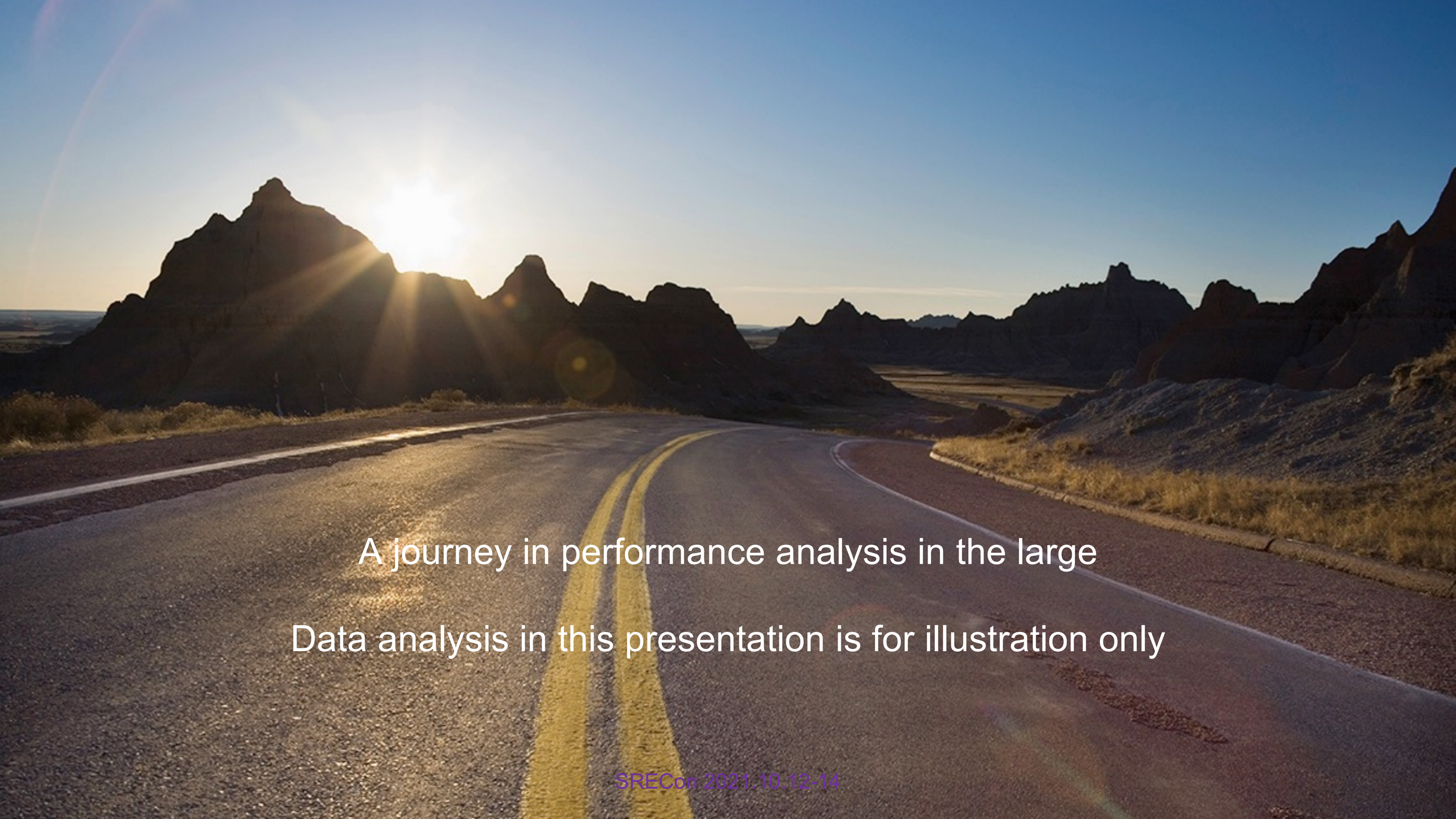
Need for SPEED
Site Performance Efficiency, Evaluation and Decision

Kingsum Chow, Alibaba Inc
Zhihao Chang, Zhejiang University

SRECon 2021.10.12-14


A journey in performance analysis in the large
Data analysis in this presentation is for illustration only
SRECon 2021.10.12-14

Outline

- Some performance metrics are misleading
  - e.g., CPU utilization in the presence of Hyper-Threading
- Performance Evaluation at Scale
- Summary and next steps

SPEED

Site Performance Efficiency, Evaluation and Decision

Estimation

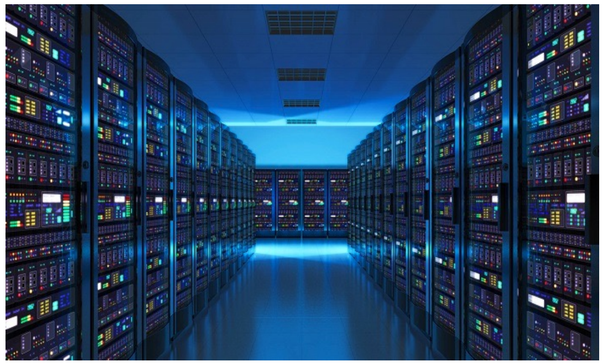
- What happens now?

Validation

- What are the real effects?

Evaluation

- What is the influence of the SW/HW upgrade?



Understanding a Performance Metric is Non-trivial

- Given: a data center that has a **CPU utilization** of 50%
- Assume: no software scaling and no interference problems
- Decide: reduce the number of servers by half

Hardware Mechanisms of Intel HT Technology

Intel HT Technology allows one physical processor core to present two *logical* cores to the operating system, which allows it to support two threads at once. The key hardware mechanism underlying this capability is an extra *architectural state* supported by the hardware, as shown in Figure 1.

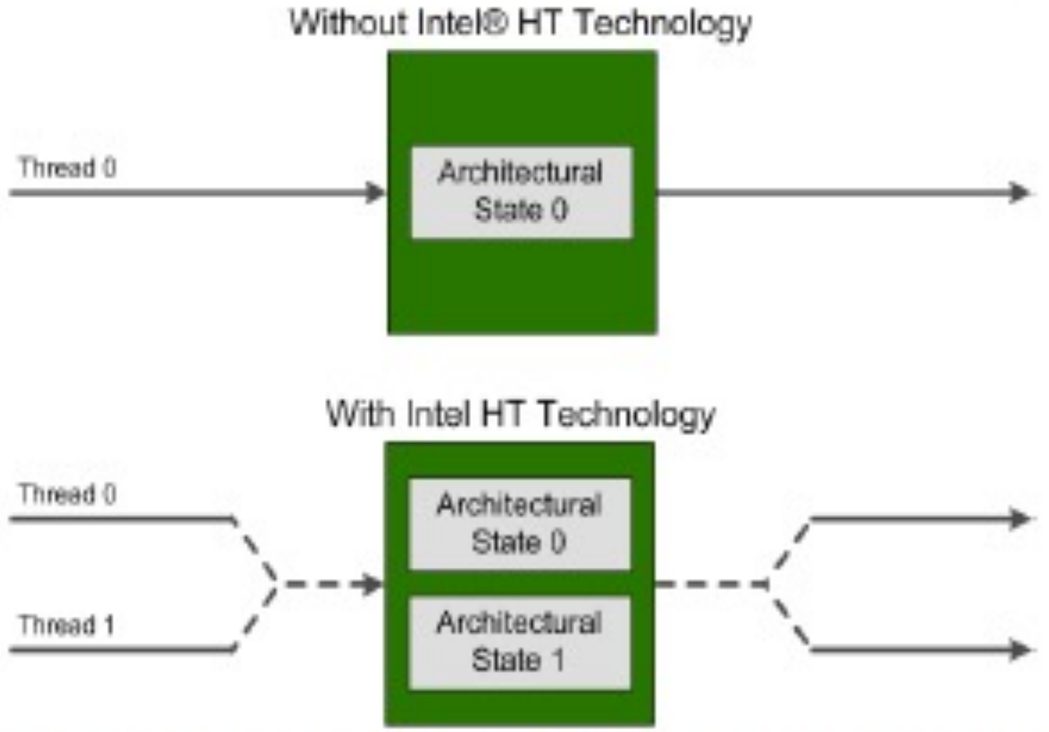


Figure 1. Intel® HT Technology enables a single processor core to maintain two architectural states, each of which can support its own thread. Many of the internal microarchitectural hardware resources are shared between the two threads.

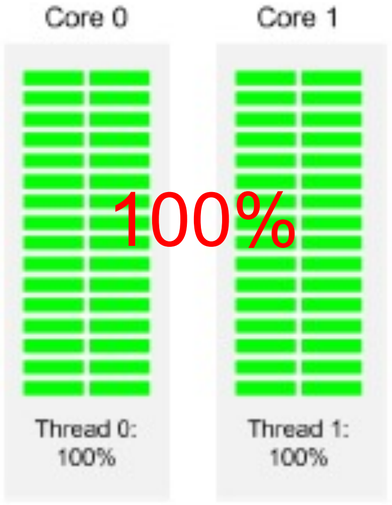


Figure 8. The system represented here has Intel® HT Technology disabled and two threads running, both at 100% utilization.

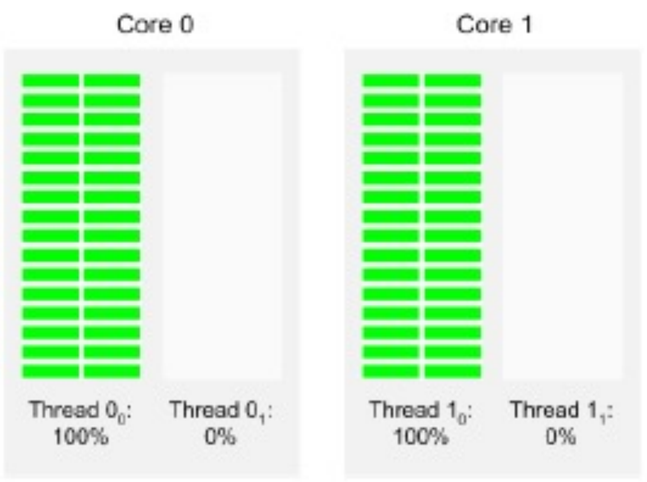


Figure 9. The system represented here has Intel® HT Technology enabled; two threads are running at 100% utilization and two threads are idle (at 0% utilization). Each core has one logical processor at 100% utilization and one logical processor at 0%.

50% != 50% ??

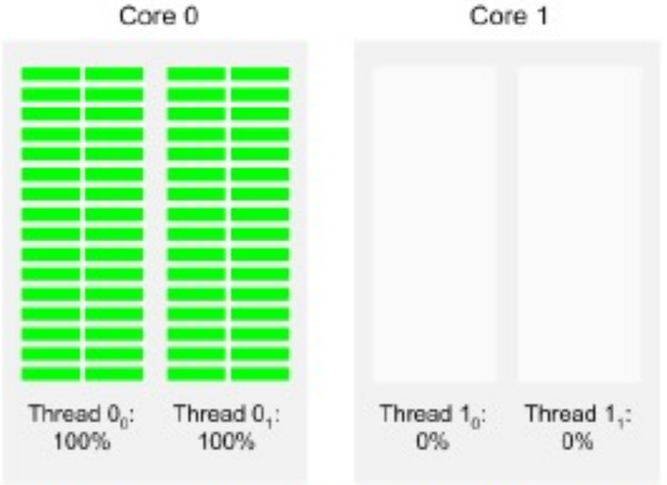
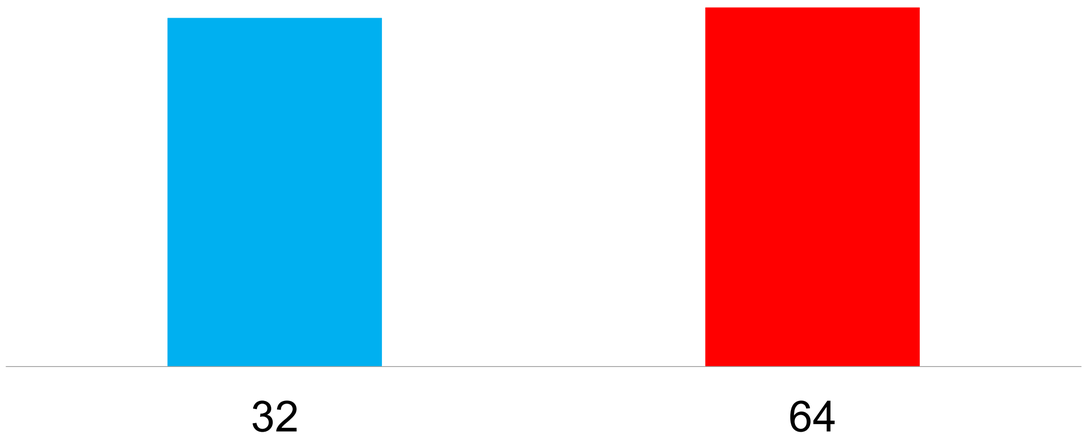


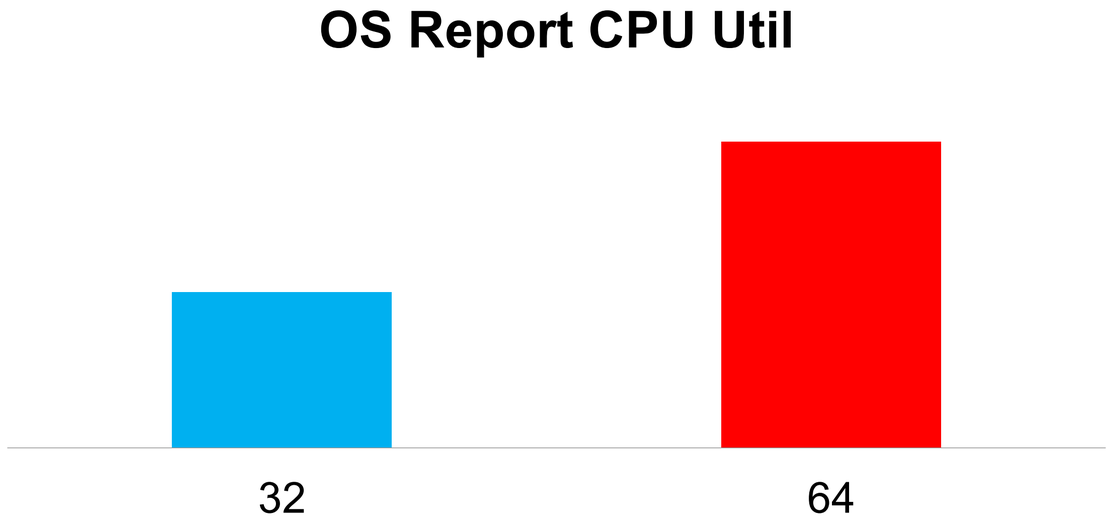
Figure 10. The system represented here has Intel® HT Technology enabled; again, two threads are running at 100% utilization and two threads are idle (at 0% utilization). Here, one core has two threads running at 100% and one core has two threads at 0%.

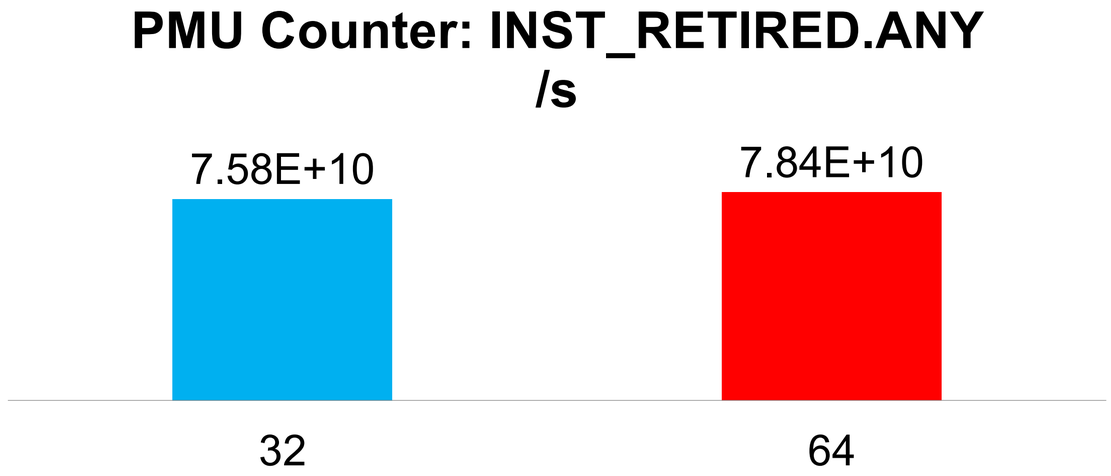
Experiment

- Intel Xeon CPU, 2 sockets, 16 cores per socket, Hyper-Threading turned on. Total 64 logical CPUs.
- SPECjbb2005 benchmark: each warehouse ran on one logical CPU.
- (1) 32 warehouses (half of all logical CPUs).
- (2) 64 warehouses (all logical CPUs).

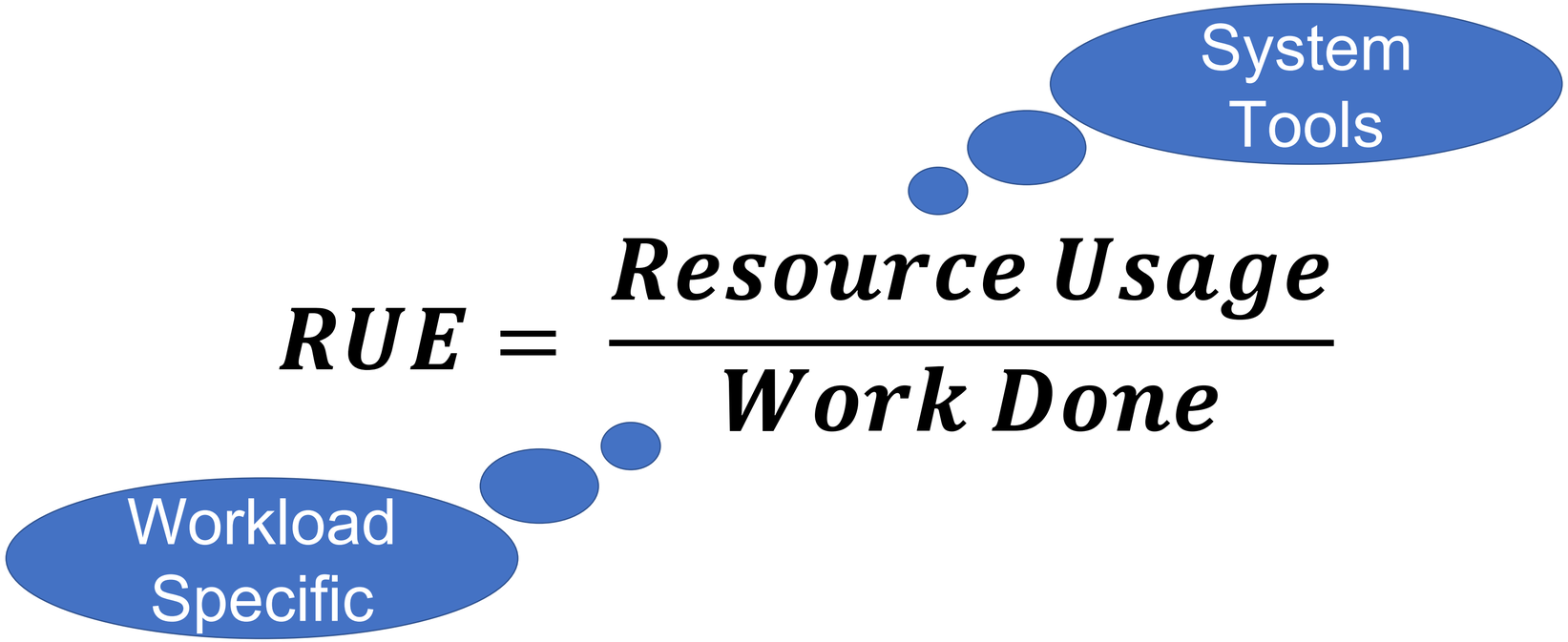
Experiment Results







Performance Estimation at Scale



Smaller is better

- Resource usage: CPU, Memory, Storage, Network
- Work Done: Queries, Tasks

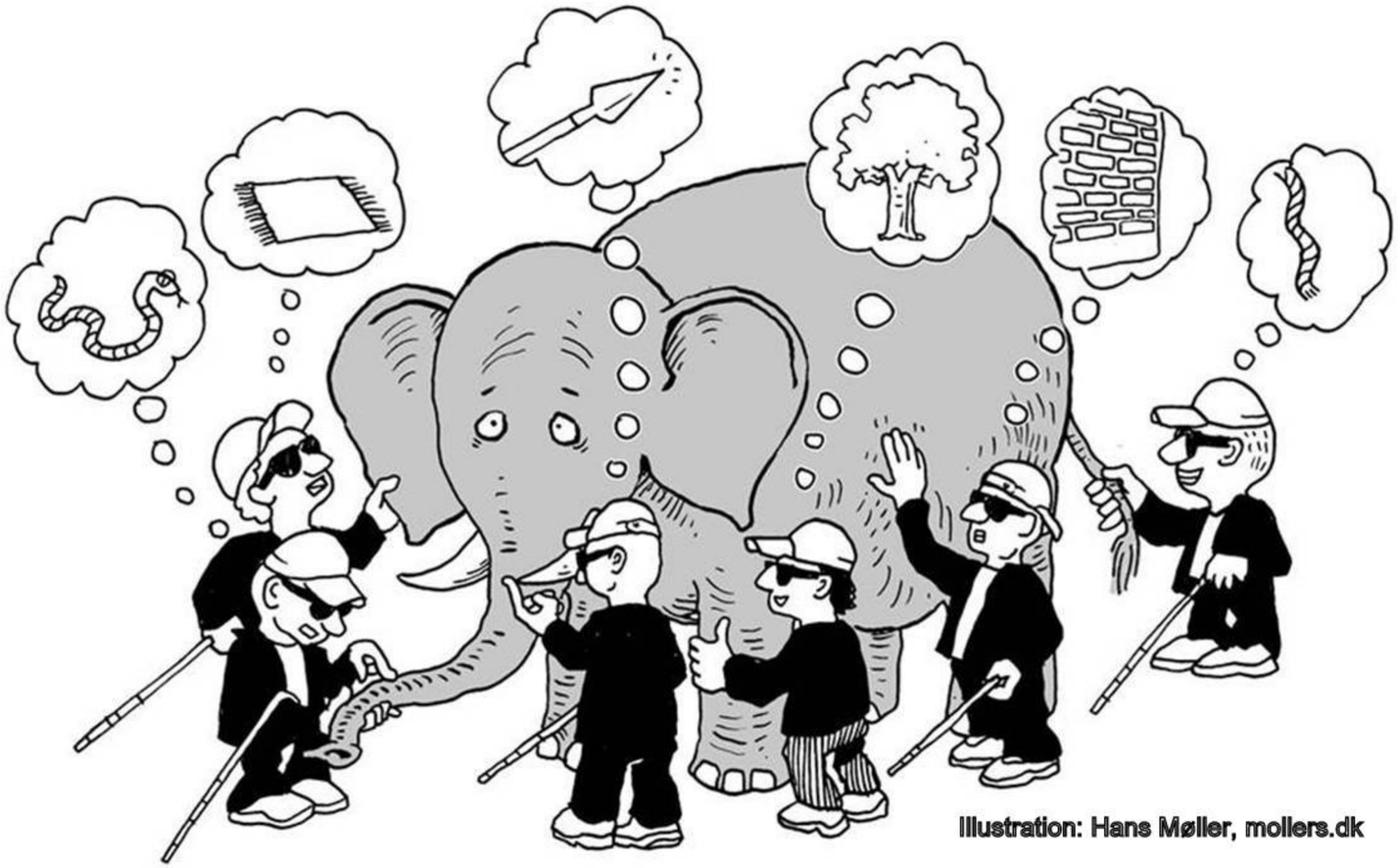
Performance Evaluation at Scale

$$\boldsymbol{Speedup} = \frac{\boldsymbol{RUE_1}}{\boldsymbol{RUE_2}}$$

Bigger is better

- RUE_1 is the RUE of configuration 1
- RUE_2 is the RUE of configuration 2

Performance Data Collection in the Large



The law of large numbers

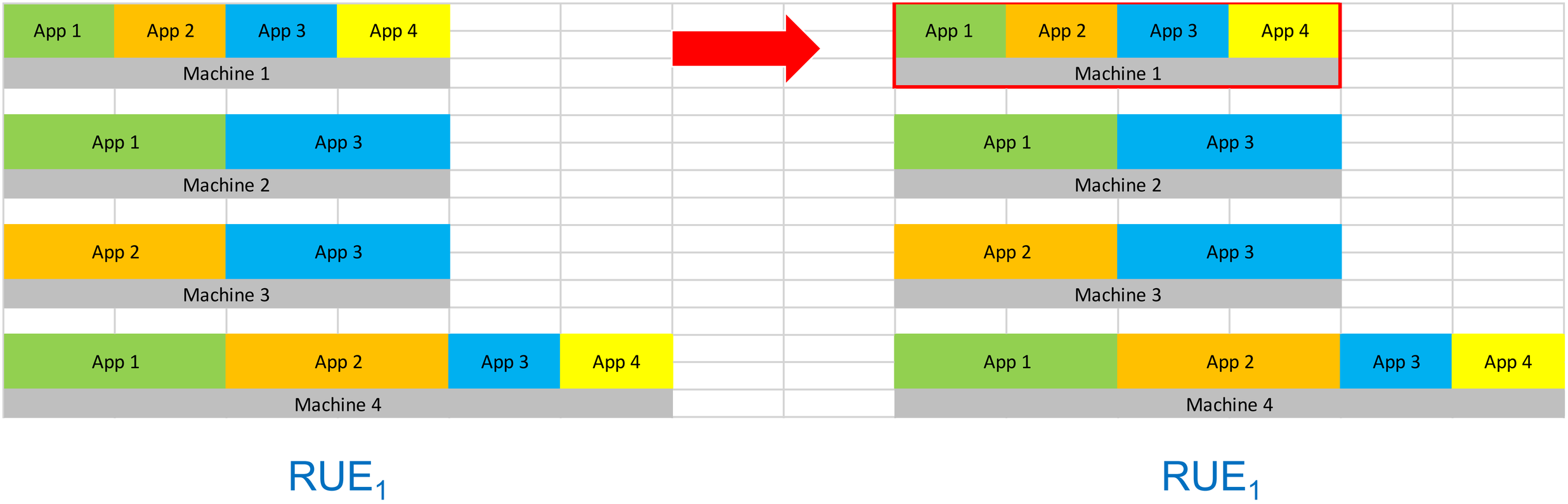
- a theorem that describes the result of performing the same experiment a large number of times. According to the law, the average of the results obtained from a large number of trials should be close to the expected value, and will tend to become closer as more trials are performed.

https://en.wikipedia.org/wiki/Law_of_large_numbers

Example: Testing a new feature

- To reduce the cost of testing
  - 1% of instances of an application ran on the new config (config 2), 99% of instances ran on the old config (config 1)
  - No change in deployments, each app might run on the new config or the old config
- We still have a large number of samples, even with 1% of the instances

Performance Evaluation at Scale



Big Data

	Config 1		Config 2		Speedup
	Proportion of App Instances	RUE_1	Proportion of App Instances	RUE_2	
App Total	99.00%	885	1.00%	815	1.09

Looks really promising, let's change ??
More samples needed?
More analysis needed?

Big Data Paradox

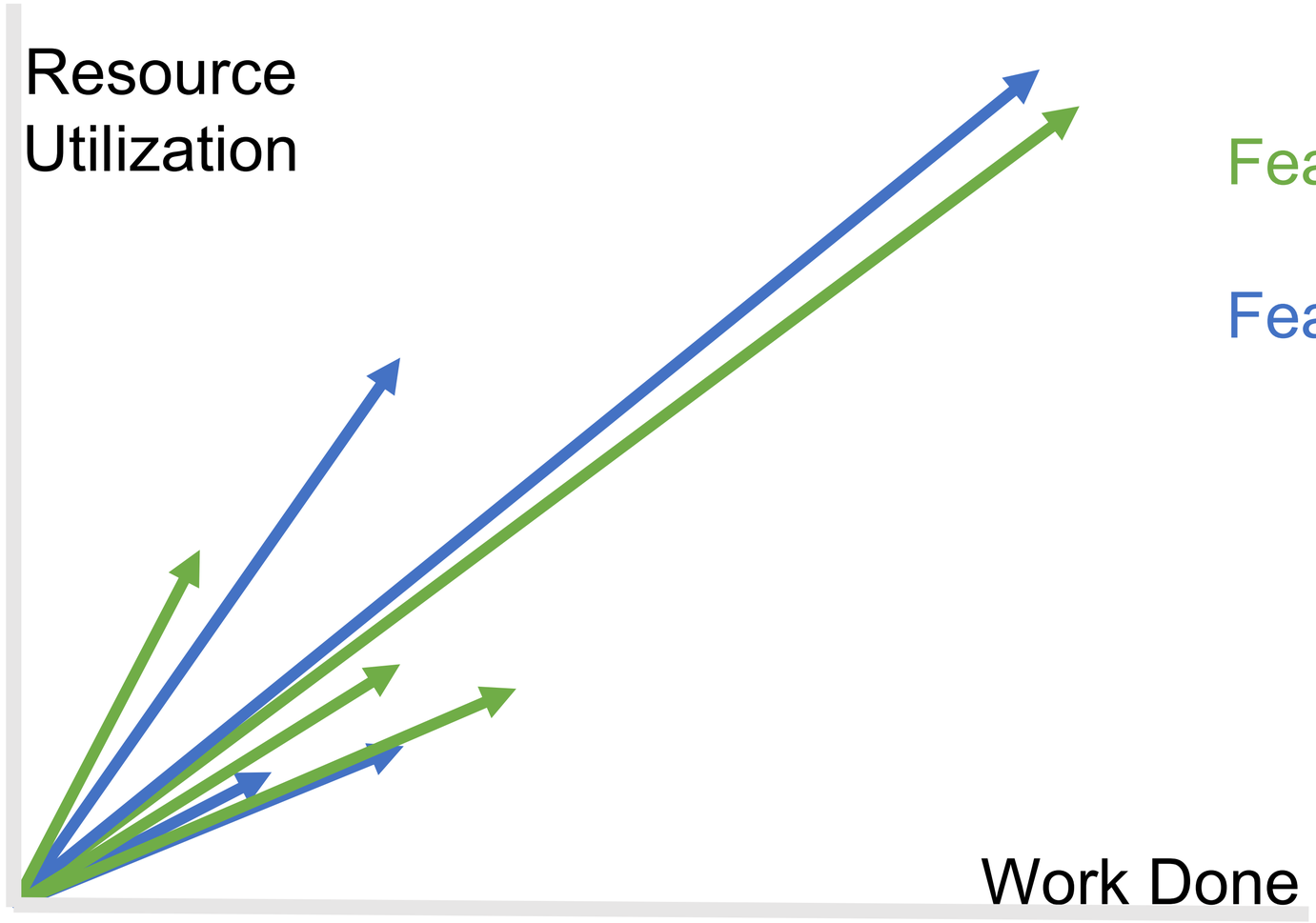
	Config 1		Config 2		Speedup
	Proportion of App Instances	RUE_1	Proportion of App Instances	RUE_2	
App Total	99.00%	885	1.00%	815	1.09
App Group 1	50.10%	1289	0.30%	1484	0.87
App Group 2	31.50%	428	0.40%	434	0.99
App Group 3	17.40%	550	0.30%	655	0.84

Simpson’s Paradox

- A trend appears in several different groups of data but disappears or reverses when these groups are combined

https://plato.stanford.edu/entries/paradox-simpson/

Simpson’s Paradox



Feature on

Feature off

Overall performance increases

Is that real?

Ready to try to detect Simpson's Paradox in your SRE performance analysis?

- https://github.com/ninoch/Trend-Simpsons-Paradox
- https://github.com/CamDavidsonPilon/simpsons-paradox
- https://github.com/ijmbarr/simpsons-paradox
- https://github.com/ehart-altair/SimpsonsParadox

Watch out

- ***CPU*** utilization may be more complicated than it looks
- ***Average*** performance change from a lot of data could be misleading

Be wary of any ratio used in performance analysis

You can find me on LinkedIn
https://www.linkedin.com/in/kingsumchow/