# Amdahl's Law in the Datacenter Era

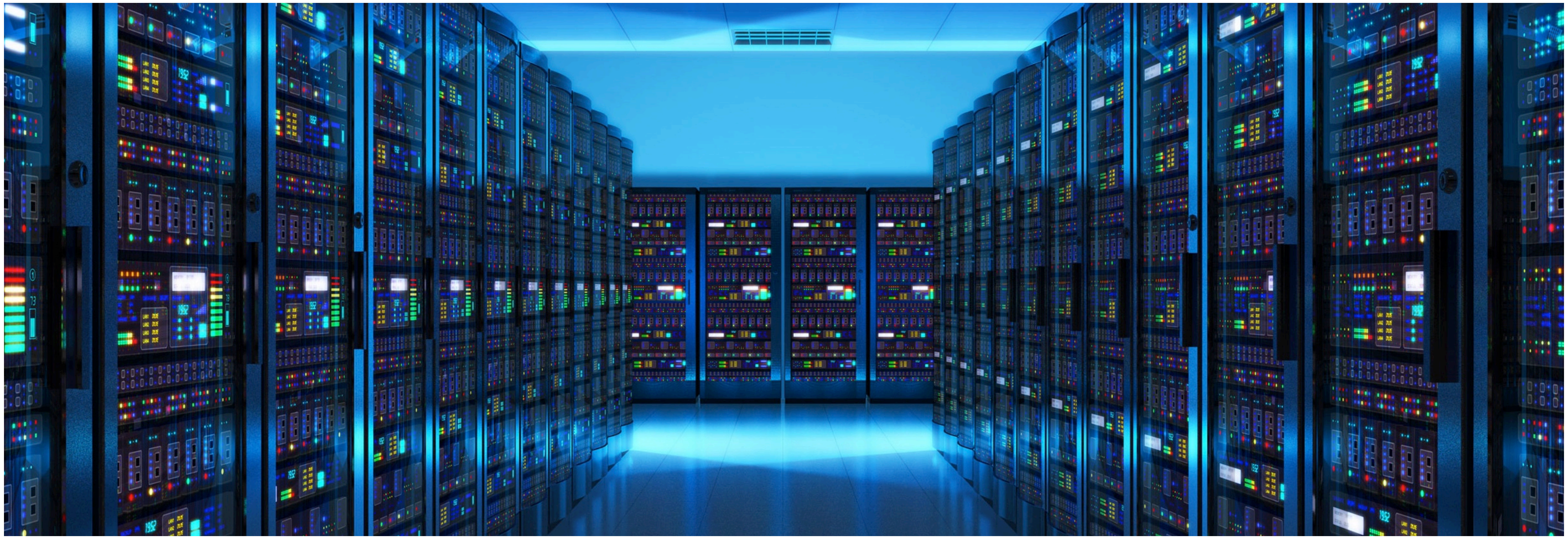
## A Market for Fair Processor Allocation

Seyed Majid Zahedi* (Duke University), Qiuyun Llull* (VMware/ Duke University), Benjamin C. Lee (Duke University)

*Equal Contributions

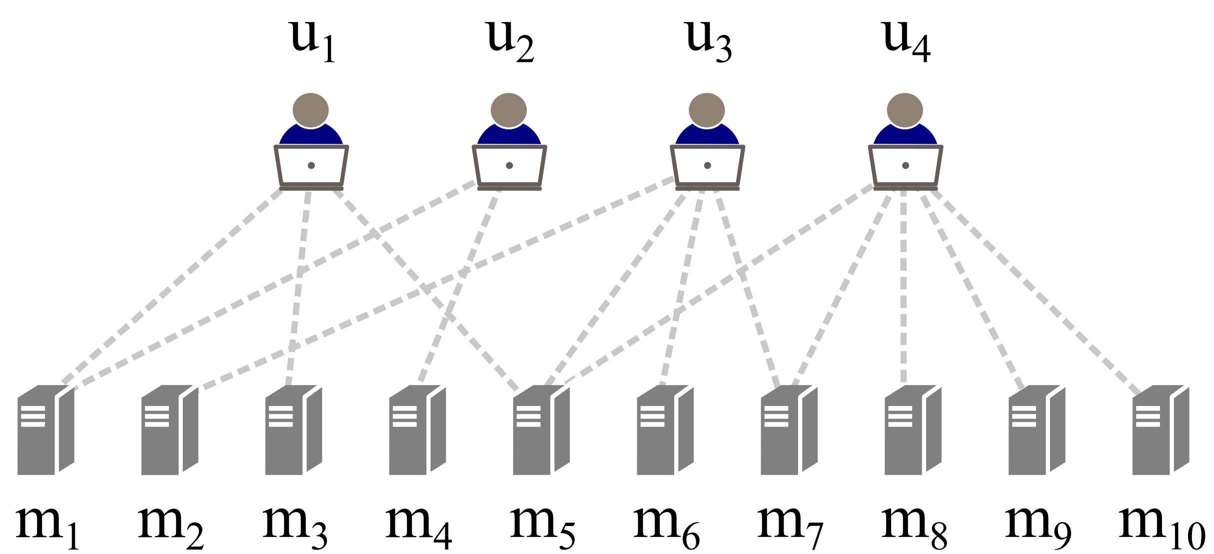
Duke
UNIVERSITY

# Sharing in Federated Data Centers



- Users **pool** resources in non-profit data centers
  - E.g., research groups within university
- Users are **entitled** to portion of resources
  - Based on contributions to shared pool

# Challenges for Modern Data Centers




- Computing resources are physically **distributed** across servers
- Users run complex jobs with **diverse** characteristics
- Users' jobs are assigned to **different** servers
- Users prefer **specific** allocations on specific servers

# Management Properties

- **Work Conservation**
  - Never leave servers **idle** if there are unsatisfied user demands
- **Sharing Incentives**
  - Guarantee users **at least** the utility from their **entitlements**

$$u_i(\mathbf{x_i}) \ \geq \ u_i(\mathbf{e_i})$$

# Management Questions

- **How can we model users' demands for processors?**
- **How can we fairly allocate processors?**



Fig. downtownrecruiting.com

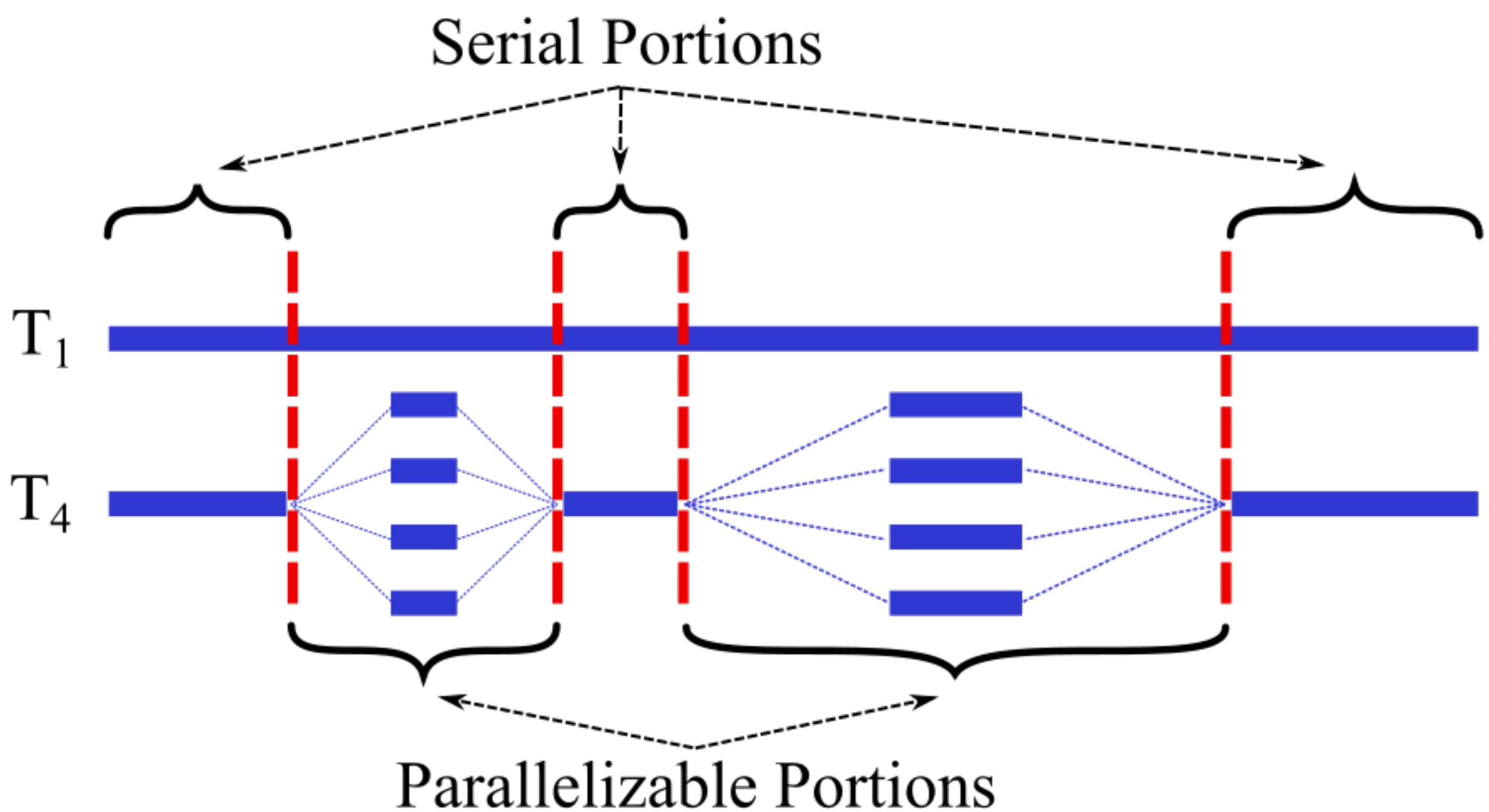
# Roadmap

- **Model user utilities**
  - Operationalize Amdahl's Law for data center workloads
  - Propose Amdahl utility using Karp-Flatt metric
- **Design market mechanism**
  - Design market for processor allocation
  - Propose Amdahl bidding procedure using closed-form equations
  - Find market equilibrium to guarantee fair division
- **Conclude**

# Amdahl's Law

[G. Amdahl 1967]

- Architects use it to estimate upper bounds on speedups




$$\text{Speedup}(x) = \frac{T_1}{T_x} = \frac{T_1}{(1-F)T_1 + F\frac{T_1}{x}} = \frac{x}{x(1-F)+F}$$

# What Portion of Code is Parallelizable?

[Allen Karp and Horace Flatt 1990]

- Expert programmers may not know!
- Fortunately, we can measure speedup

Measure  Set  Calculate

$$s(x) = \frac{x}{x(1-F)+F}$$

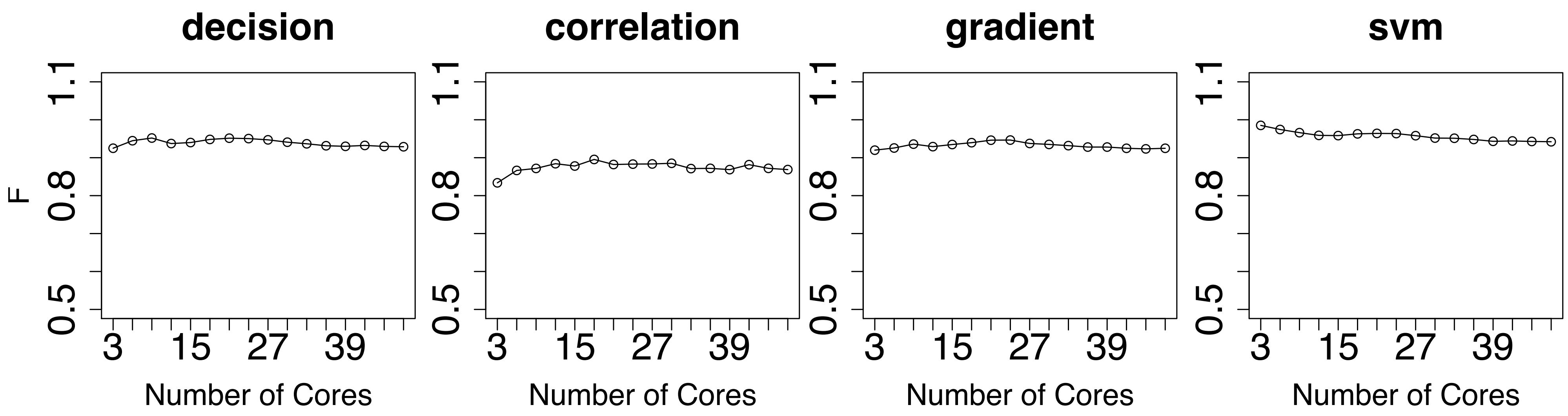
Karp-Flatt Metric

$$F = \left(1 - \frac{1}{s(x)}\right)\left(1 - \frac{1}{x}\right)$$

?

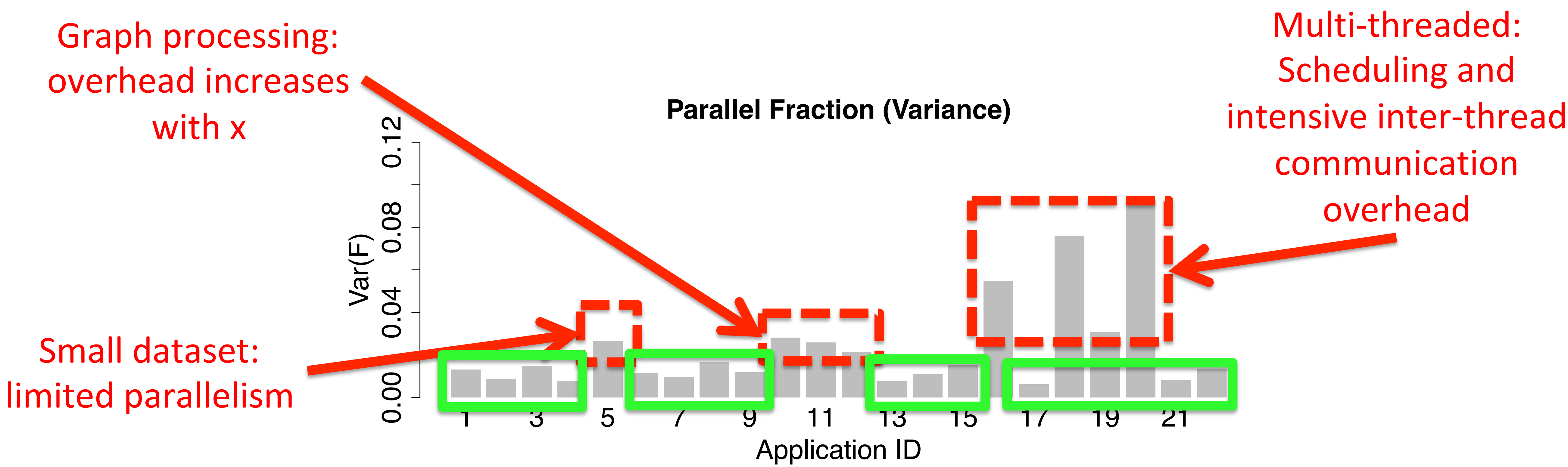
Fig. criticallyrated.files.wordpress.com

# Karp-Flatt Metric in Practice



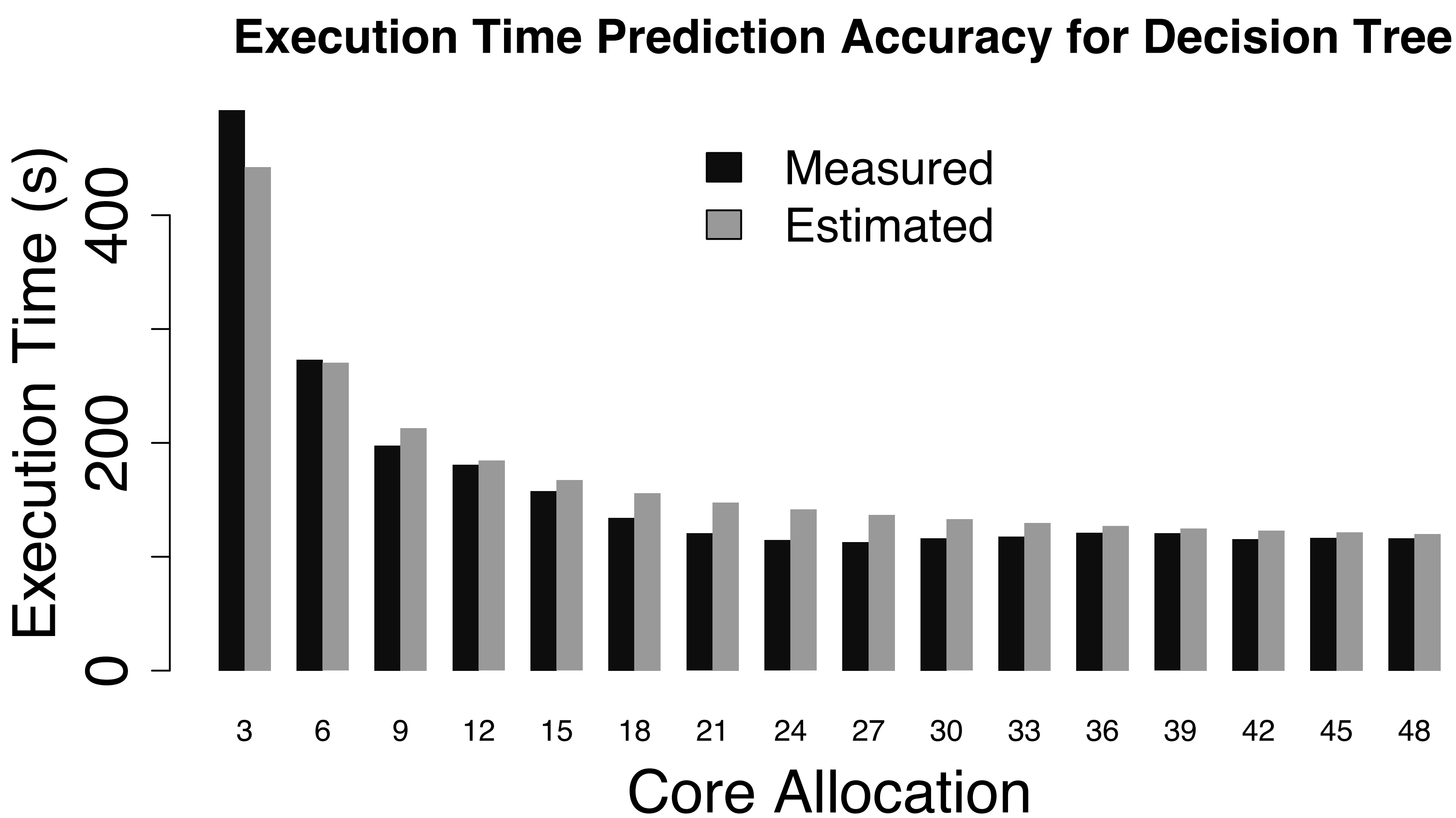

- For many Spark and PARSEC workloads, Karp-Flatt has **low variance**
  - Abundant, fine-grained parallelism
  - Few serial bottlenecks
- Constant Karp-Flatt metric indicates **accuracy** of Amdahl's Law

# Limitations of Karp-Flatt Metric




- There are some **exceptions**
  - High **correlation** between serial and parallel portion
  - High **scheduling** and inter-thread **communication**
  - Very **limited** parallelism

# Amdahl's Law in Practice




- Measured performance **tracks** estimated speedup
  - Amdahl's Law can drive processor allocation

# Amdahl Utility

- Measures normalized progress across servers

Work completed for user *i* in unit of time on one core of server *j*

$$\text{Speedup}_{ij}(x_{ij}) = \frac{x_{ij}}{F_{ij} + (1 - F_{ij})x_{ij}}$$

$$u_i(\mathbf{x_i}) = \frac{\sum_{j=1}^{m} w_{ij} s_{ij}(x_{ij})}{\sum_{j=1}^{m} w_{ij}}$$

# Roadmap

- **Model user utilities**
  - Operationalize Amdahl's Law for data center workloads
  - Propose Amdahl utility using Karp-Flatt metric
- **Design market mechanism**
  - Design market for processor allocation
  - Propose Amdahl bidding procedure using closed-form equations
  - Find market equilibrium to guarantee fair division
- **Conclude**

# Market for Fair Allocation

- Users receive **budgets** in proportion to their entitlements
- Market sets **prices** for processors on each server
- Users **demand** processors that maximize their utility

$$\begin{aligned} \text{max.} \quad & u_i(\mathbf{x}_i), \\ \text{s.t.} \quad & \sum_{j=1}^{m} x_{ij} p_j \leq b_i \end{aligned}$$

- At **equilibrium prices**, market **clears**

$$\sum_{i=1}^{n} x_{ij} = C_j$$

# Amdahl Bidding Procedure


- Users iteratively bid for processors using closed-form equation

$$b_{ij}(t+1) \propto \sqrt{f_{ij}\ p_j(t)}\ w_{ij} s_{ij}(x_{ij}(t))$$

- Market sets prices based on bids

$$p_j(t) = \sum_{i=1}^{n} b_{ij} / C_j$$

- Iterate until prices are stationary



Fig. searchengineland.com

# Properties of Amdahl Bidding Procedure

- Allocations are work-conserving
- Market guarantees sharing incentives
- Users bid truthfully in large, competitive systems
- Market does all of these with low overhead

# Mechanisms for Evaluation

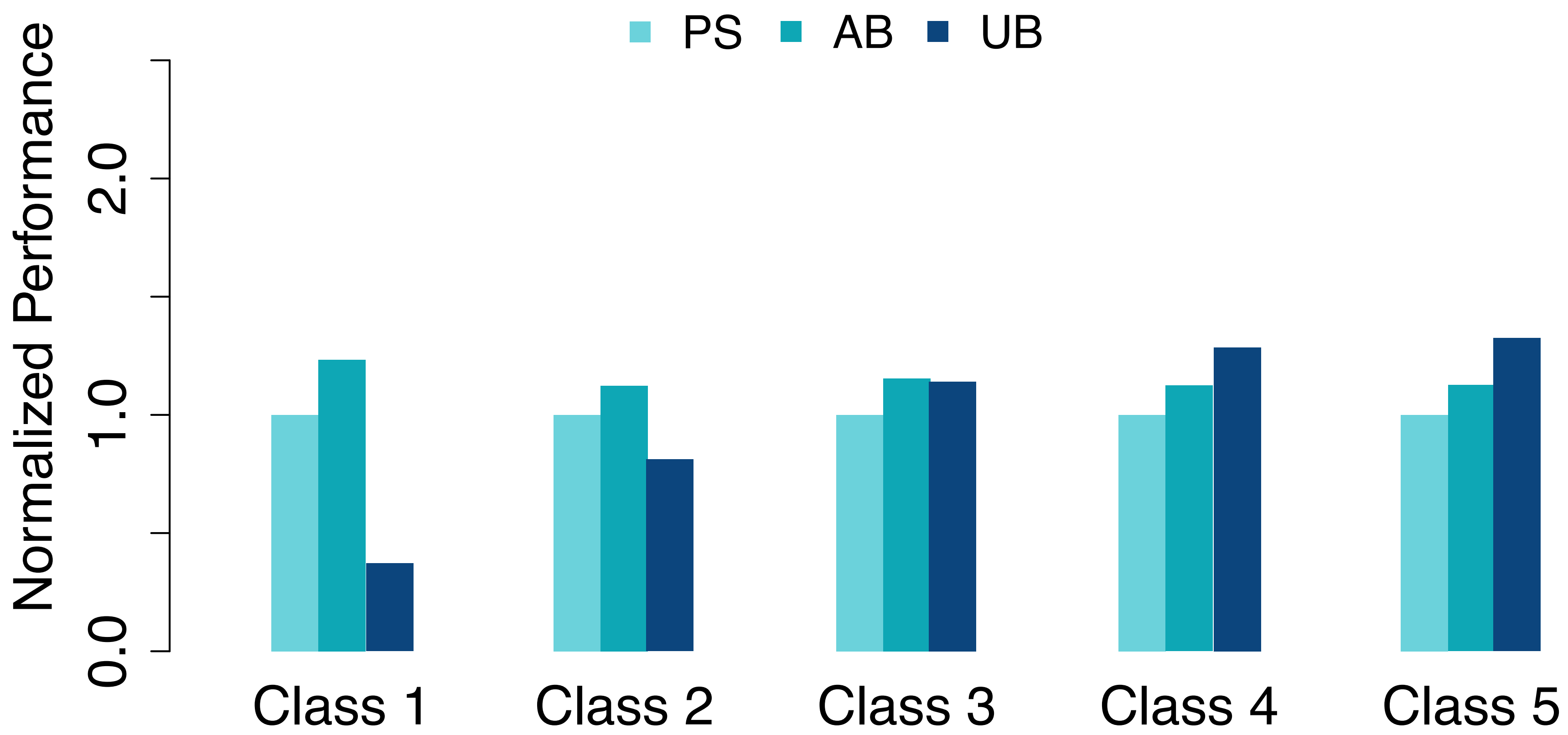
- **Proportional Sharing (PS)**

  Allocate cores in proportion to entitlements on each server

- **Upper-Bound (UB)**

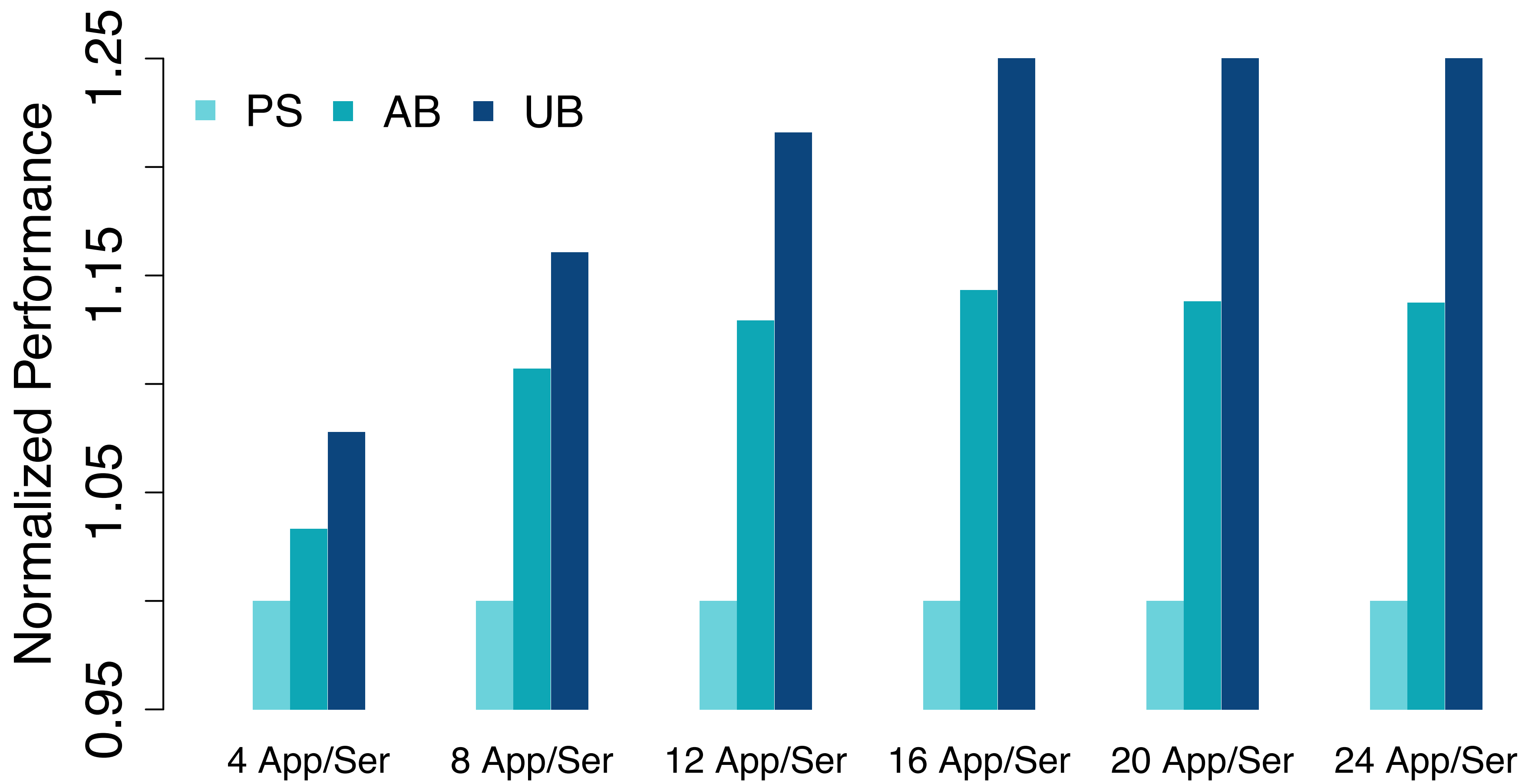
  Allocate cores to maximize system progress

# Sharing Incentives




- PS – provides SI by definition
- UB – treats users unfairly, starves users with low entitlements
- AB – provides SI with market equilibrium, low overhead

# System Performance




- PS – ignores demands
- UB – achieves highest performance
- AB – outperforms PS, low overhead

# Computational Overheads

- Updating bids: 0.1ms
- Termination check and setting prices: 0.85ms
- Convergence rate of 10 iterations, on average
- Overhead of 12.5ms with 0.25ms network delay

# Summary and Future Direction

- Amdahl utility measures progress using **Karp-Flatt metric**
- Amdahl bidding procedure finds **market equilibrium**
  - Allocations are **work-conserving**
  - Market guarantees **sharing incentives**
  - Users bid **truthfully** in large, competitive systems

# Thank You