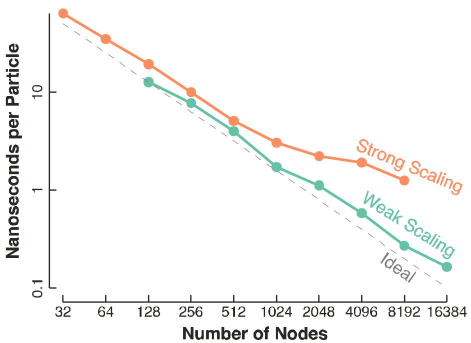
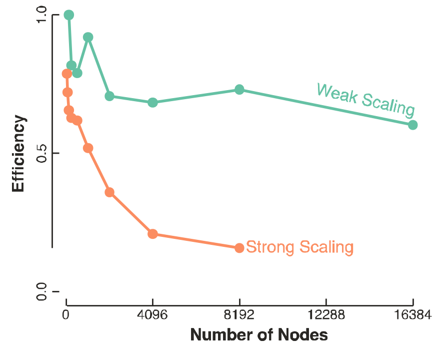
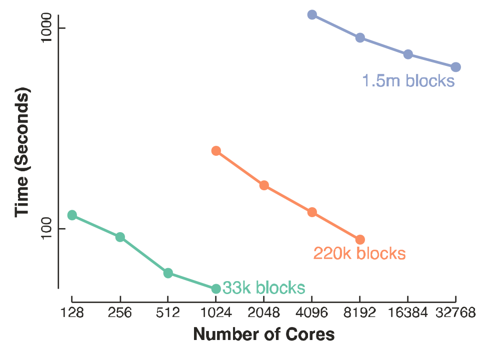
Exceptional service in the national interest

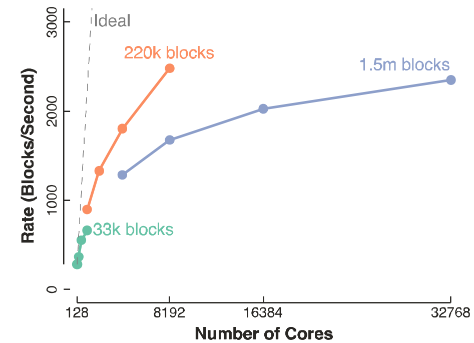
SAND2015-8683PE

Sandia National Laboratories









Measuring Parallel Software Scalability: You’re Doing it Wrong

SNL Computational Science Seminar Series

October 13, 2015

Kenneth Moreland

Sandia National Laboratories

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND 2015-4577 C

Parallel Algorithm Speedup

$$S(n,p) = \frac{T^*(n)}{T(n,p)}$$

Parallel Algorithm Speedup

Serial time for large problem sizes

Cannot be measured in practice

$$S(n,p) = \frac{T^*(n)}{T(n,p)}$$

Efficiency

$$E(n,p) = \frac{S(n,p)}{p}$$

Efficiency

$$E(n,p) = \frac{S(n,p)}{p} = \frac{T^*(n)}{p\,T(n,p)}$$

Karp-Flatt Metric

$$e(n,p) = \frac{\frac{1}{S(n,p)} - \frac{1}{p}}{1 - \frac{1}{p}}$$

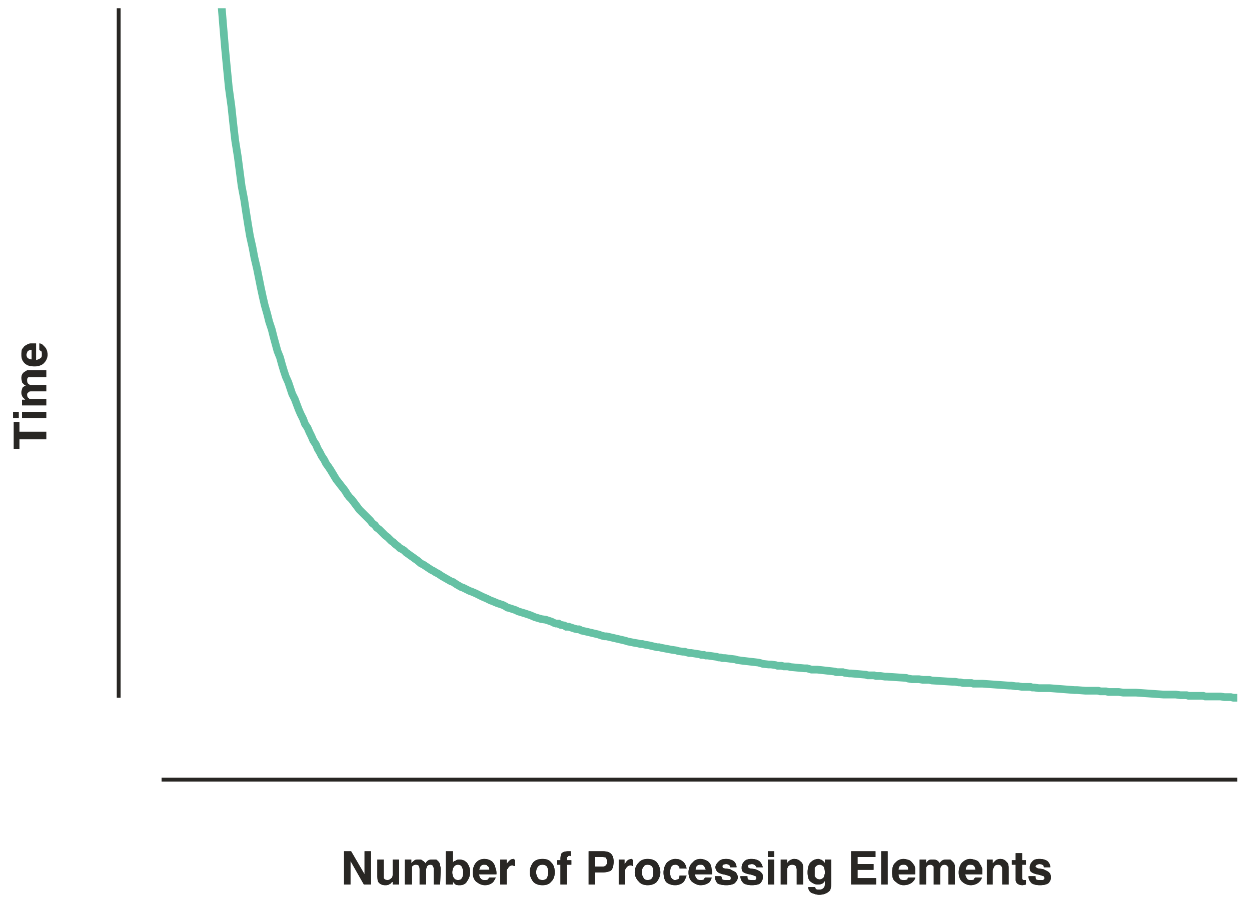
Isoefficiency Metric

$$T(n,1) \geq \frac{E_d}{1-E_d} T_o(n,p)$$

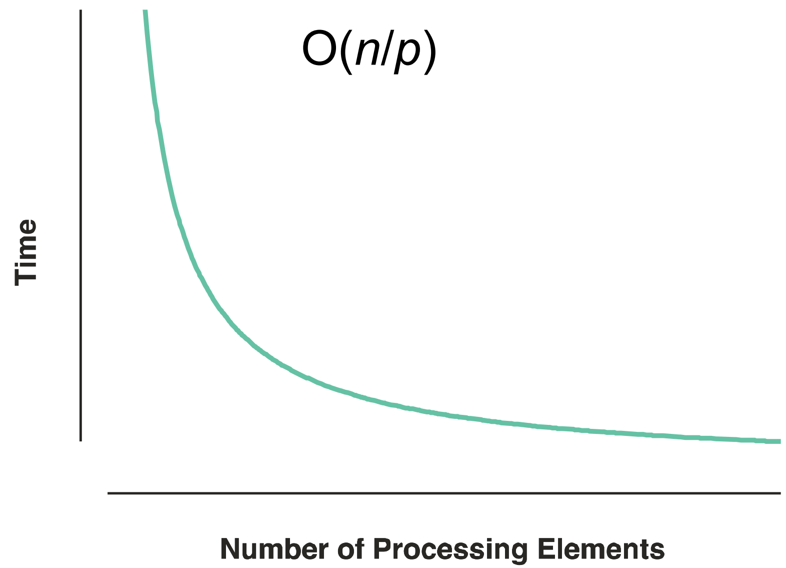
Measuring Scalability in Practice

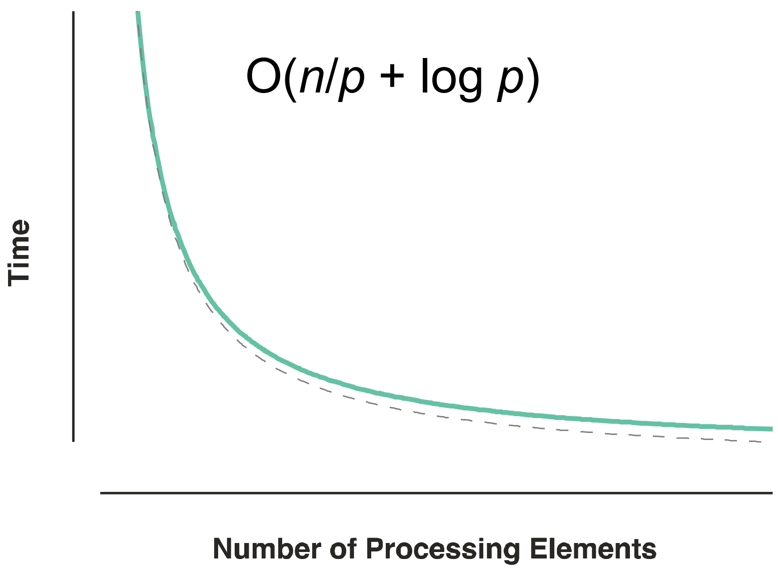
- Strong Scaling: Behavior as processing elements are increased and problem size held constant.
  - Per Amdahl’s Law, strong scaling always has its limits.
- Weak Scaling: Behavior as processing elements and job size are increased proportionally.
  - Per Gustafson-Barsis Law, weak scaling can possibly be increased indefinitely.
- Scaling is often demonstrated with absolute run time over different scales.

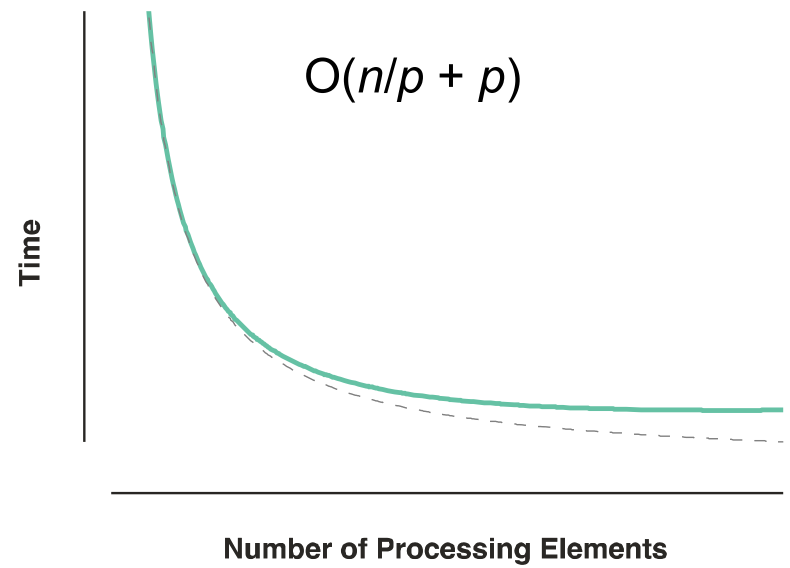
Demonstrating Strong Scaling



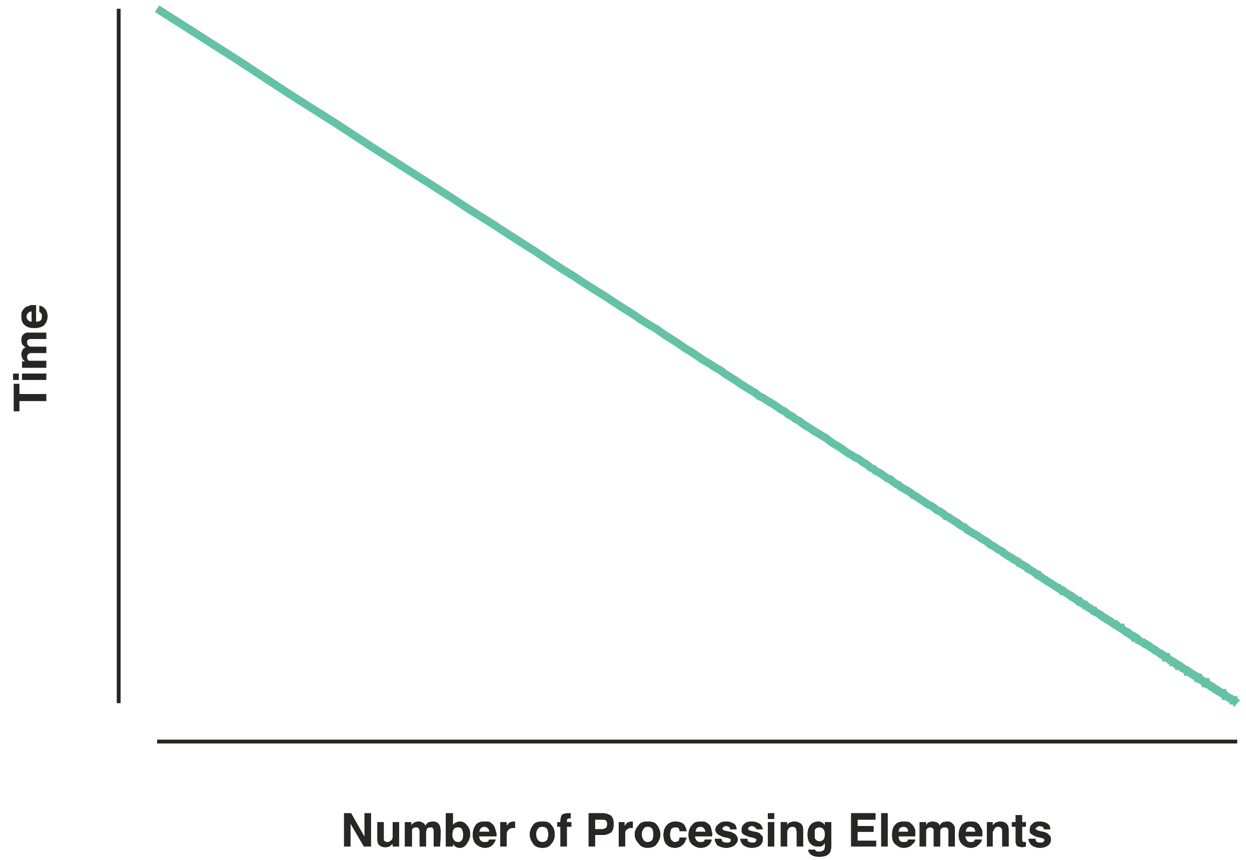
Measuring Strong Scaling



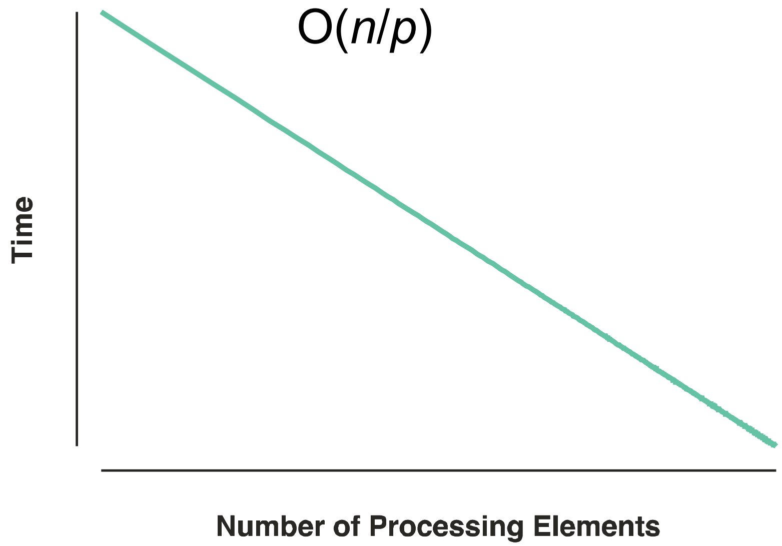


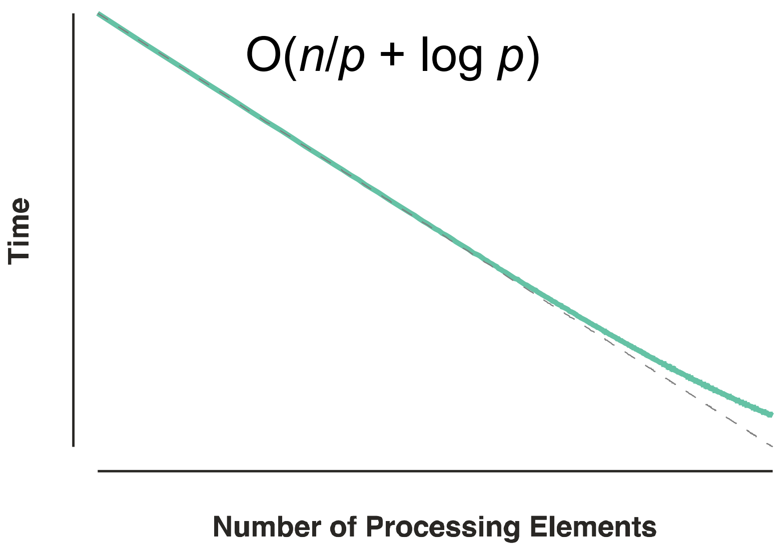


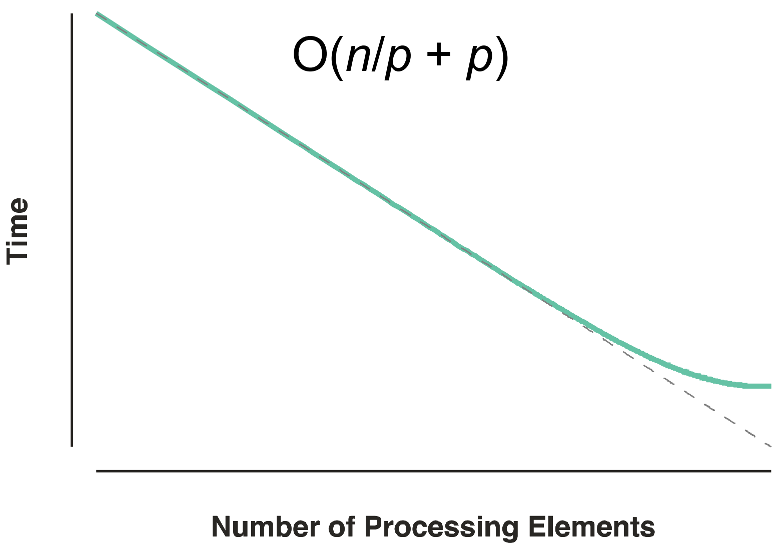
Strong Scaling with Log Axes



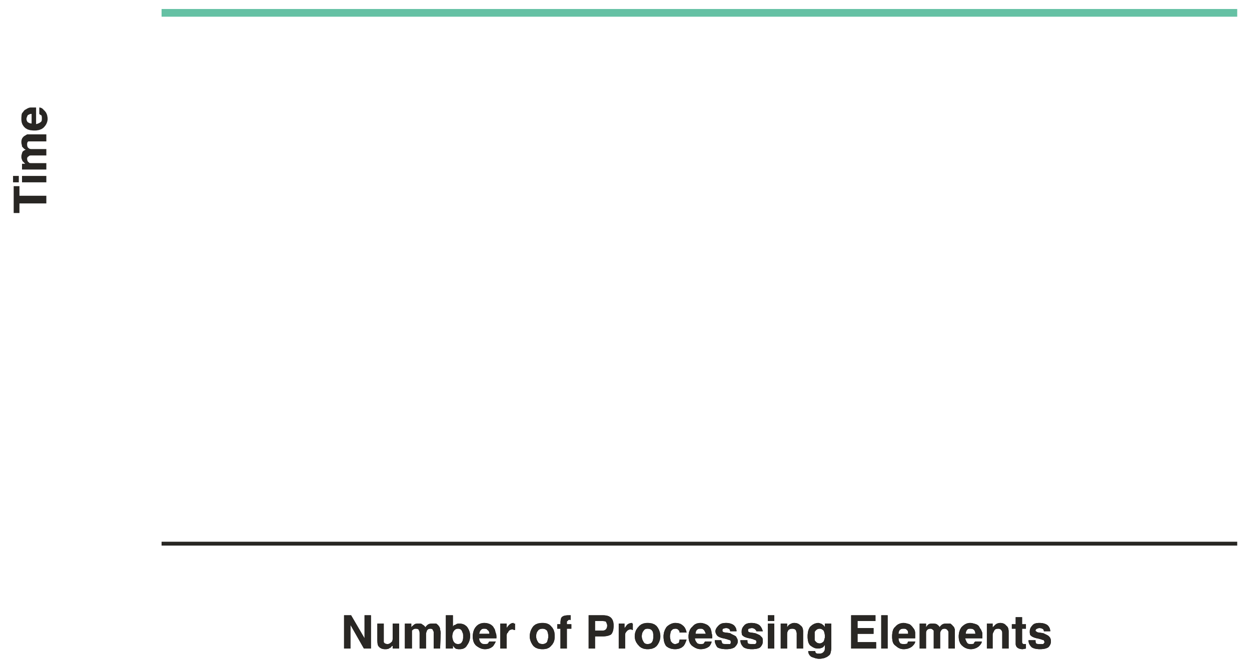
Measuring Strong Scaling with Log



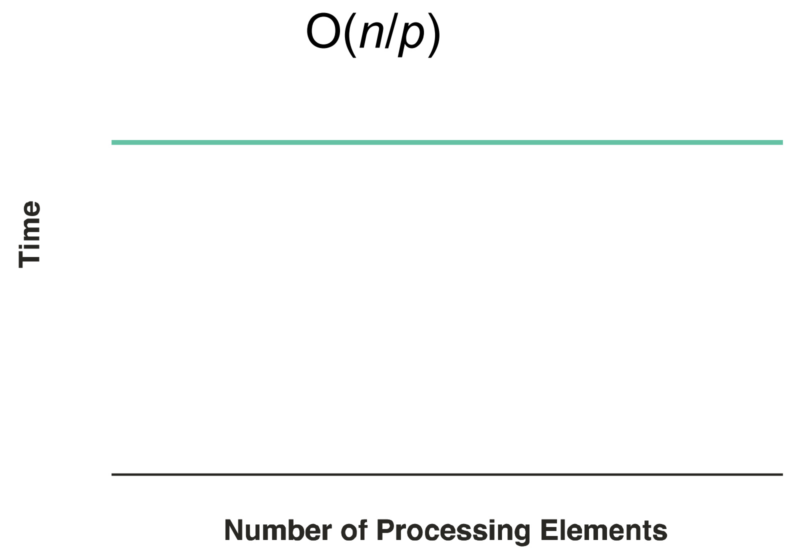


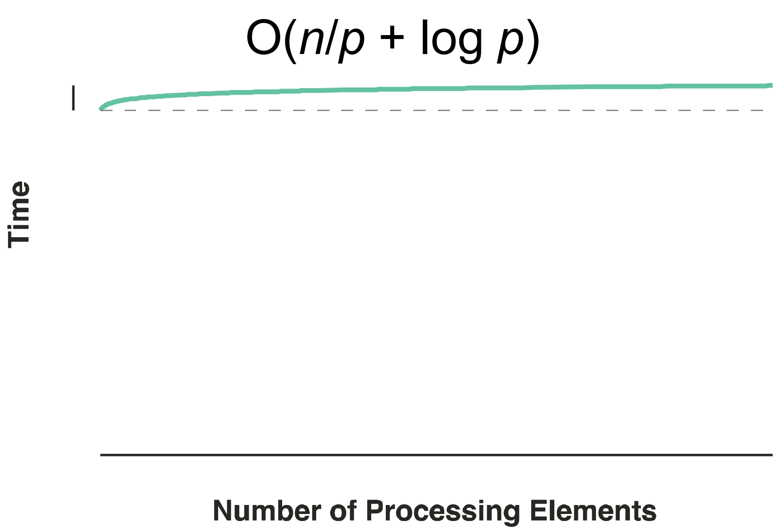


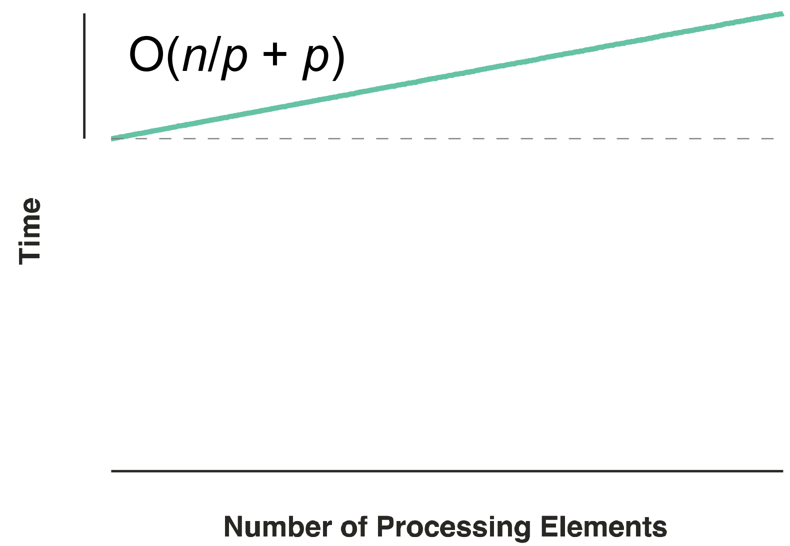
Demonstrating Weak Scaling



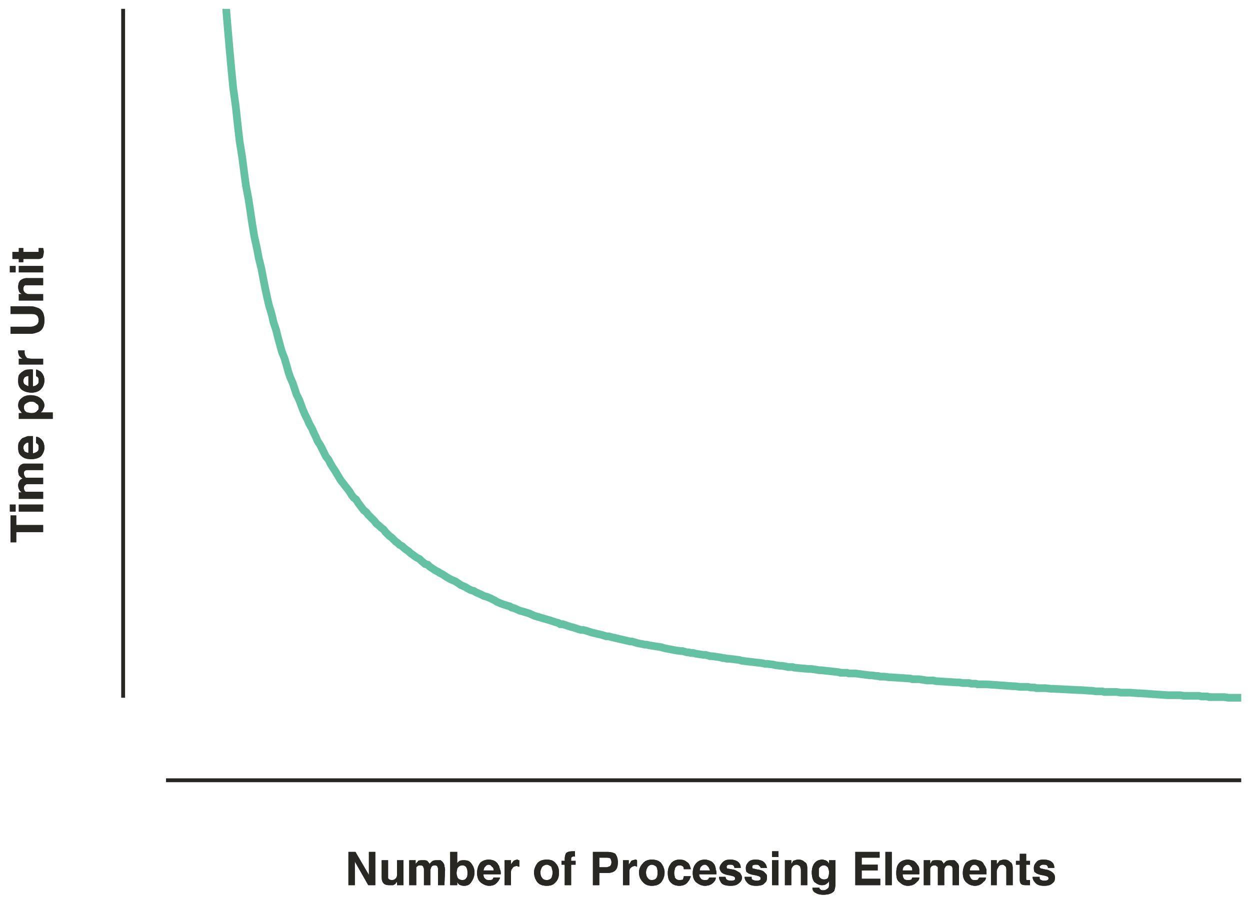
Measuring Weak Scaling







Normalized Weak Scaling



Scaling with More Visual Precision

- Our position statement: rate and efficiency better represent scaling behavior.
- Although neither rate nor efficiency is a new concept, there is not a lot of consistency in the community.
- Through algebra and examples I will show why rate and efficiency are the “right” metrics to use.

Rate

$$R(n,p) = \frac{n}{T(n,p)}$$

Why Use Rate?

$$S(n,p) = \frac{T^*(n)}{T(n,p)}$$

Why Use Rate?

$$S(n,p) = \frac{T^*(n)}{T(n,p)} = \frac{T^*(n)}{n} R(n,p)$$

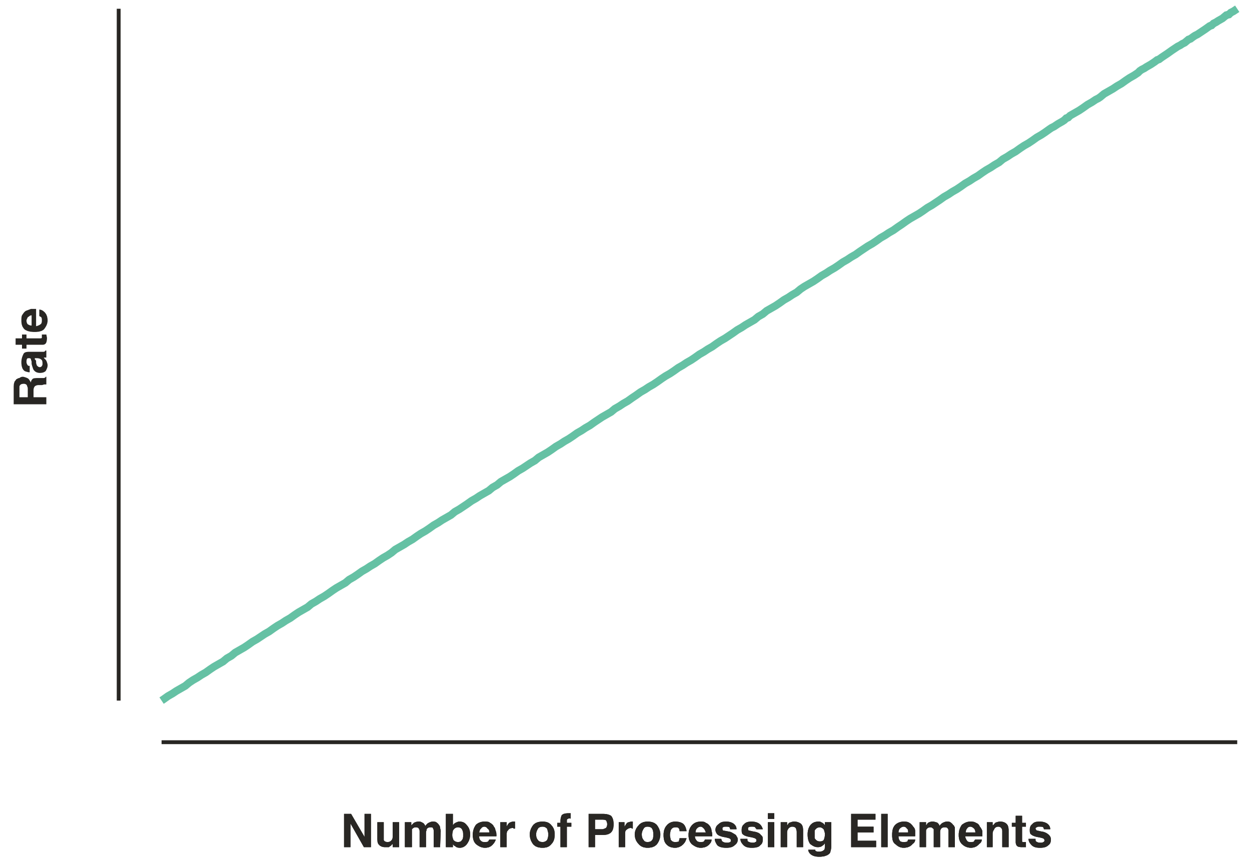
Why Use Rate?

$$S(n,p) = \frac{T^*(n)}{T(n,p)} = \underbrace{\frac{T^*(n)}{n}}_{\text{Becomes a constant with } n \text{ is constant.}} R(n,p)$$

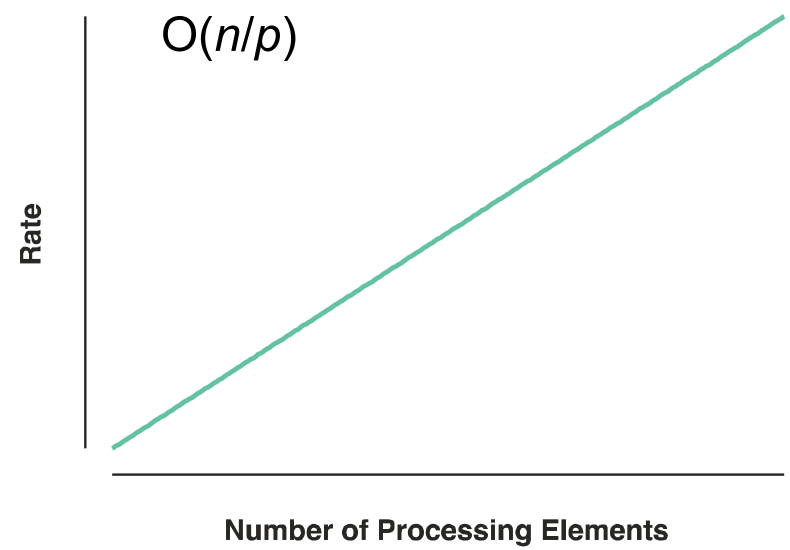
Why Use Rate?

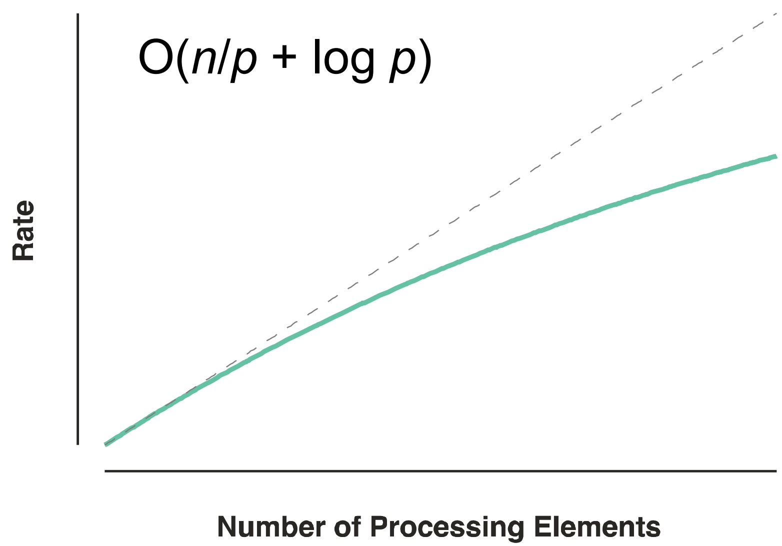
$$S_n(p) \propto R_n(p)$$

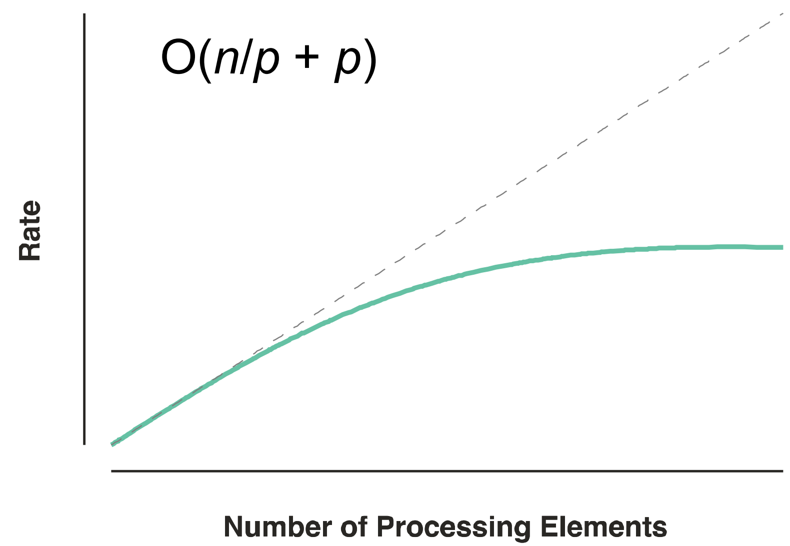
Scaling with Rate



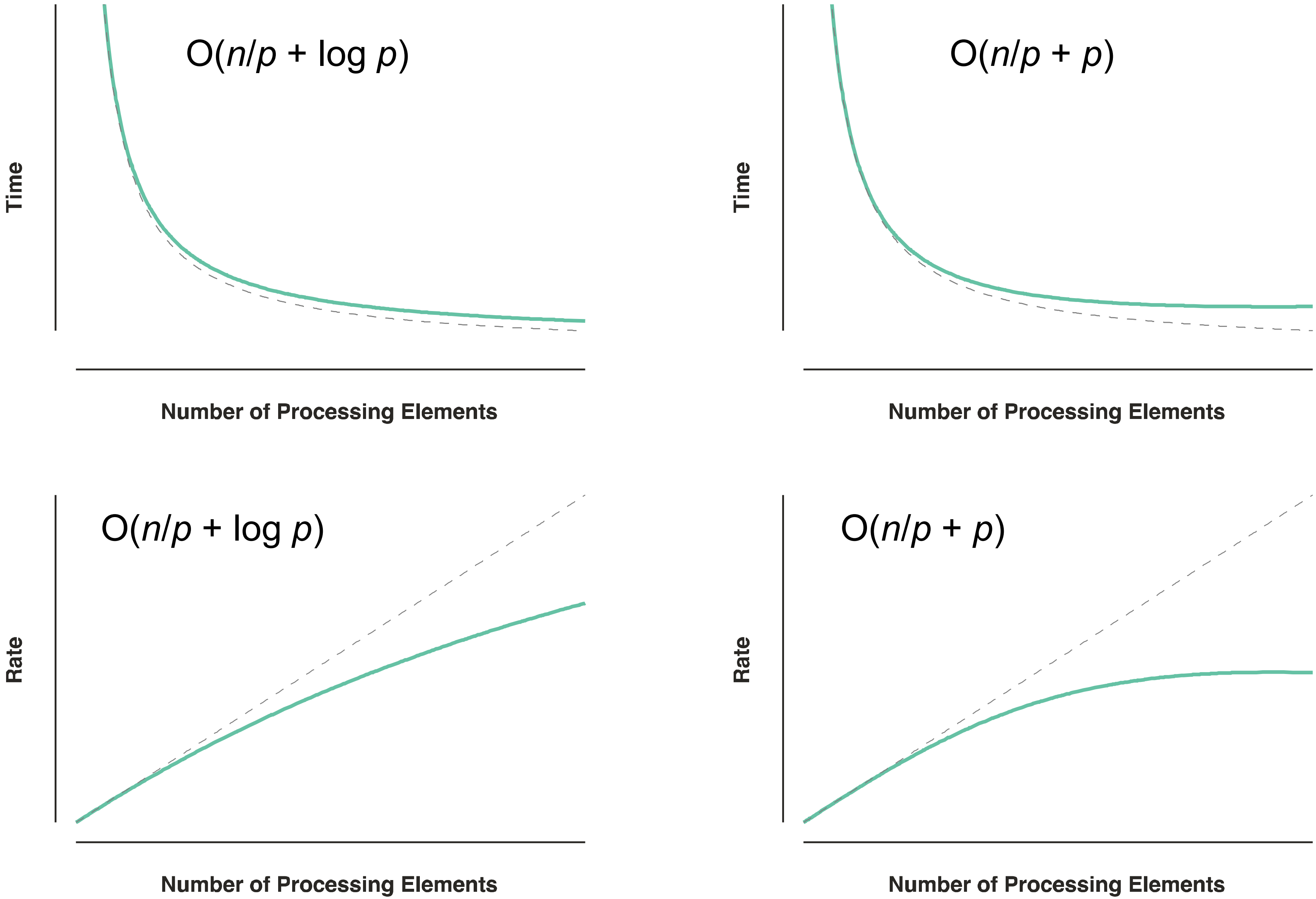
Measuring Scaling with Rate







Measuring Scaling with Rate



Efficiency

$$E(n,p) = \frac{S(n,p)}{p} = \frac{T^*(n)}{p\,T(n,p)}$$

Measuring Efficiency from Cost

Sandia National Laboratories

$$C(n,p) = p\,T(n,p)$$

Measuring Efficiency from Cost

Minimum (best) cost

$$E(n,p) = \frac{C^*(n)}{C(n,p)}$$

Measuring Efficiency from Cost

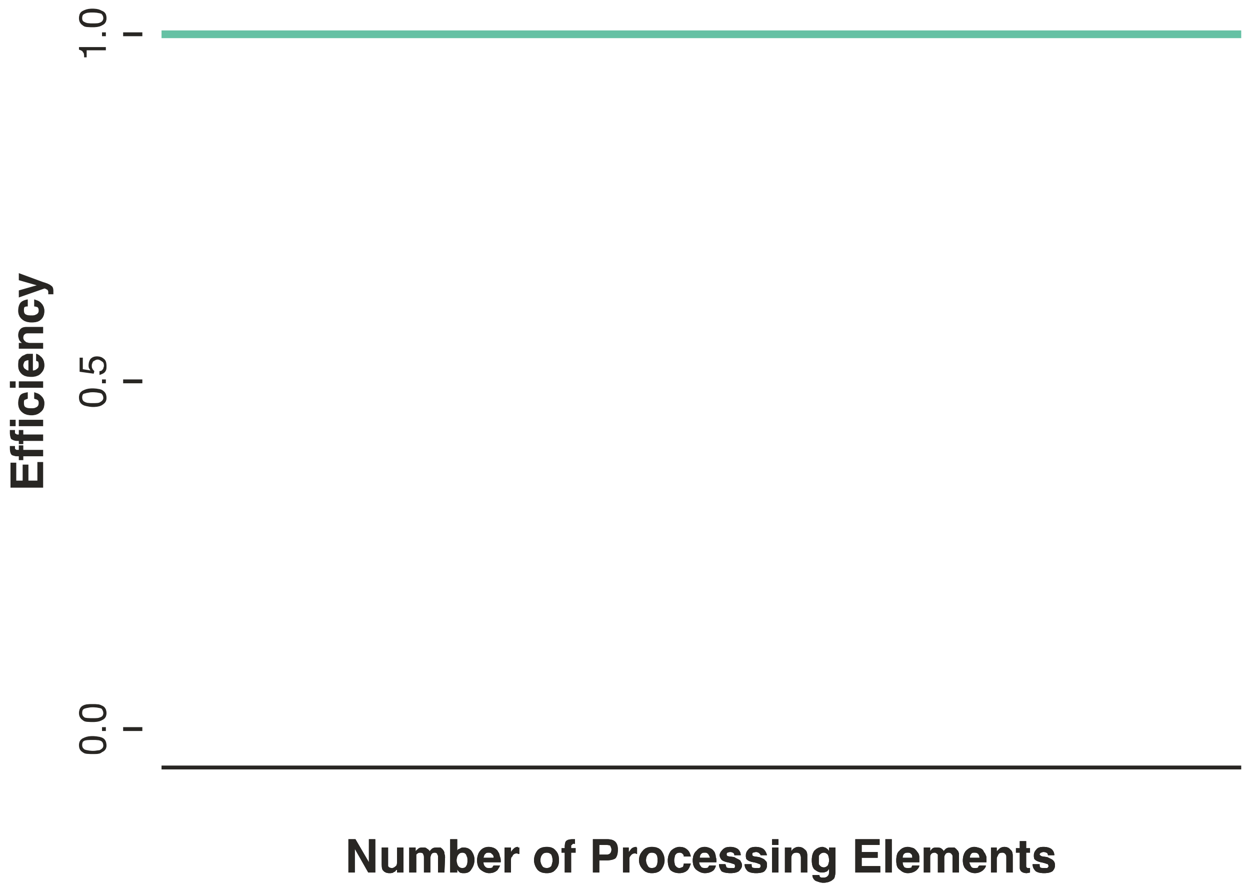
Minimum (best) cost

$$E(n,p) = \frac{C^*(n)}{C(n,p)}$$

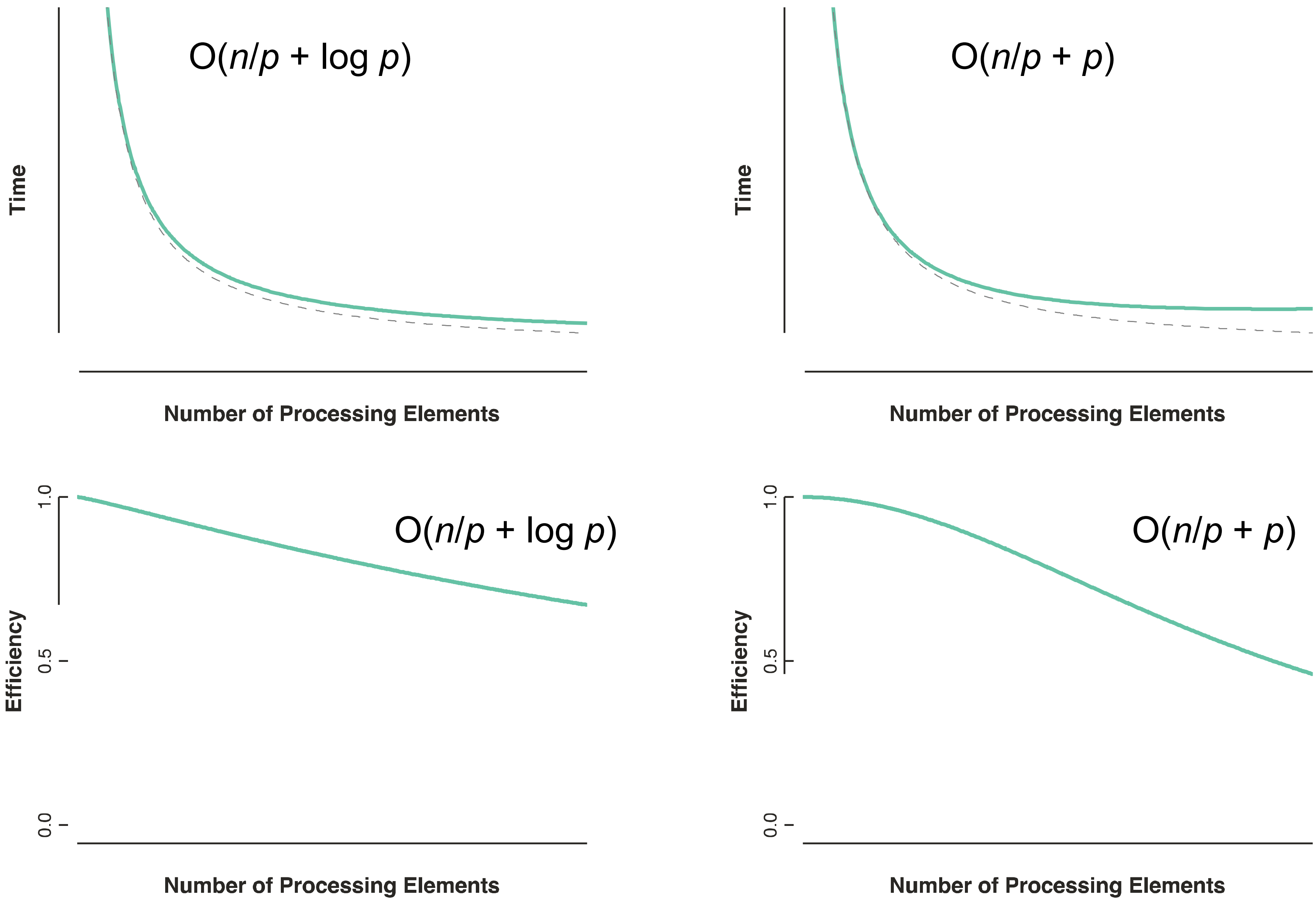
if $C^*(n) = T^*(n)$

$$E(n,p) = \frac{T^*(n)}{C(n,p)} = \frac{T^*(n)}{p\,T(n,p)}$$

Scaling with Efficiency



Measuring Scaling with Efficiency



[This Slide Left Intentionally Blank]

Unifying Strong and Weak Scaling

$$C_u(n,p) = \frac{C(n,p)}{n} = \frac{p\,T(n,p)}{n}$$

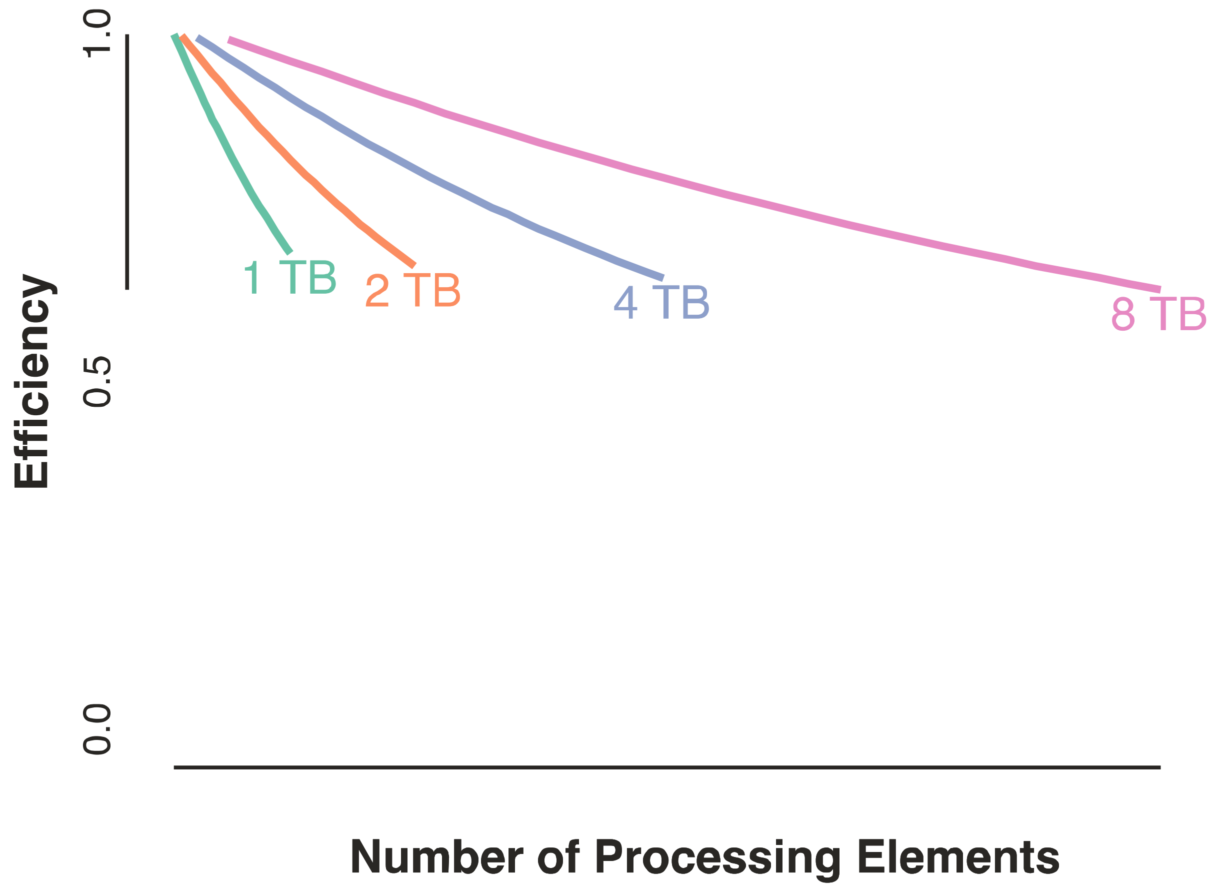
Unifying Strong and Weak Scaling

$$
\begin{aligned}
C_{u,\mathrm{ideal}}(n,p) &= \frac{p\,T(n,p)}{n} \\
&= \frac{p\,O(n/p)}{n} \\
&= O(1)
\end{aligned}
$$

Unifying Strong and Weak Scaling

$$E(n,p) = \frac{C_u^*}{C_u(n,p)}$$

Efficiency Across Data Scales



Unifying Rate Across Data Scales

$$E(n,p) = \frac{C_u^*}{C_u(n,p)}$$

Unifying Rate Across Data Scales

$$E(n,p) = \frac{C_u^*}{C_u(n,p)}$$

$$R(n,p) = \frac{p\ E(n,p)}{C_u^*}$$

Unifying Rate Across Data Scales

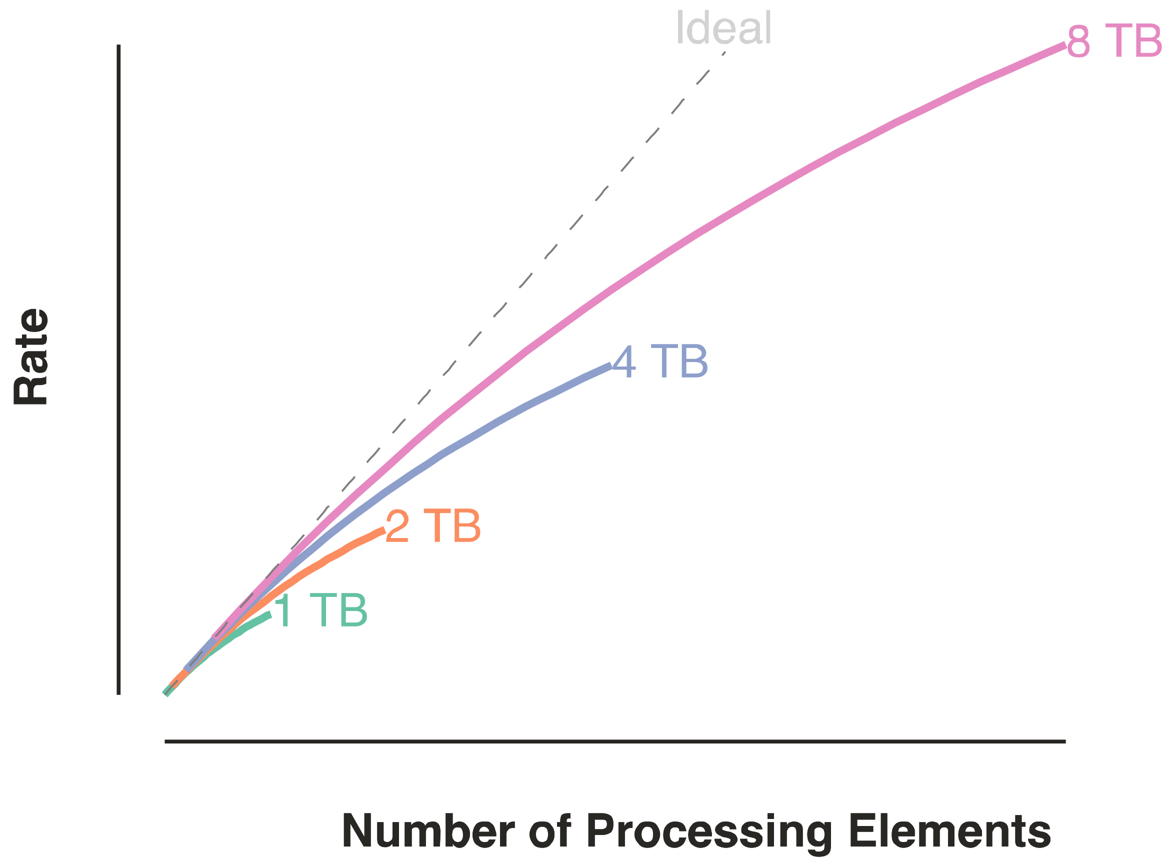
$$E(n,p) = \frac{C_u^*}{C_u(n,p)}$$

$$R(n,p) = \frac{p\ E(n,p)}{C_u^*}$$

$E_{\text{ideal}}(n,p) = 1$

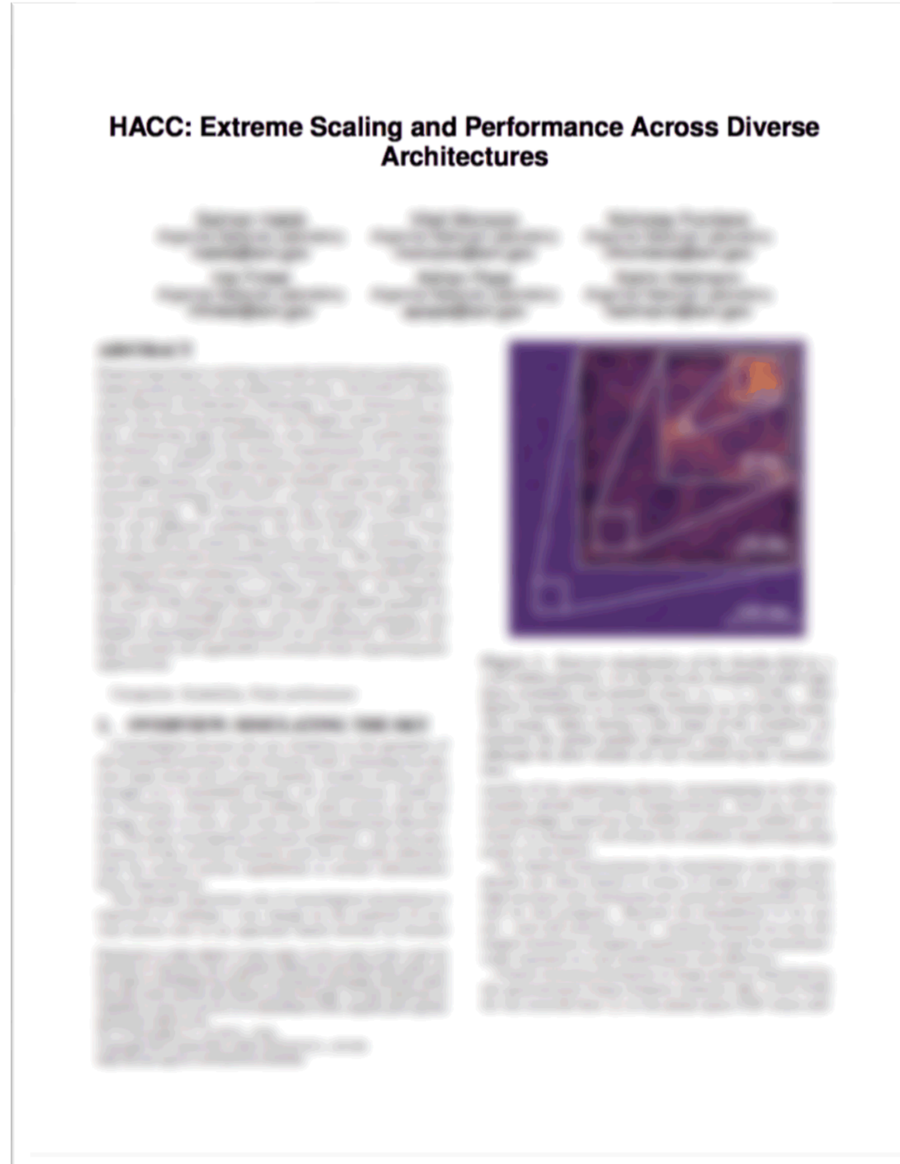
$$R_{\text{ideal}}(n,p) = \frac{p}{C_u^*}$$

Rate Across Data Scales

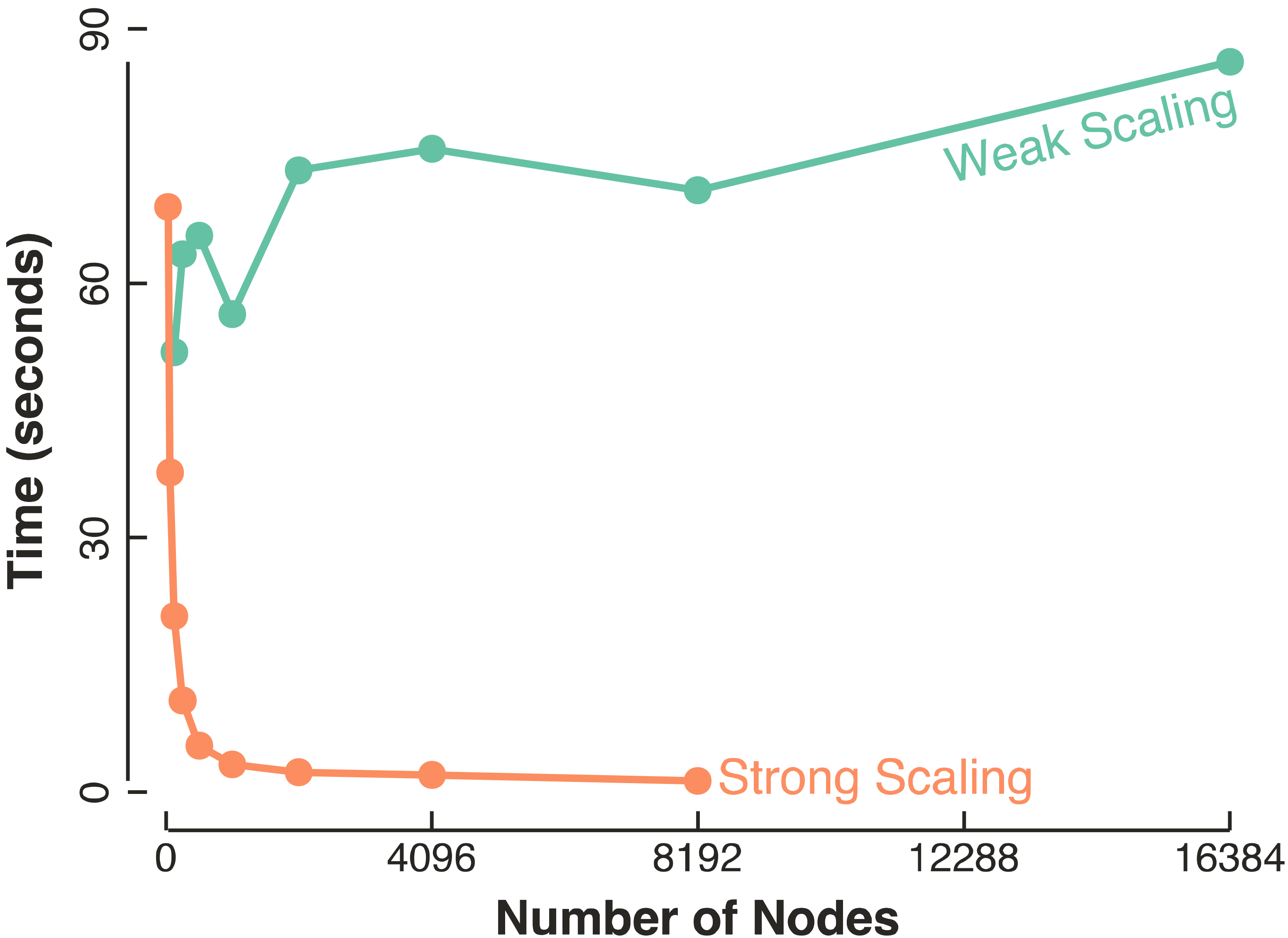


Use Case 1: Gordon Bell Finalist

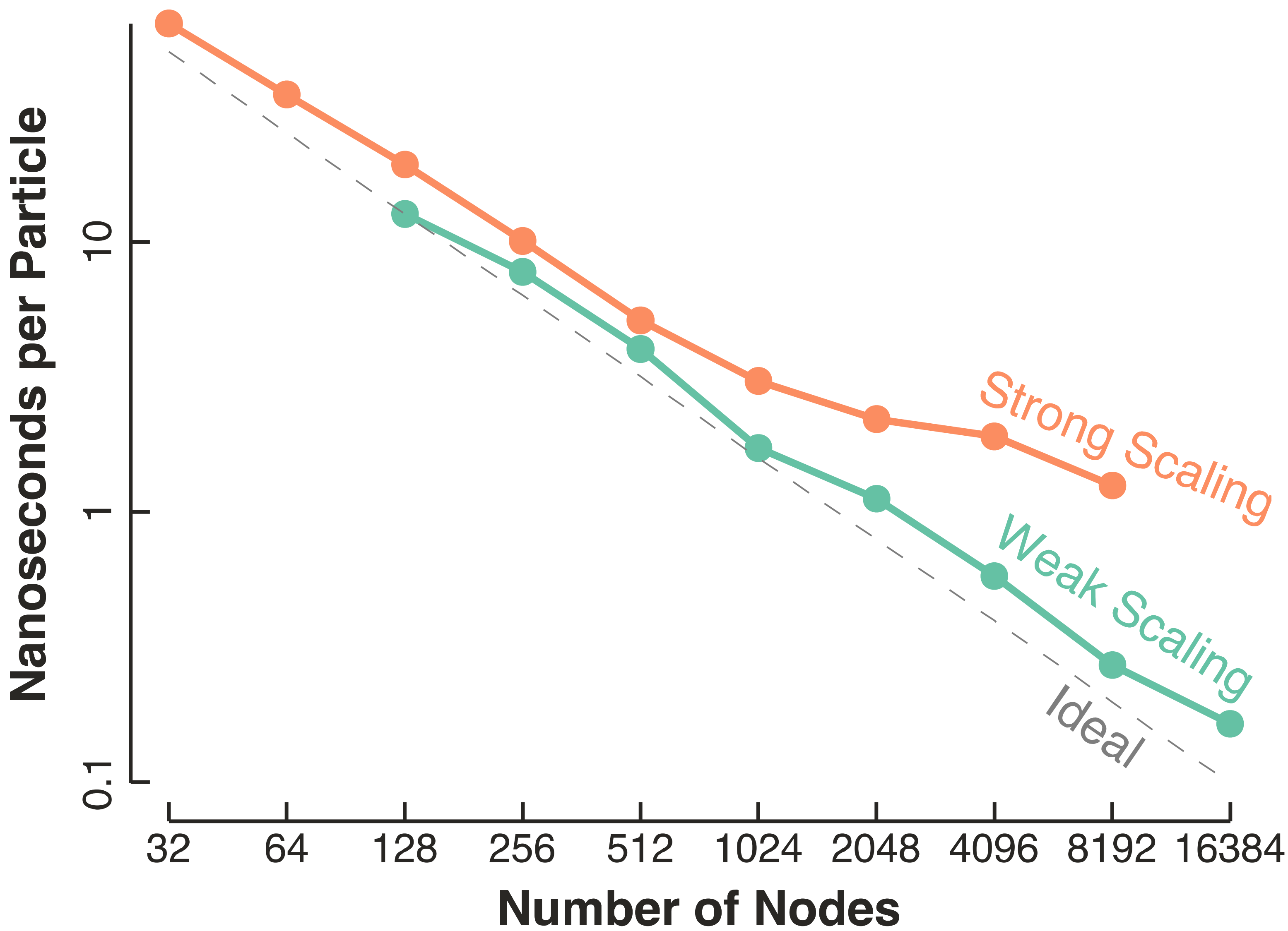
- Measurements of HACC code performance
- Excellent Scalability
- Measurements across many scales
- Lots of data provided in paper


HACC: Extreme Scaling and Performance Across Diverse Architectures

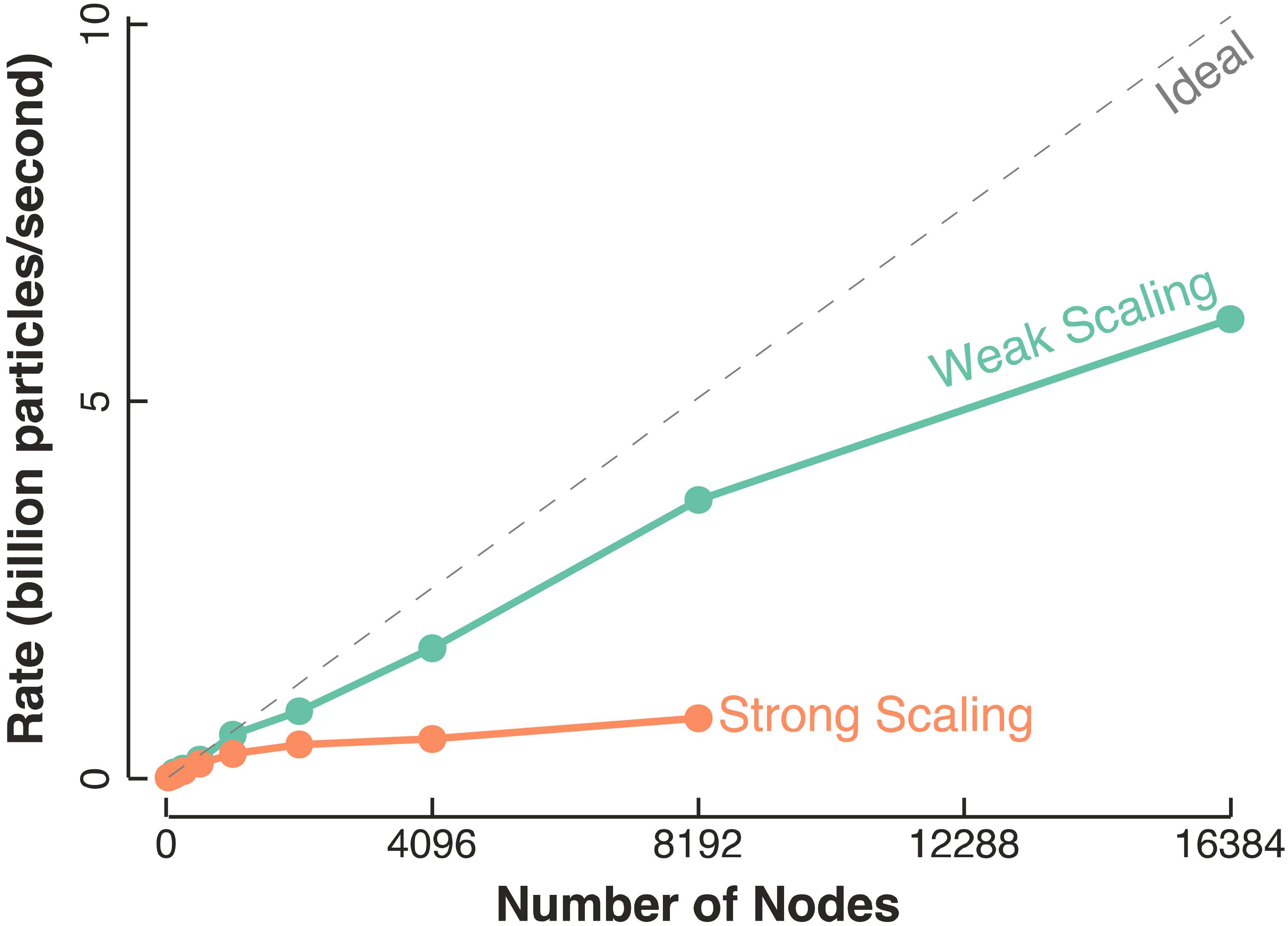
Use Case 1: Gordon Bell Finalist



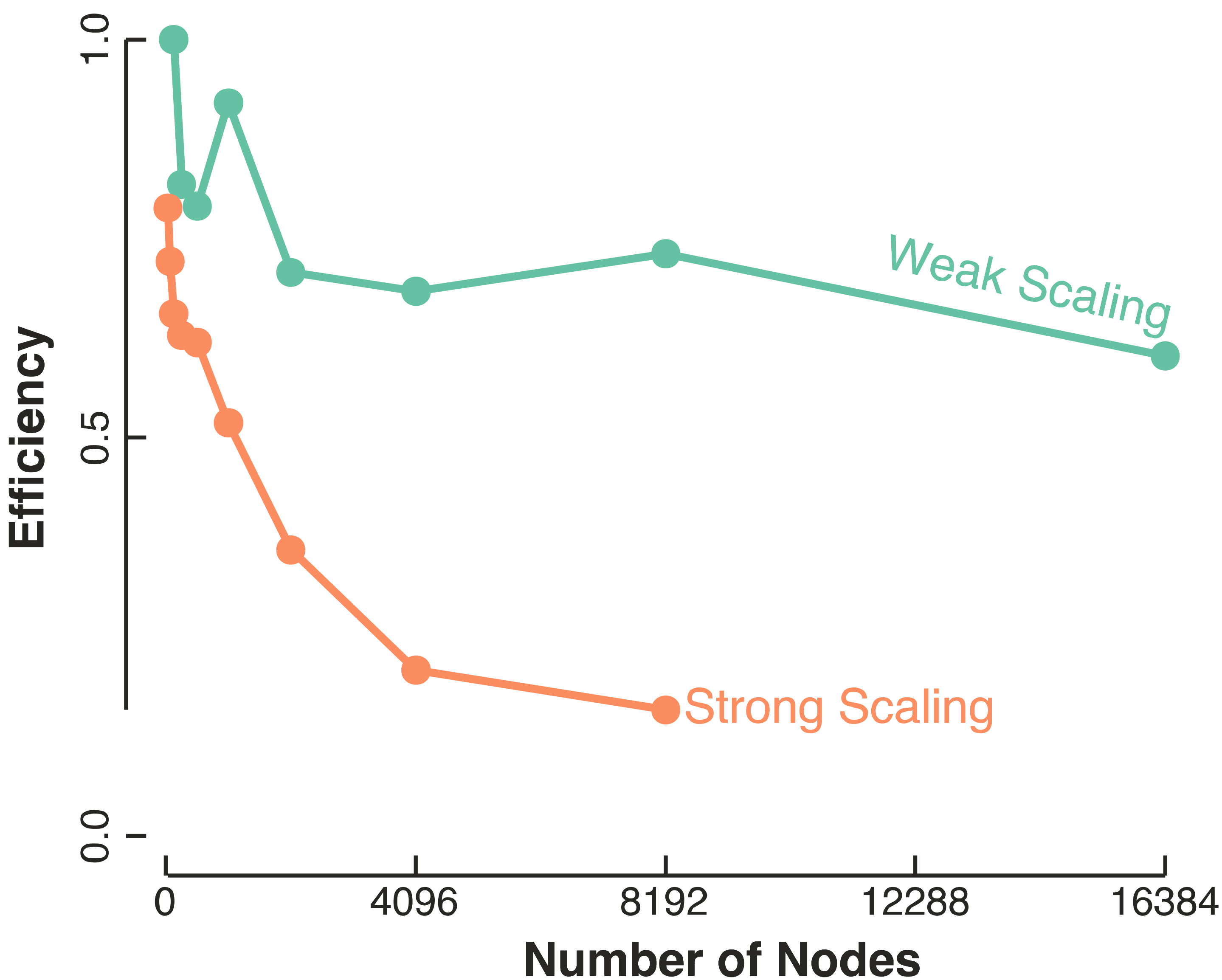
Use Case 1: Gordon Bell Finalist



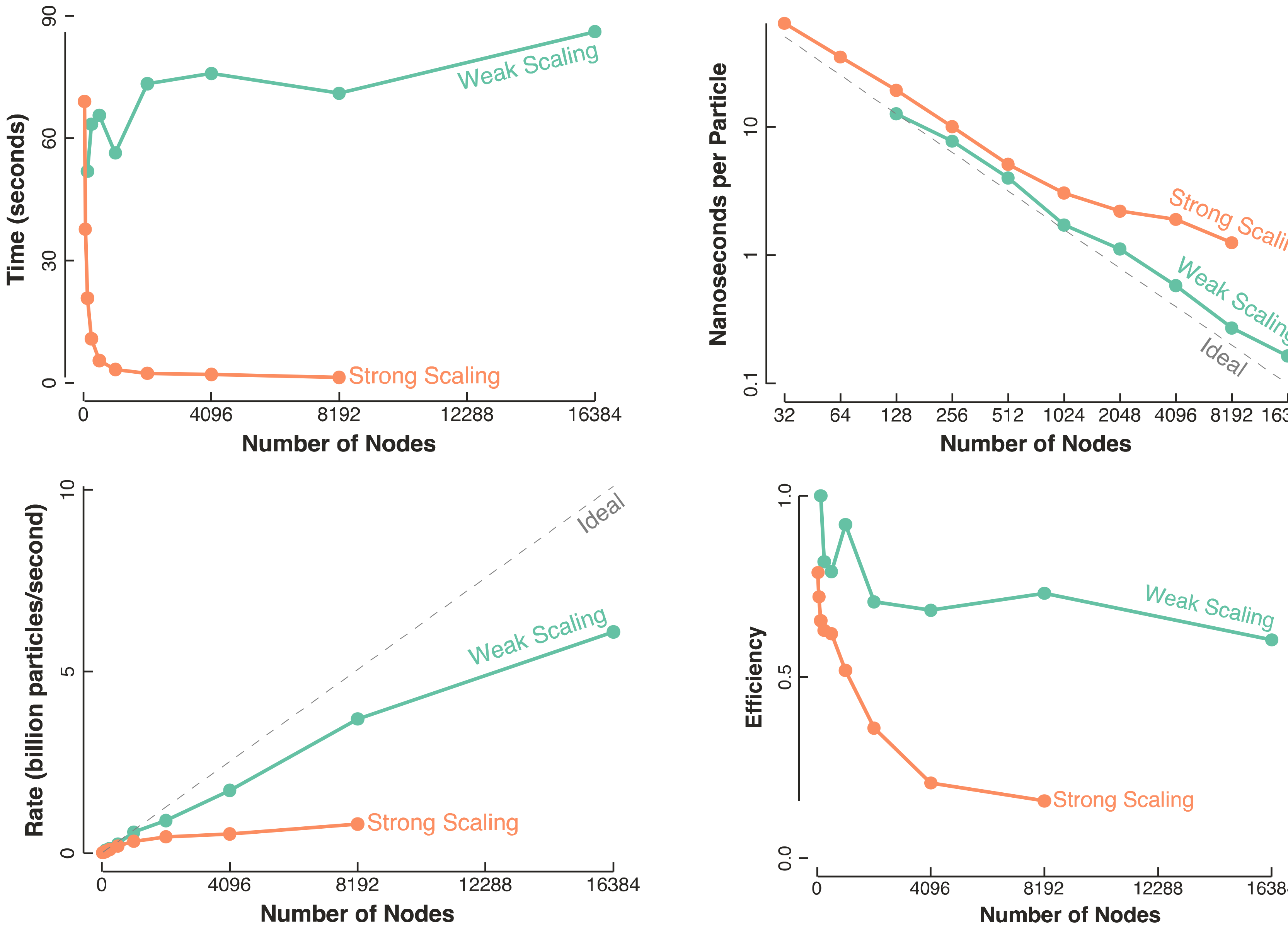
Use Case 1: Gordon Bell Finalist



Use Case 1: Gordon Bell Finalist



Use Case 1: Gordon Bell Finalist



Use Case 2: Imperfect Scaling

- Measures visualization algorithm
- A high communication overhead severely limits scalability

Evaluation of Methods to Integrate Analysis into a Large-Scale Shock Physics Code

Ron A. Oldfield
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1327
raoldfi@sandia.gov

Kenneth Moreland
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1326
kmorel@sandia.gov

Nathan Fabian
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-1326
ndfabia@sandia.gov

David Rogers
Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, NM 87545
dhr@lanl.gov

ABSTRACT

Exascale supercomputing will embody many revolutionary changes in the hardware and software of high-performance computing. For example, projected limitations in power and I/O-system performance will fundamentally change visualization and analysis workflows. A traditional post-processing workflow involves storing simulation results to disk and later retrieving them for visualization and data analysis; however, at Exascale, post-processing approaches will not be able to capture the volume or granularity of data necessary for analysis of these extreme-scale simulations. As an alternative, researchers are exploring ways to integrate analysis and simulation without using the storage system. In situ and in transit are two options, but there has not been an adequate evaluation of these approaches to identify strengths, weaknesses, and trade-offs at large scale. This paper provides a detailed performance and scaling analysis of a large-scale shock physics code using traditional post-processsing, in situ, and in transit analysis to detect material fragments from a simulated explosion.

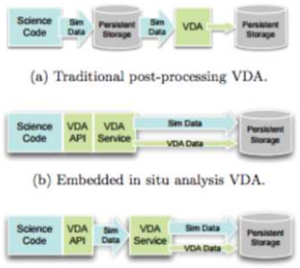
Categories and Subject Descriptors

I.6.6 [**Simulation Output Analysis**]; H.3.4 [**Systems and Software**]: Performance evaluation (efficiency and effectiveness)

Keywords

Case study, fragment detection, in situ analysis, in transit analysis, shock physics

ICS'14, June 10–13 2014, Munich, Germany.
ACM 978-1-4503-2642-1/14/06.
http://dx.doi.org/10.1145/2597652.2597668..



(a) Traditional post-processing VDA.

(b) Embedded in situ analysis VDA.

(c) Service-oriented in transit VDA.

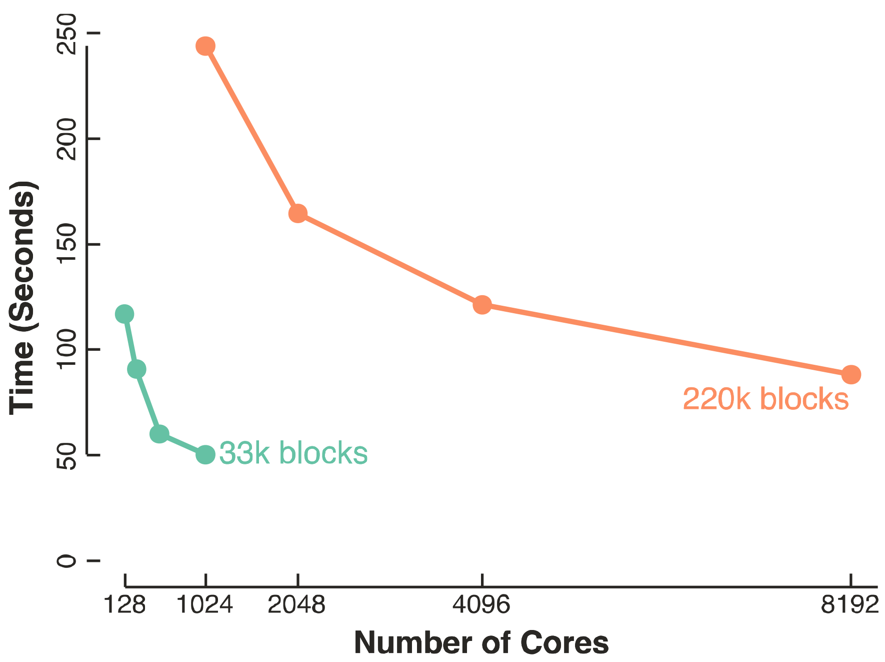
Figure 1: Traditional and emerging workflow diagrams showing the flow of information from simulation to persistent storage.

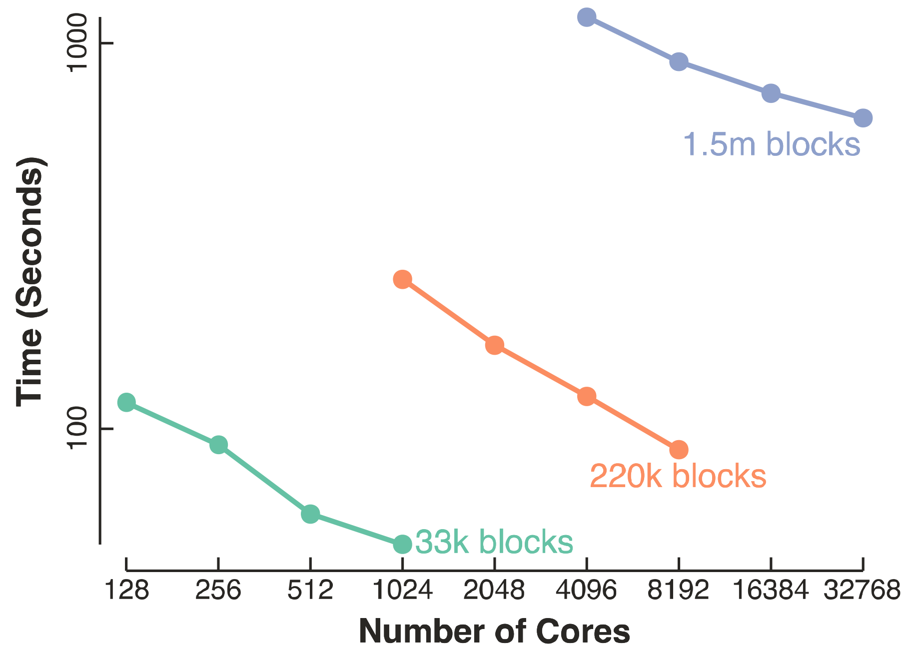
1. INTRODUCTION

High-performance computing (HPC) applications produce complex datasets that are increasingly difficult to explore and understand using traditional post-processing workflows. The primary reason is the increasing gap between computation and communication performance and the performance of parallel file systems. This gap has been a known problem for several decades [13,15] and has motivated numerous innovations to improve parallelism [16,31], caching [32], processing [33], and scheduling [7] in I/O systems. Despite these innovations, the gap widens at an alarming rate. At Exascale, with a projected storage system rate of 60 TB/s [3], I/O system throughput will be less than 1% of the generating capacity of an HPC simulation. These trends are driving an evolution away from application workflows consisting of sequences of independent simulation and analysis steps to integrated approaches that perform these steps concurrently.

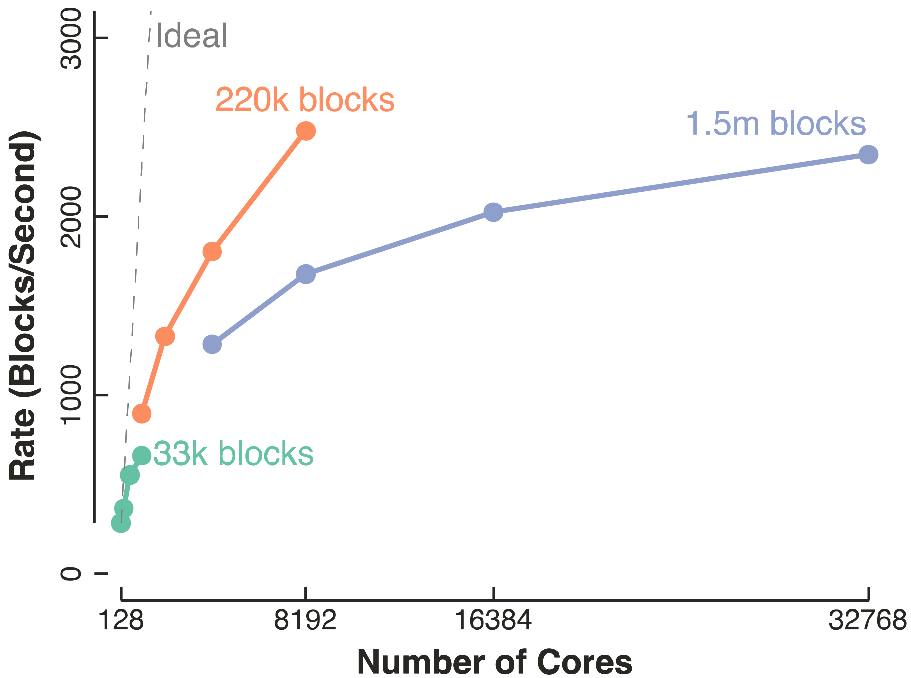
This paper provides a comprehensive evaluation of three approaches (see Figure 1) to integrate visualization and data

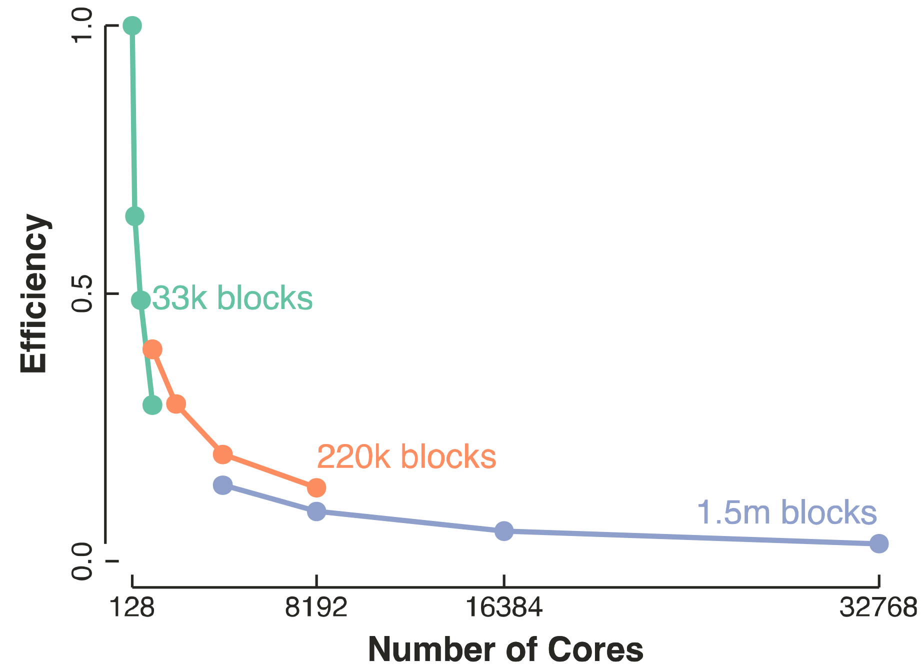
Use Case 2: Imperfect Scaling



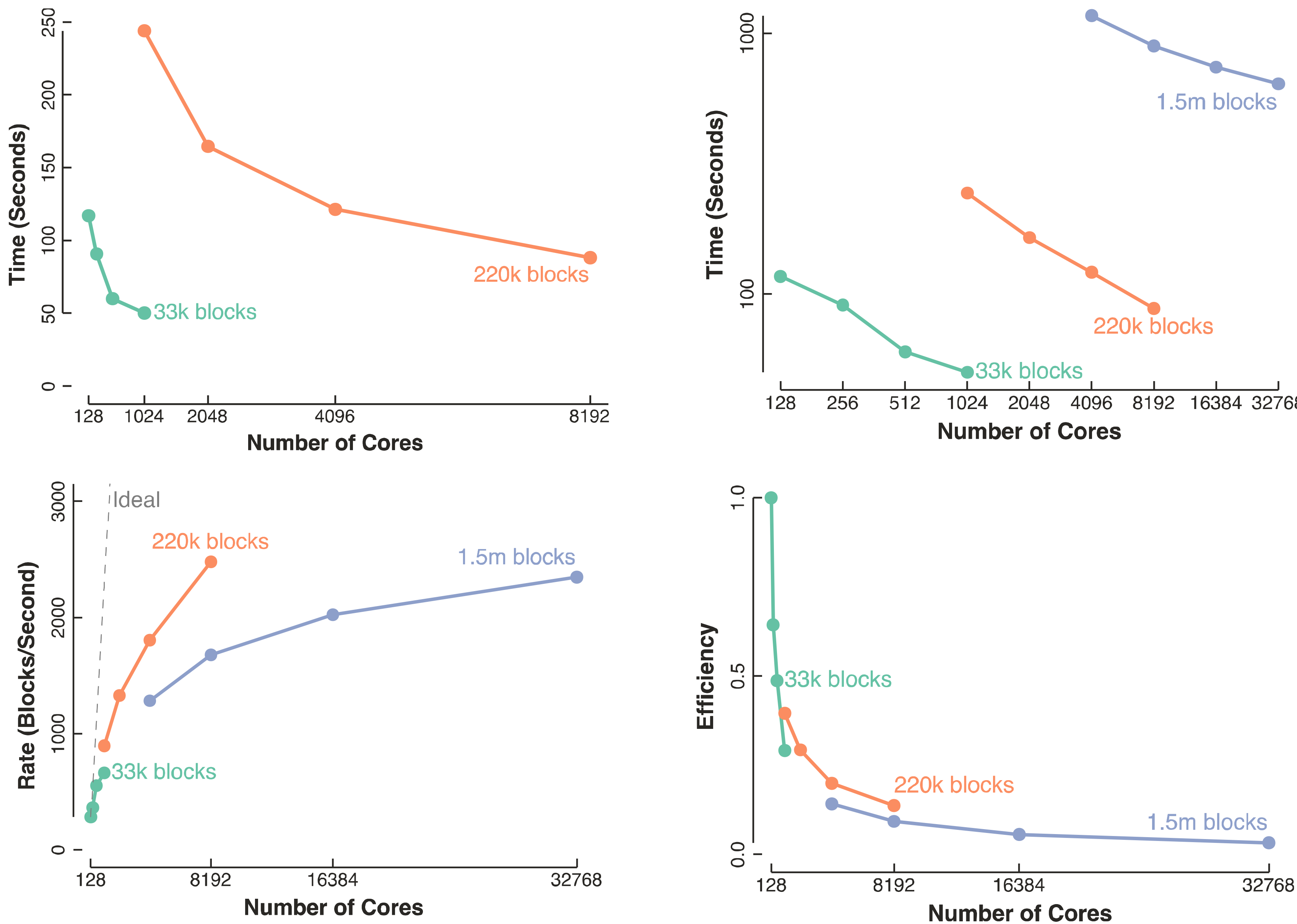


Use Case 2: Imperfect Scaling





Use Case 2: Imperfect Scaling



Final Recommendations

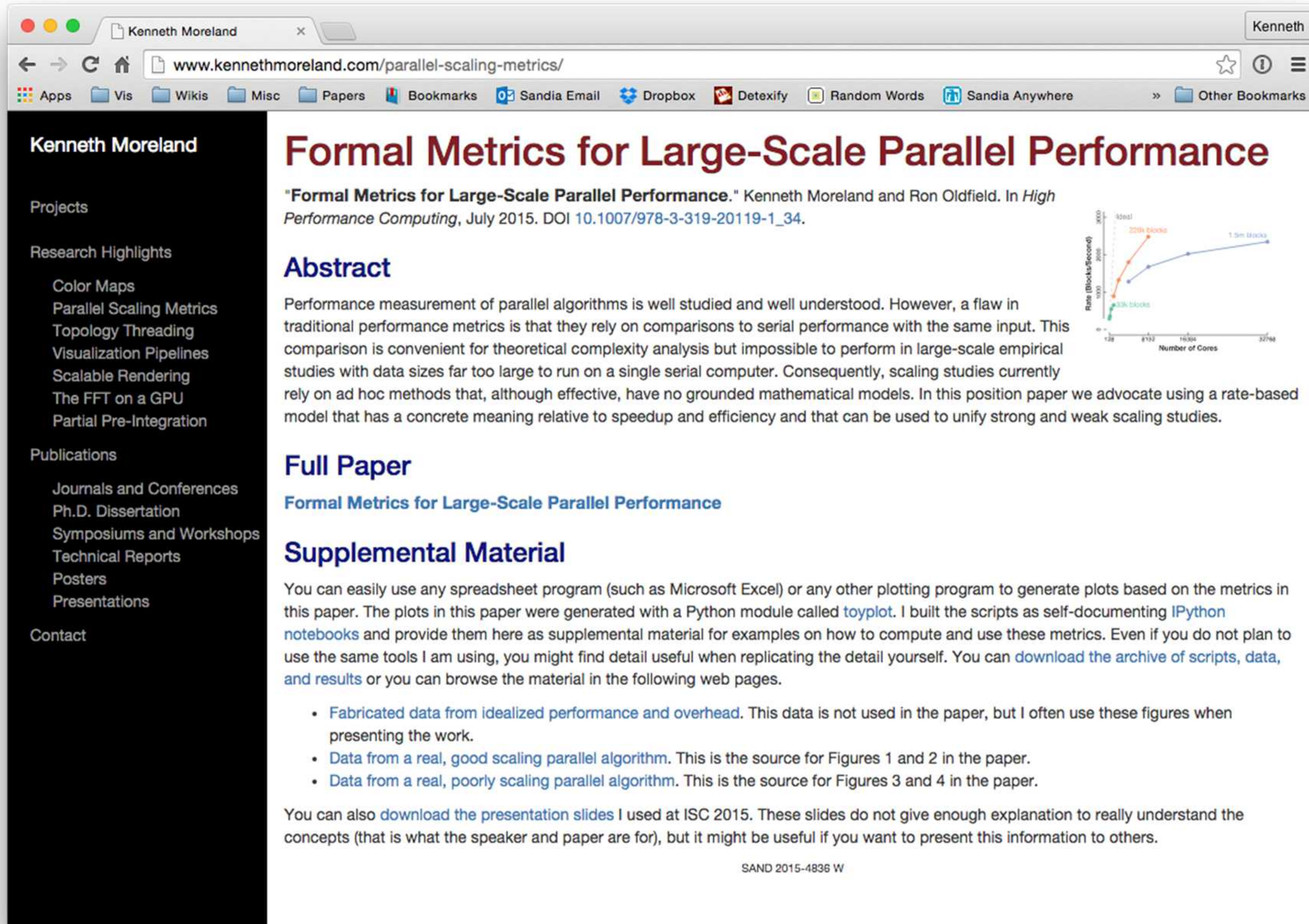
- **Do not rely on running time** for performance analysis. Instead use rate, efficiency, or both.
- **Avoid using log-log scaling** on plot axes, which hides major inefficiencies. If necessary, repeat linear plots at different scales.
- Rather than performing them separately, **incorporate weak and strong scaling studies in one**. Perform several strong scaling studies at different scales of data size. Then find an overall minimal practical cost per unit and plot all the measurements together as demonstrated in the figures in this paper.

Acknowledgements

- This material is based in part upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Scientific Discovery through Advanced Computing (SciDAC) program under Award Number 12-015215.
- This material is based in part upon work supported by the U.S. Department of Energy, National Nuclear Security Administration, Advanced Simulation and Computing (ASC).

More Resources

http://www.kennethmoreland.com/parallel-scaling-metrics/



Kenneth Moreland

www.kennethmoreland.com/parallel-scaling-metrics/

Apps Vis Wikis Misc Papers Bookmarks Sandia Email Dropbox Detexify Random Words Sandia Anywhere Other Bookmarks

Kenneth Moreland

Projects

Research Highlights

- Color Maps
- Parallel Scaling Metrics
- Topology Threading
- Visualization Pipelines
- Scalable Rendering
- The FFT on a GPU
- Partial Pre-Integration

Publications

- Journals and Conferences
- Ph.D. Dissertation
- Symposiums and Workshops
- Technical Reports
- Posters
- Presentations

Contact

Formal Metrics for Large-Scale Parallel Performance

"**Formal Metrics for Large-Scale Parallel Performance.**" Kenneth Moreland and Ron Oldfield. In *High Performance Computing*, July 2015. DOI 10.1007/978-3-319-20119-1_34.

Abstract

Performance measurement of parallel algorithms is well studied and well understood. However, a flaw in traditional performance metrics is that they rely on comparisons to serial performance with the same input. This comparison is convenient for theoretical complexity analysis but impossible to perform in large-scale empirical studies with data sizes far too large to run on a single serial computer. Consequently, scaling studies currently rely on ad hoc methods that, although effective, have no grounded mathematical models. In this position paper we advocate using a rate-based model that has a concrete meaning relative to speedup and efficiency and that can be used to unify strong and weak scaling studies.

Full Paper

Formal Metrics for Large-Scale Parallel Performance

Supplemental Material

You can easily use any spreadsheet program (such as Microsoft Excel) or any other plotting program to generate plots based on the metrics in this paper. The plots in this paper were generated with a Python module called toyplot. I built the scripts as self-documenting IPython notebooks and provide them here as supplemental material for examples on how to compute and use these metrics. Even if you do not plan to use the same tools I am using, you might find detail useful when replicating the detail yourself. You can download the archive of scripts, data, and results or you can browse the material in the following web pages.

- Fabricated data from idealized performance and overhead. This data is not used in the paper, but I often use these figures when presenting the work.
- Data from a real, good scaling parallel algorithm. This is the source for Figures 1 and 2 in the paper.
- Data from a real, poorly scaling parallel algorithm. This is the source for Figures 3 and 4 in the paper.

You can also download the presentation slides I used at ISC 2015. These slides do not give enough explanation to really understand the concepts (that is what the speaker and paper are for), but it might be useful if you want to present this information to others.

SAND 2015-4836 W