Lecture 2.3
Weak Scaling

EN 600.320/420

Instructor: Randal Burns

31 January 2018

Department of Computer Science, *Johns Hopkins University*

Gustafson's Law

Reformulate Amdahl's law for a fixed amount of time, rather than a fixed problem size

$$S_{\text{latency}}(s) = 1 - p + sp$$

- *p* is the optimized fraction
- *s* is the number of scaling resources (cores)
- $S_{latency}$ is the speedup realized (same as Amdahl's law)

Captures concept that as resources increase, we tend to solve bigger problems on more hardware.

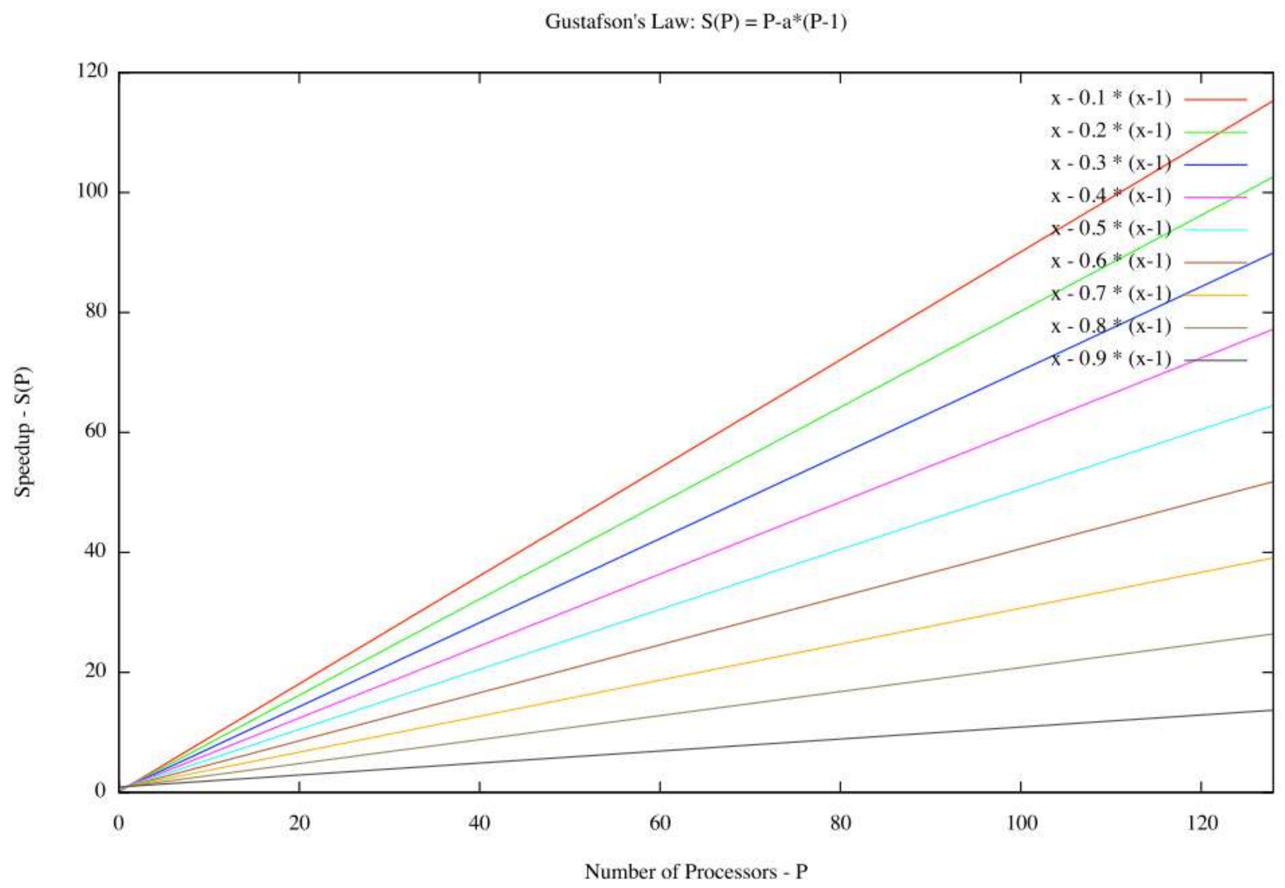
Weak versus Strong Scaling

- Strong scaling
  - how the solution time varies with the number of processors for a fixed *total* problem size
  - Amdahl's law
- Weak scaling
  - how the solution time varies with the number of processors for a fixed problem size *per processor*
  - *Gustavson's law*

- *Parallel efficiency applies to both concepts*

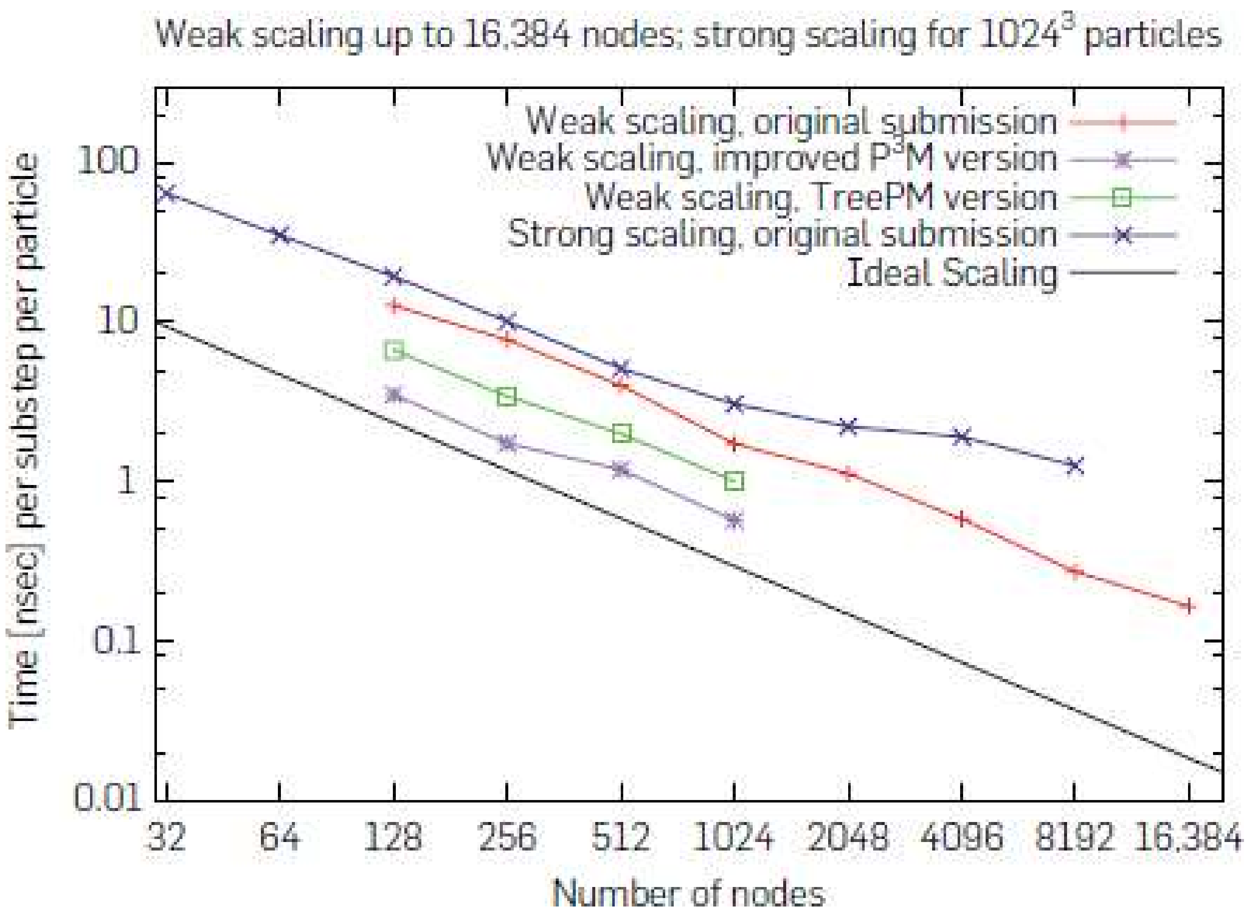
Visualizing Weak Scaling

- Using a scaleup chart
  - Speedup over serial for bigger problem sizes



Visualizing Weak Scaling

- As overhead – very confusing



(De)Merits of Weak Scaling

- Bigger problems = bigger science
  - Dominant trend in HPC over last 25 years
- Weak scaling coming to an end
  - Processor growth is far outstripping memory growth
  - Weak scaling can't continue
- Weak scaling more robust measurement (often)
  - Strong hits a wall as reduced total time magnifies fixed startup costs
- Weak scaling difficult to express for algorithms
  - Scaling happens in total work, e.g. O(n log n), not in the input size

Problem size != input size

Final Thoughts

- In this class
  - Efficiency = efficiency chart (relative to 1.0)
  - Speedup is a speedup chart (relative to linear speedup)
- Express your scaling model
  - Weak (variable problem size)
  - Strong (fixed problem size)
- Try not to mix weak and strong scaling
  - Confuses measures (more next week)