capability (B) clusters
  - Cluster A: Many cores, Gigabit Ethernet
  - Cluster B: Many cores, Infiniband (20 Gbit/s)
- *Idealized* Bandwidth-bound Problems:

Compute Time	GigE Time	GigE Max SpeedUp	Compute Time	IB Time	IB Max SpeedUp
90%	10%	10	99.5%	0.5%	200
			98%	2%	50
			95%	5%	20

Amdahl's Law in Practice (2)

- Capacity (A) vs Capability (B) clusters
  - Cluster A: Many cores, Gigabit Ethernet
  - Cluster B: Many cores, Infiniband (20 Gbit/s)
- What if you have 20% faster cores, same GigE:

Compute Time	GigE Time	GigE Max SpeedUp
72%	28%	3.57

- That's a big drop in maximum possible speedup.

1/20/09 11

The Good News

- Often, you don't "need" parallelism
  - Naturally/embarassingly parallel problems
    - Parameter sweeps
- Sometimes, your problem size grows along with your machine size
  - Scale problem size with machine size, increasing the computational work
    - Sometimes referred to as "scale-up"
    - See Gustafson's Law
- Often, you don't have to do the coding
  - Just make sure you give the right resources to those who are doing that work.
  - In a future presentation, I will discuss "The Need for Diversity in HPC"

1/20/09 12

Concluding Remarks

- Amdahl's Law tells us that even if there is a 10% sequential bottleneck, then the maximum speedup is 10.
- Fortunately, your problem might be naturally parallel, or it can scale up.

