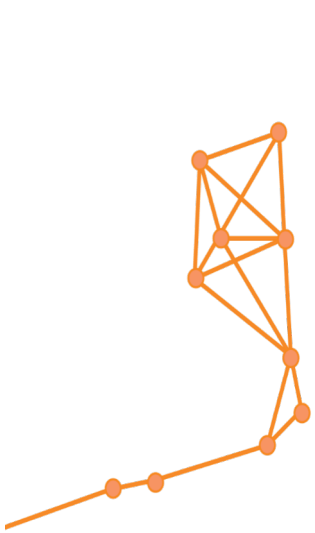


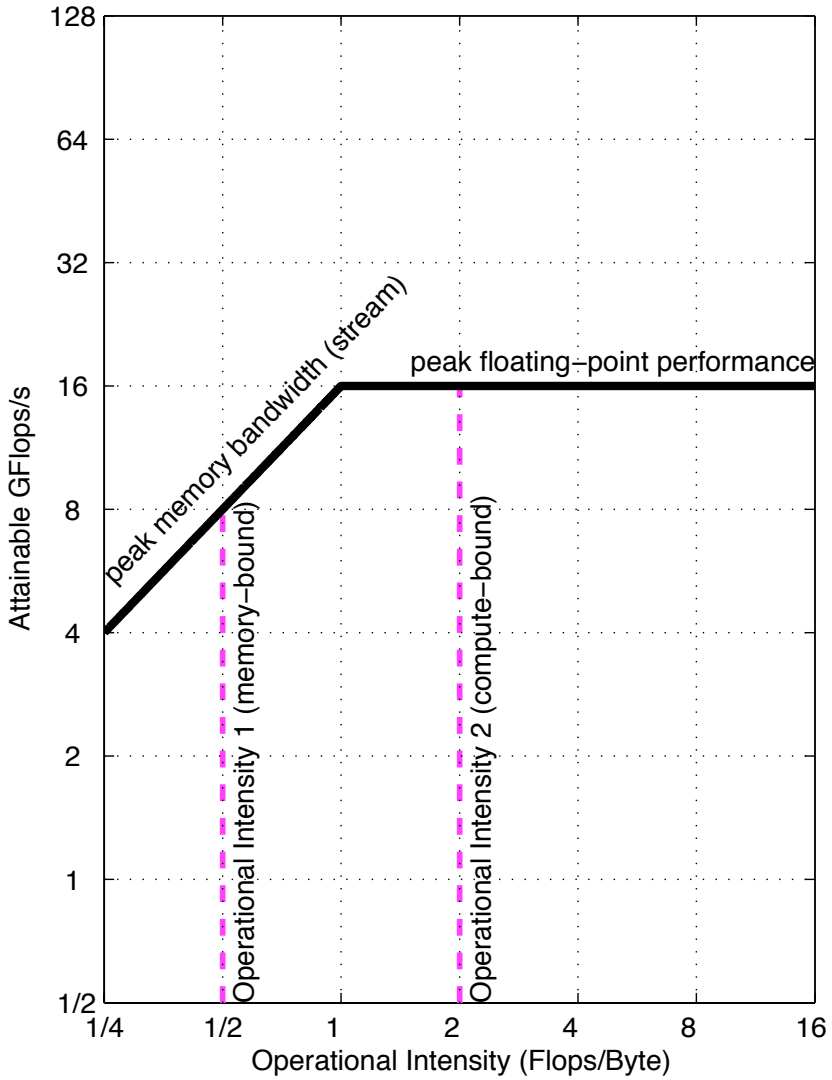

Modelling Load Imbalance In Shared Memory Multicore Systems

Johannes Langguth
joint work with Xing Cai and James Trotter

ODISSEE

simula

Performance Modelling for Multicore: the Roofline Model

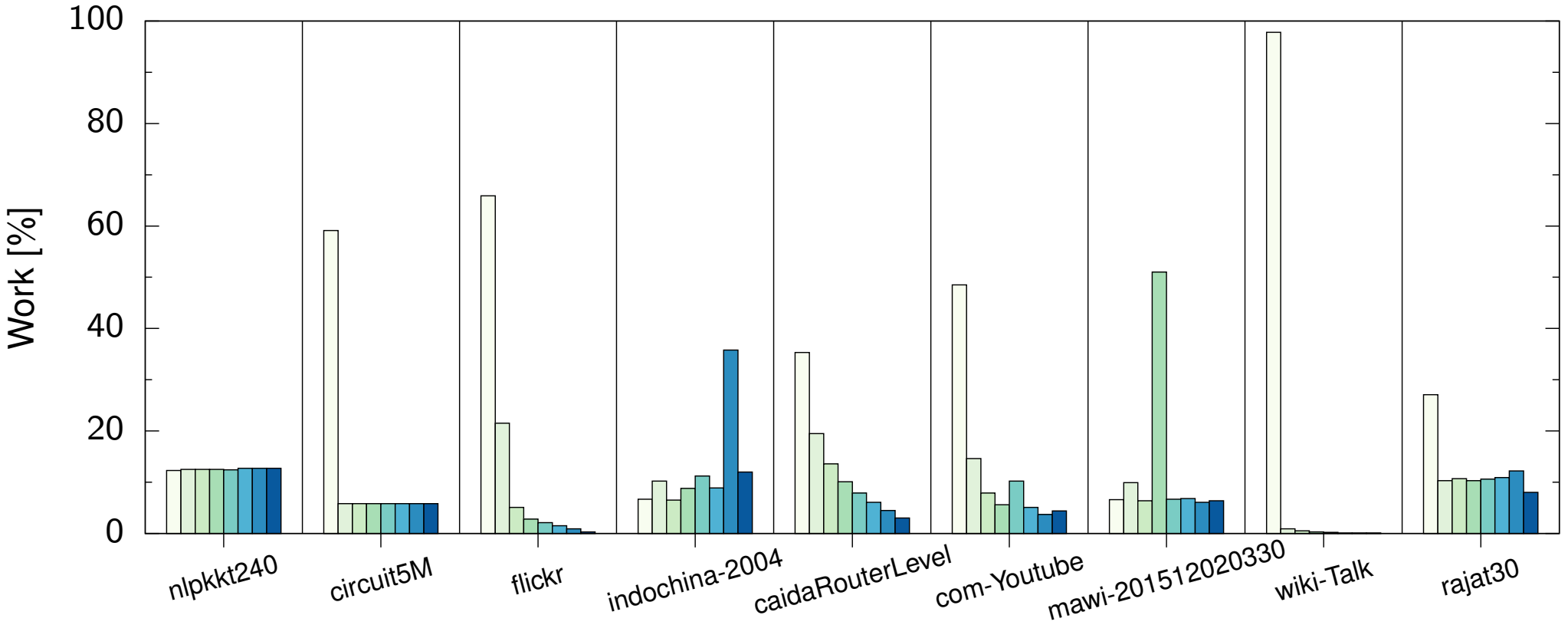


- Widely accepted standard
- Not very accurate or detailed, but very *insightful*
- Performance prediction, but can easily be converted to time
- No special consideration of parallelism
- Works for parallel computation as long as load is balanced

Williams et al., *Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures, 2008*

simula

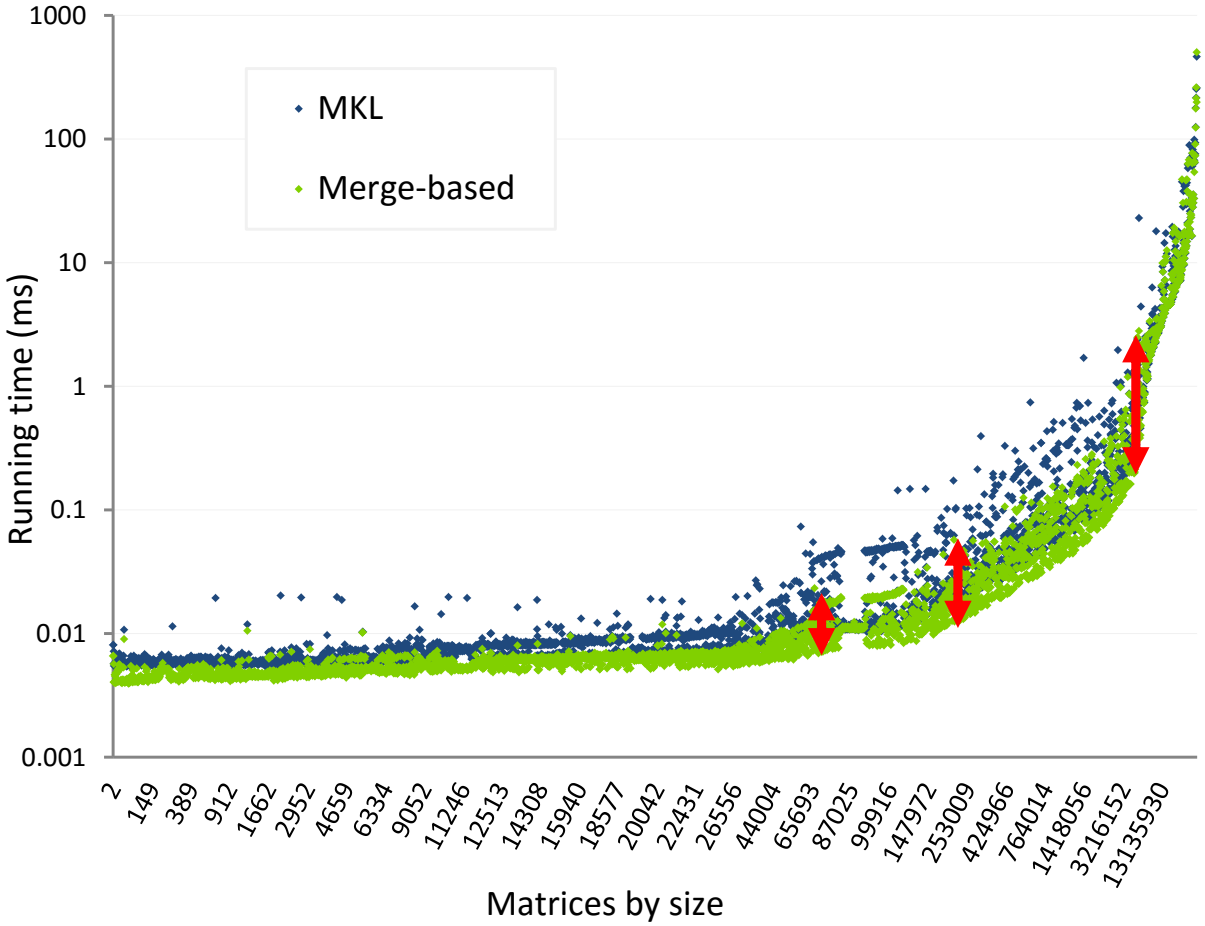
Common Problem: Load Imbalance



Number of nonzeros in 8-way even row split

simula

Hard to avoid Load Imbalance

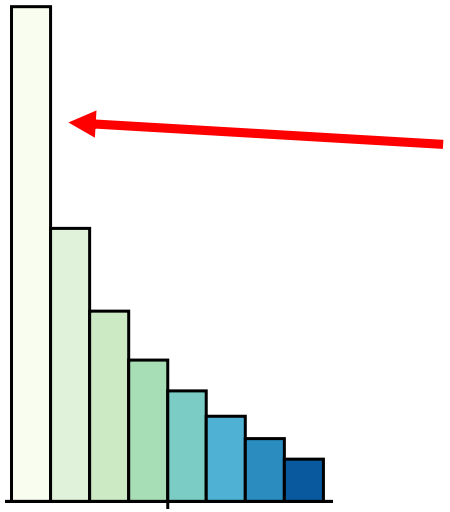


- Even if memory access volume is balanced, actual work performed can vary
- Example here: caching behaviour in SpMV
- Many other examples

Duane Merrill and Michael Garland: *Merge-based parallel sparse matrix-vector multiplication*, 2016

simula

The Roofline Model under Load Imbalance

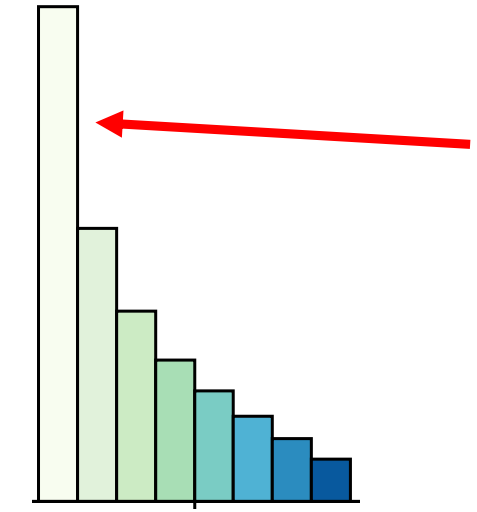


caidaRouterLevel

Obvious Idea:
Take the longest task and apply the model to a single core

simula

How can we apply the Roofline Model here?

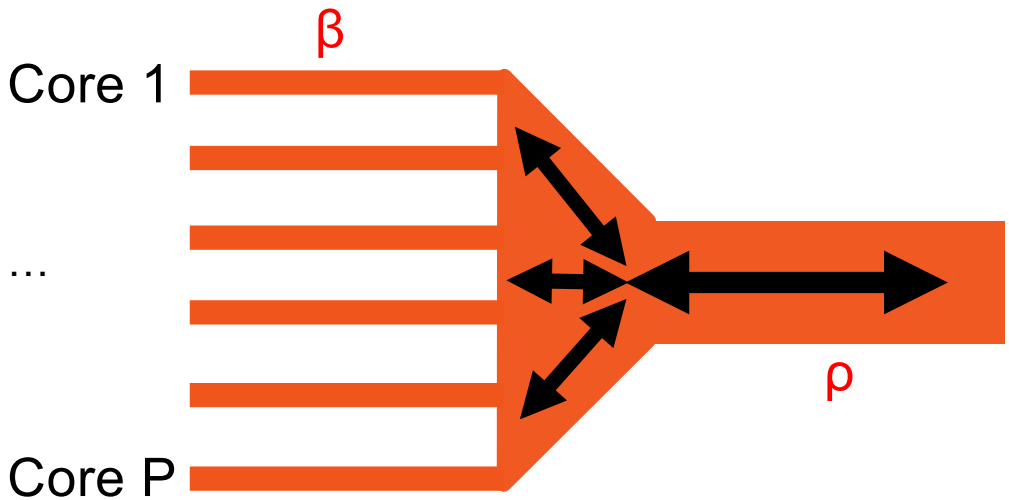


caidaRouterLevel

Obvious Idea:
Take the longest task and apply the model to a single core

- If compute bound, we are done (thus, no need to discuss arithmetic intensity here)
- If memory bound, what is the bandwidth for the longest task?

A simple model of Multicore Memory Bandwidth

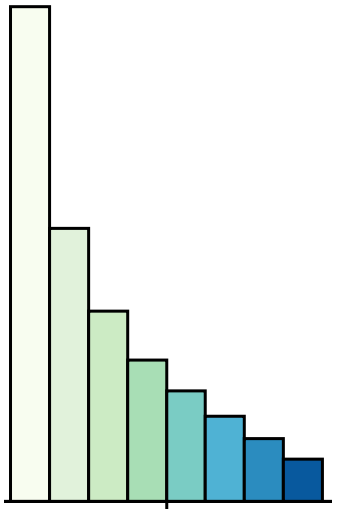


- Effectively a Max-Rate model
- Works for multicore and NIC
- Assume $\beta < \rho < P\beta$

Gropp et al.: *Modeling MPI communication performance on SMP nodes: Is it time to retire the ping pong test*, 2016

simula

How can we apply the Roofline Model here?



caidaRouterLevel

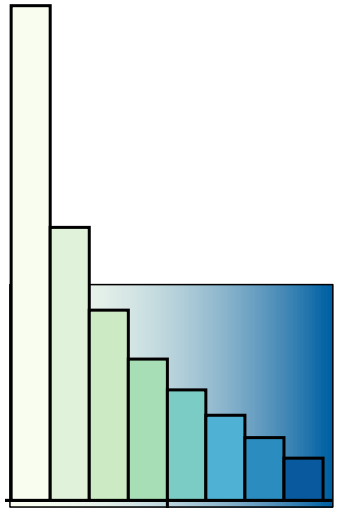
Obvious Idea:
Take the longest task and apply the model to a single core

- If compute bound, we are done
- If memory bound, what is the single core bandwidth?

A. Full Contention: ρ/P (all core STREAM by core count P)
B. No Contention: β (single core STREAM)

simula

How can we apply the Roofline Model here?



caidaRouterLevel

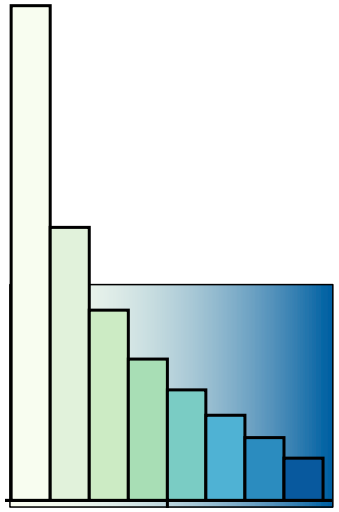
Another Idea (for the memory bound case):
Take the sum of all tasks and apply the model to all cores

For convenience:

- Let M_i be the workload of Processor i
- Sort processors in descending order of workload

simula

How can we apply the Roofline Model here?



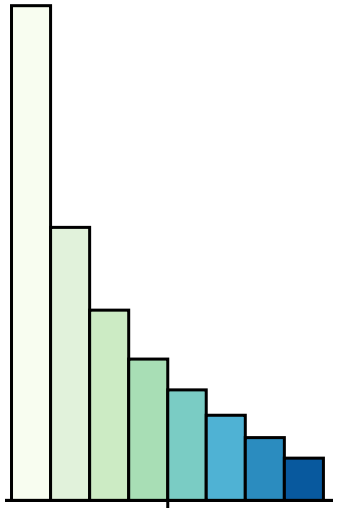
caidaRouterLevel

Another Idea (for the memory bound case):
Take the sum of all tasks and apply the model to all cores

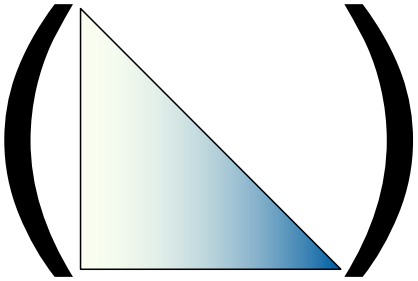
A. *Full Contention:* $M_1 / (\rho/P)$
B. *No Contention:* M_1 / β
C. *No Imbalance:* $\sum_i M_i / \rho$

Let's test these Models

We need imbalanced workloads



caidaRouterLevel



triangular matrix

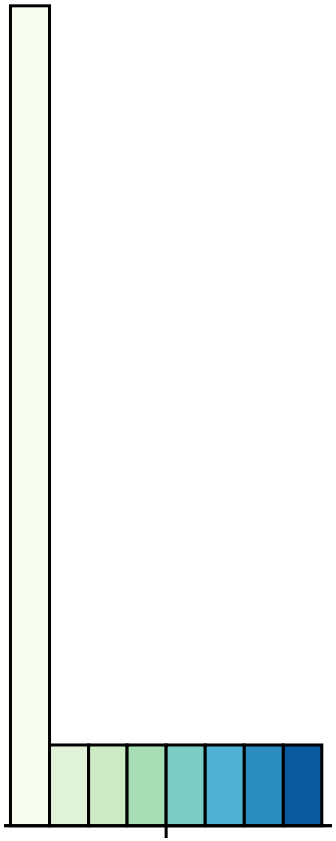
Triangular workload
Inspired by dense triangular matrix

Test code:
Triangular matrix-vector multiply

```
#pragma omp parallel for
for (int i = 0; i < N; i++) {
        double z = 0.0;
        for (int j = 0; j < i; j++)
                z += A[i*N+j]*x[j];
        y[i] += z;
}
```


simula

A second imbalanced Workload



circuit5M

Simple imbalanced workload
related to Amdahl's Law

$M_1 = (P+1)S$, $M_{i>1} = S$, for some value of S
(same work with and without contention)

Test code:
Imbalanced STREAM benchmark "imbaSTREAM"

Triangular Results on various CPUs

	Measured	*Full Contention*		*No Contention*		*No Imbalance*		Phase 1
CPU	[GB/s]	[GB/s]	Error	[GB/s]	Error	[GB/s]	Error	Time
AMD Epyc 7302P (Rome)	60.24	46.92	-22%	188.53	213%	90.91	51%	91.30%
AMD Epyc 7413 (Milan)	88.21	52.38	-41%	390.03	342%	102.58	16%	96.70%
AMD Epyc7601 (Naples)	66.9	43.39	-35%	295.02	341%	85.42	28%	96.40%
AMD Epyc7763 (Milan)	127.07	61.09	-52%	997.44	685%	121.23	-5%	99.40%
Intel Xeon Gold 6130	44.1	38.58	-13%	110.82	151%	74.74	69%	81.50%
Intel Xeon Platinum 8168	63.98	35.21	-45%	144.68	126%	68.96	8%	90.60%
Intel Xeon Platinum 8360Y	89.67	80.22	-11%	272.01	203%	158.21	76%	85.70%
Cavium Thunder X2	74.09	60.21	-19%	252.12	240%	118.54	60%	90.40%
HiSilicon Kunpeng 920	92.46	66.29	-28%	398.32	331%	131.54	42%	95.00%
NVIDIA Grace GH200	273.57	159.34	-42%	984.78	260%	316.45	16%	95.30%

Results use the bandwidth versions of the models

Mostly OK

Terrible

Mostly OK

simula

Amdahl Results on various CPUs

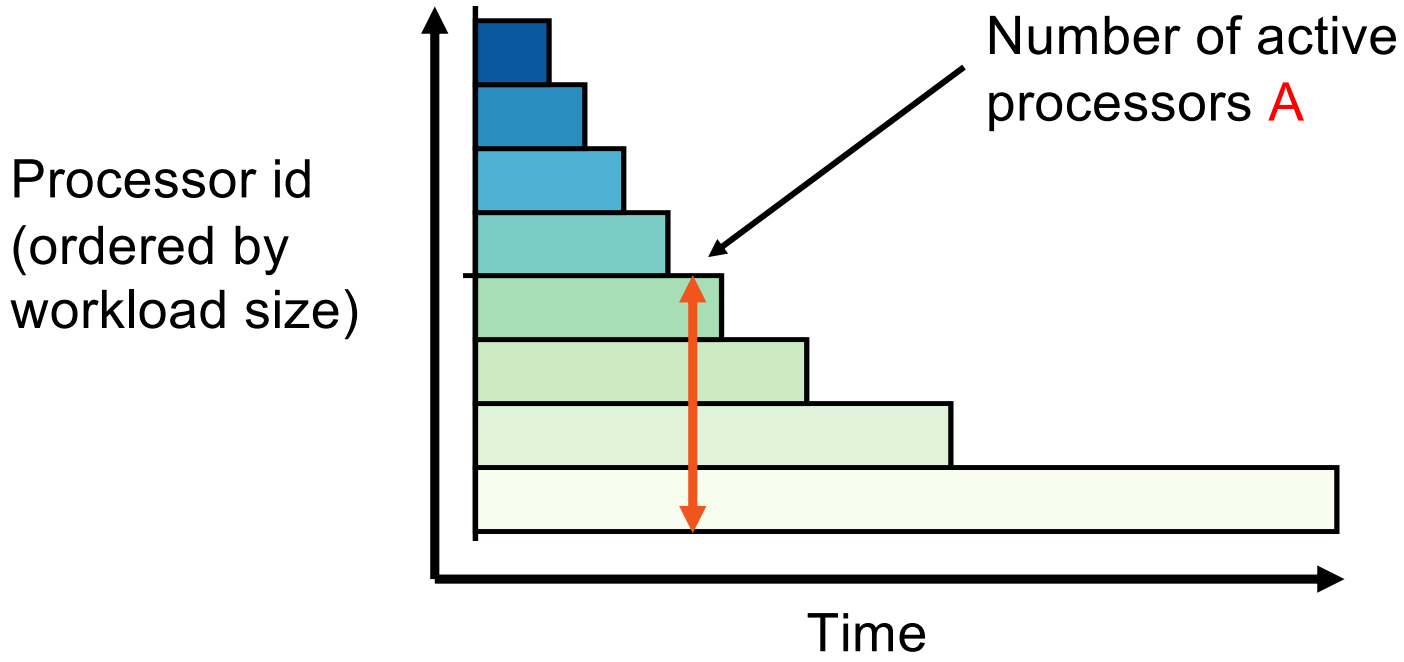
	Measured	*Full Contention*		*No Contention*		*No Imbalance*	
CPU	[GB/s]	[GB/s]	Error	[GB/s]	Error	[GB/s]	Error
AMD Epyc 7302P (Rome)	33.07	10.69	-67.66%	42.97	+29.94%	90.91	+174.89%
AMD Epyc 7413 (Milan)	43.52	8.21	-81.15%	61.10	+40.40%	102.58	+135.69%
AMD Epyc7601 (Naples)	22.46	5.18	-76.95%	35.20	+56.72%	85.42	+280.30%
AMD Epyc7763 (Milan)	47.34	3.73	-92.12%	60.90	+28.65%	121.23	+156.09%
Intel Xeon Gold 6130	22.07	8.79	-60.16%	25.26	+14.46%	74.74	+238.64%
Intel Xeon Platinum 8168	20.75	5.52	-73.41%	22.67	+9.25%	68.96	+232.36%
Intel Xeon Platinum 8360Y	28.14	8.55	-69.61%	29.00	+3.03%	158.21	+462.14%
Cavium Thunder X2	26.90	7.18	-73.29%	30.08	+11.84%	118.54	+340.71%
HiSilicon Kunpeng 920	22.21	4.05	-81.78%	24.32	+9.50%	131.54	+492.29%
NVIDIA Grace GH200	51.84	8.67	-83.28%	53.58	+3.36%	316.45	+510.42%

Terrible Quite good Terrible

Results use the bandwidth versions of the models

simula

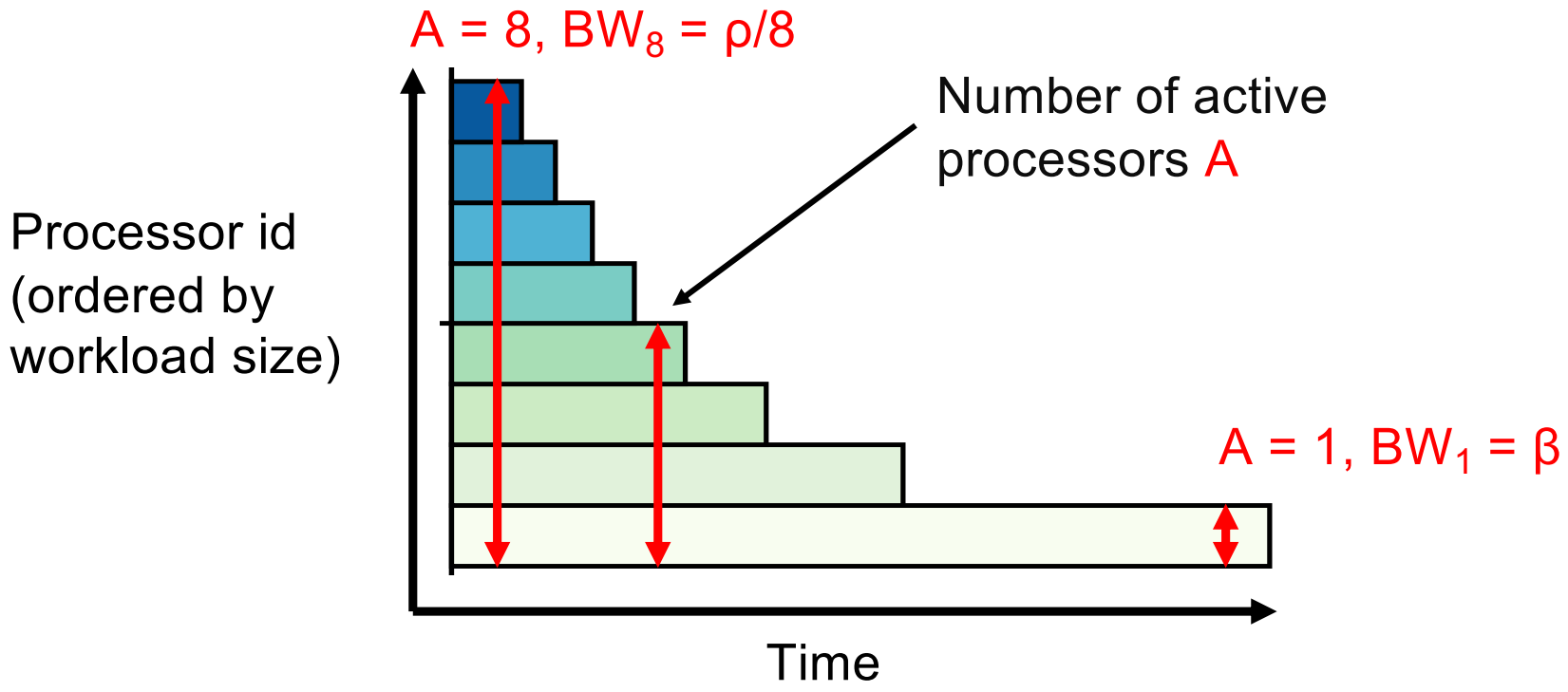
Why are the Models so Inaccurate?



Processor activity over time

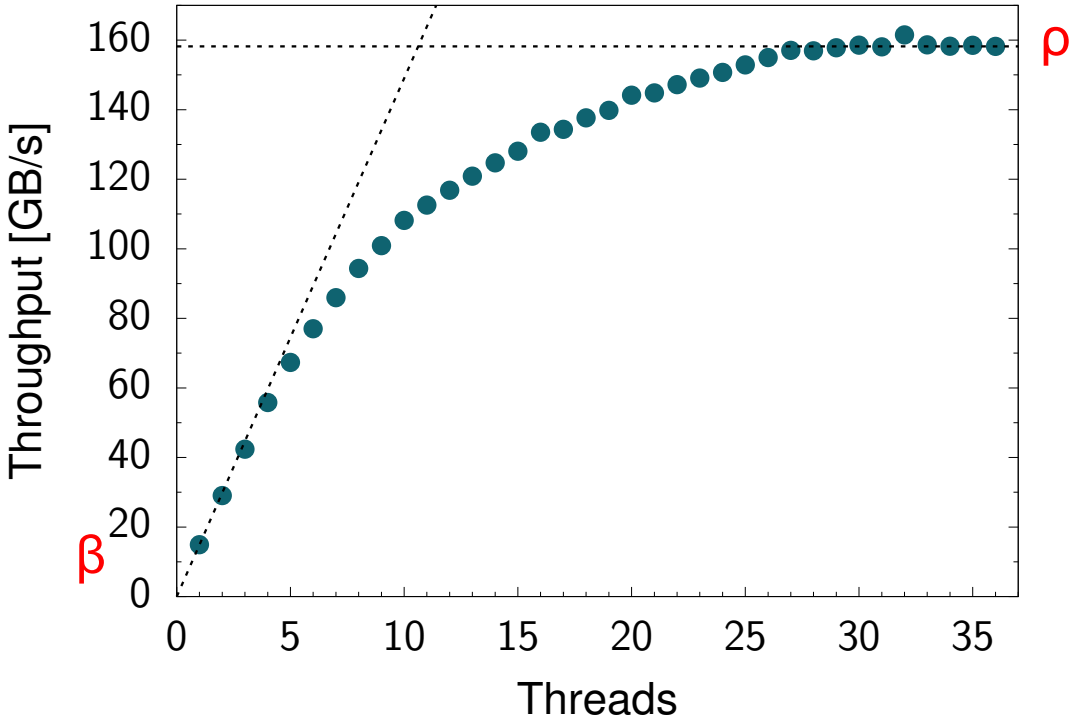
simula

Performance changes during the computation

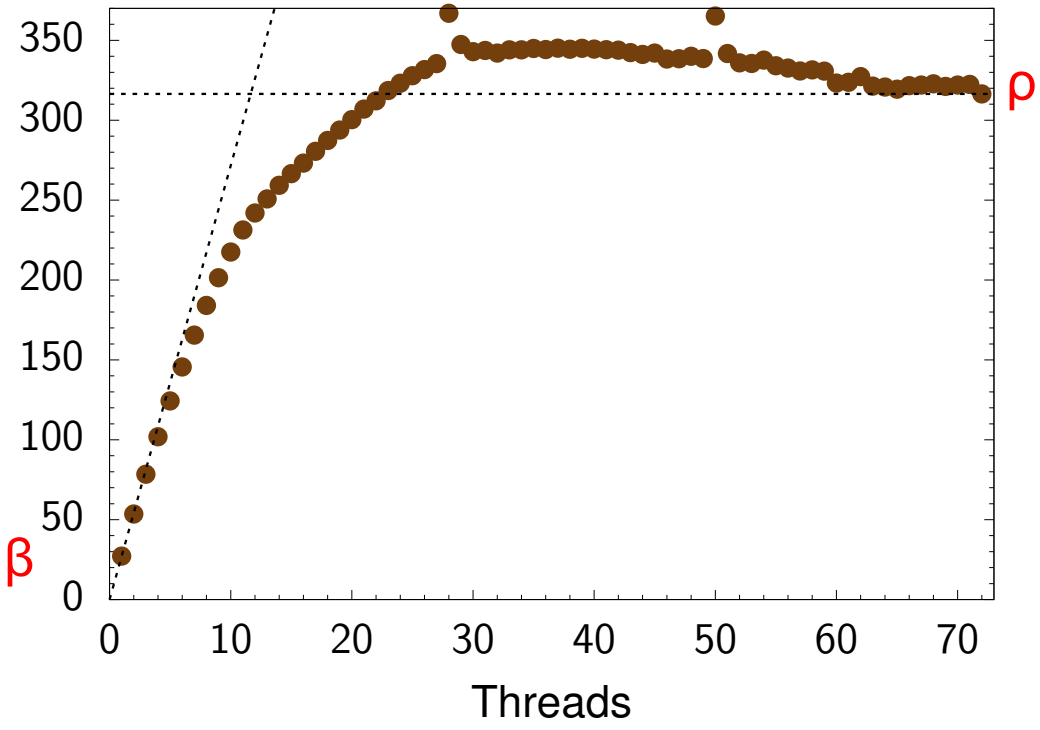


simula

Accurate Modelling: need STREAM by #active cores



Intel Xeon Platinum 8360Y (Ice Lake)

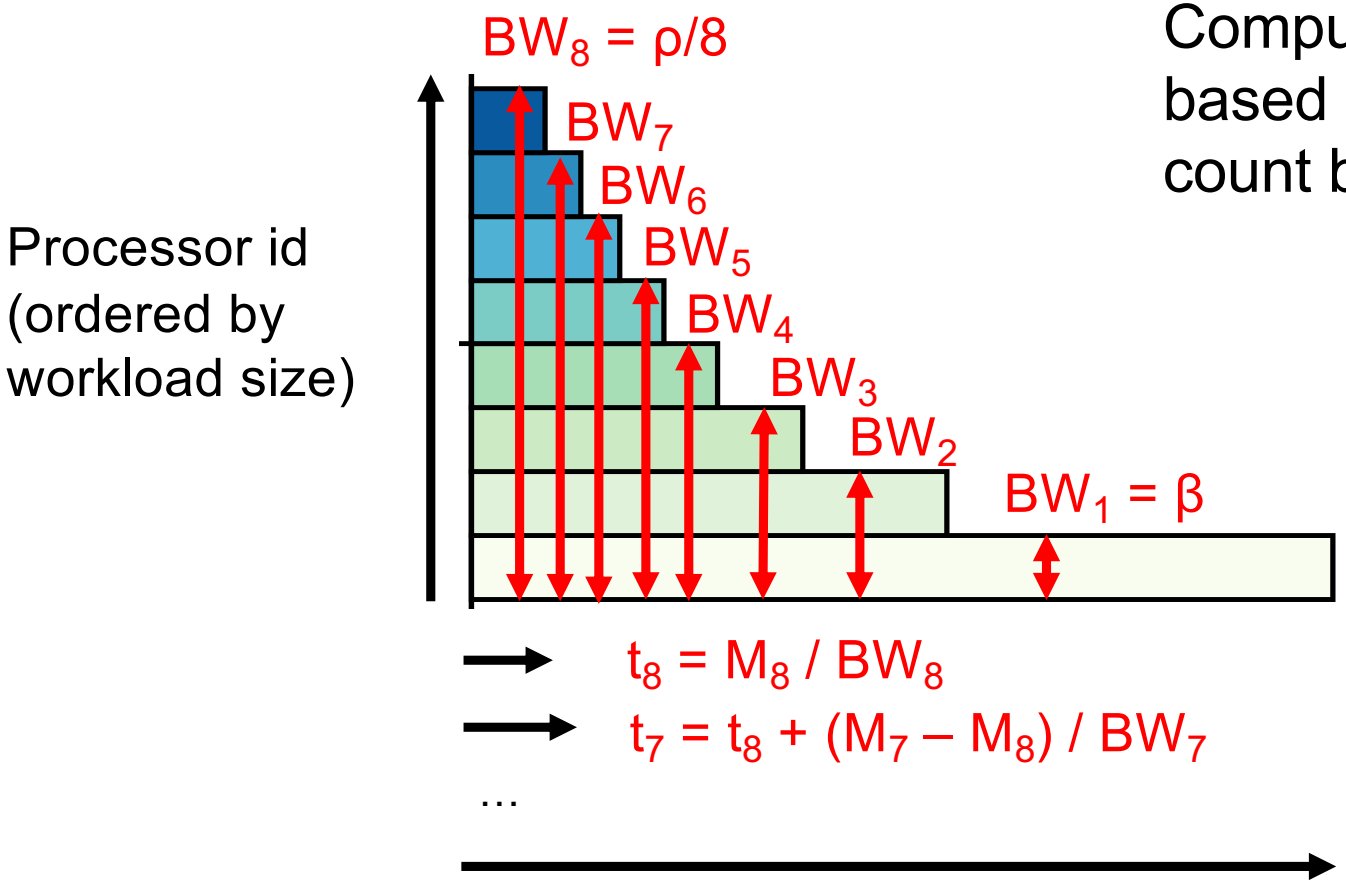


NVIDIA Grace GH200 CPU

Set of measurements BW_i = bandwidth with i active cores

simula

The Staircase Model

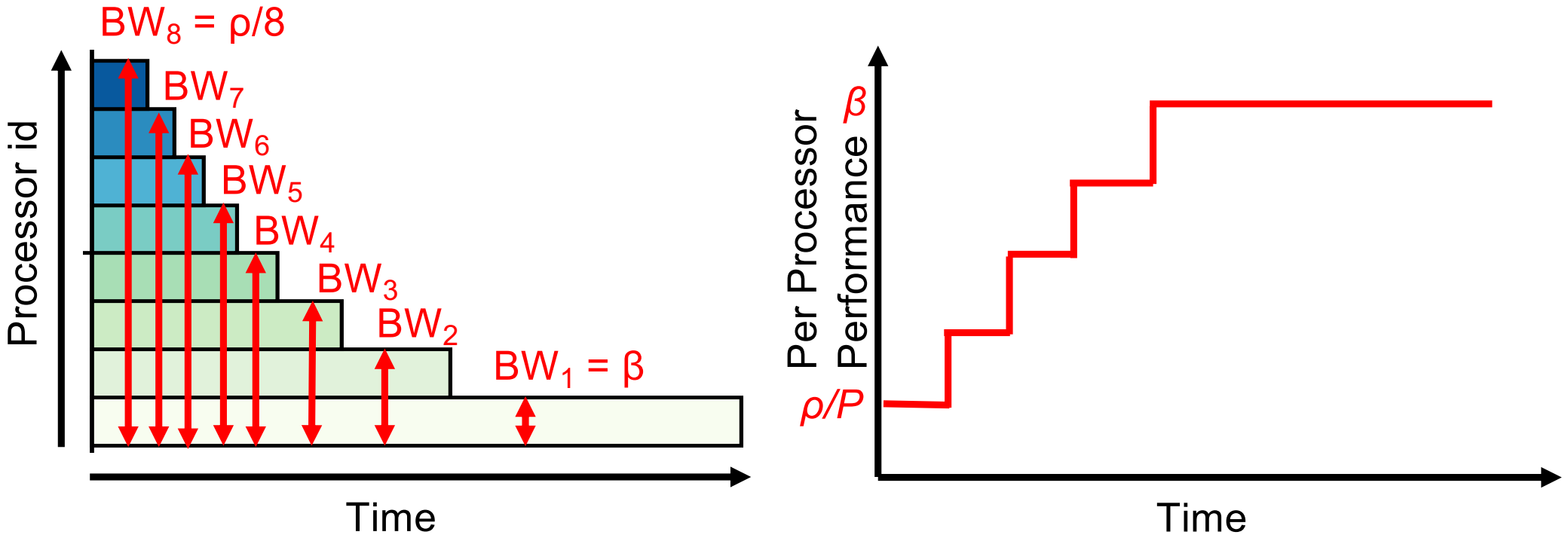


Computes time prediction based on individual processor count bandwidth values BW_i

Thune et al., Detailed Modeling of Heterogeneous and Contention-Constrained Point-to-Point MPI Communication, 2023

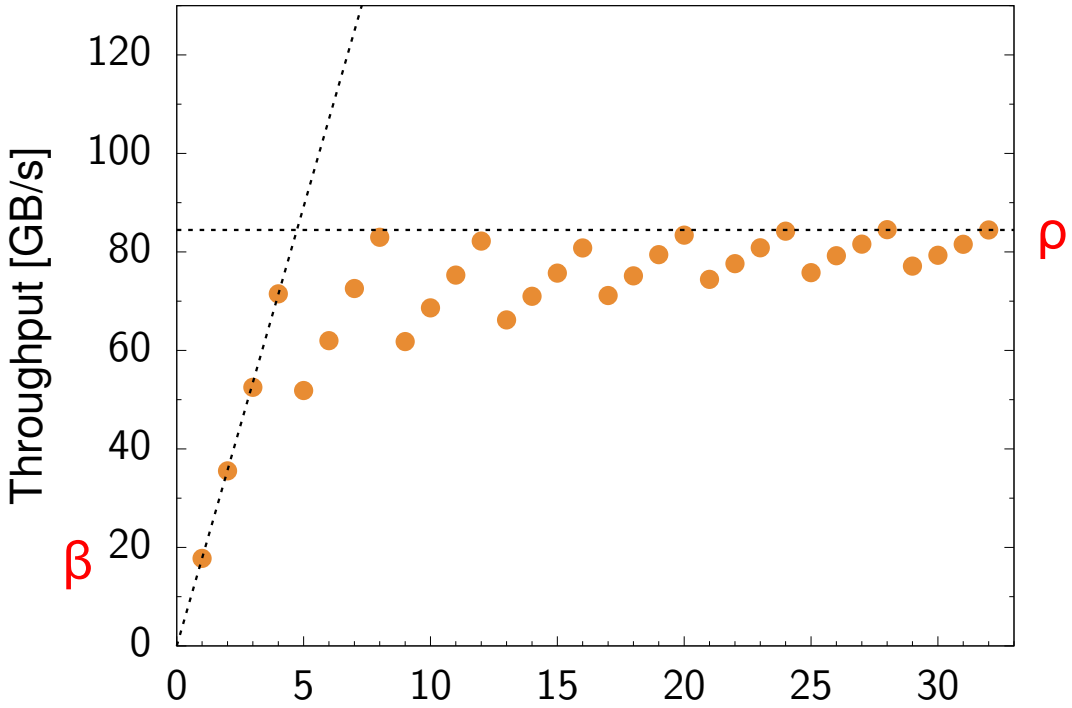
simula

Performance per Core over Time

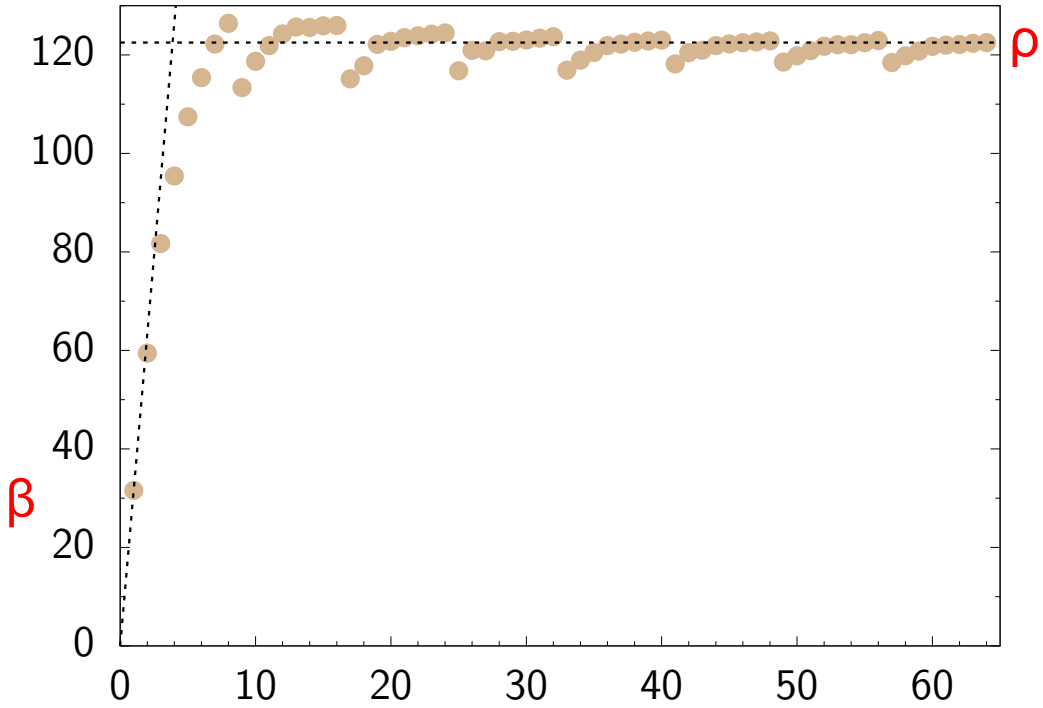


simula

Problems with the Staircase Model



AMD Epyc 7601 Naples

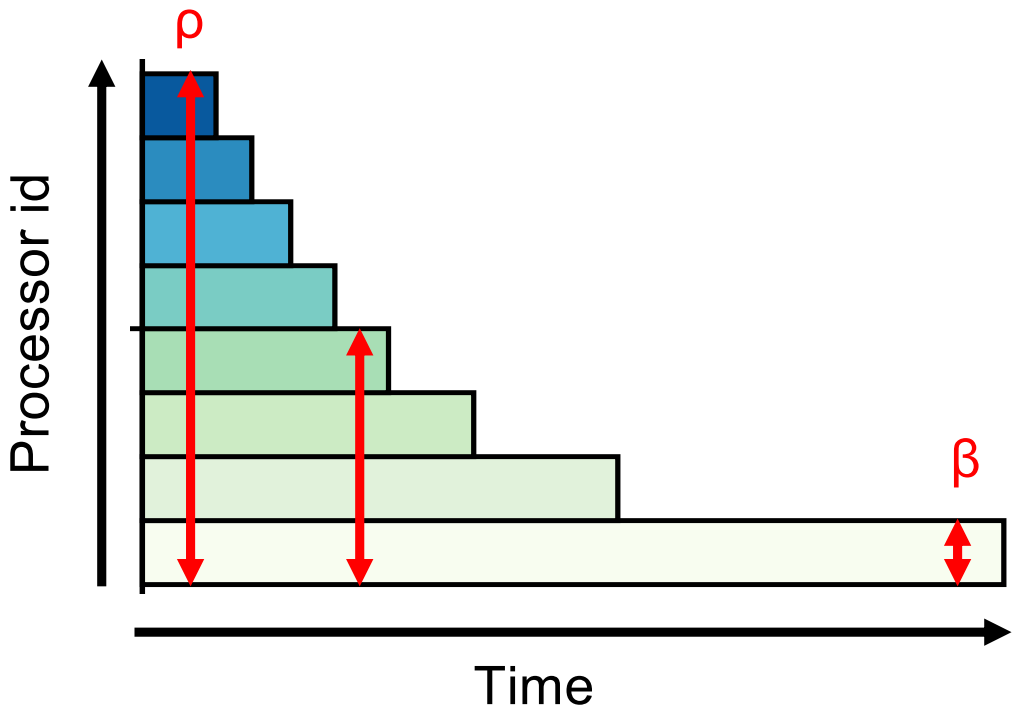


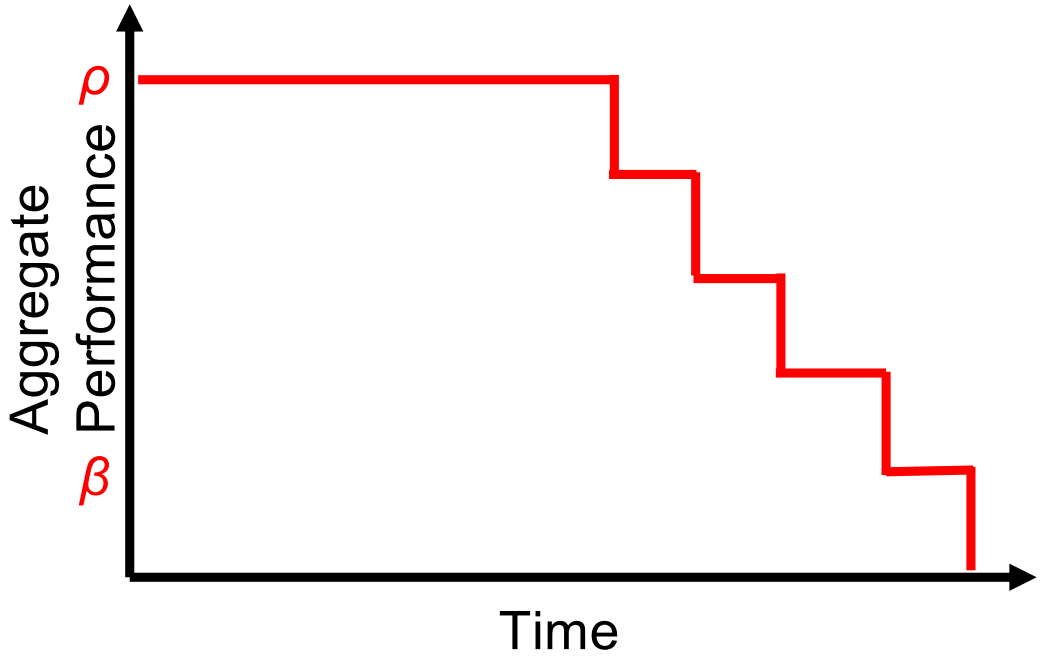
AMD Epyc 7763 Milan

- BW_i is not a unique value, can depend on exact cores
- Model is very complex, not insightful

simula

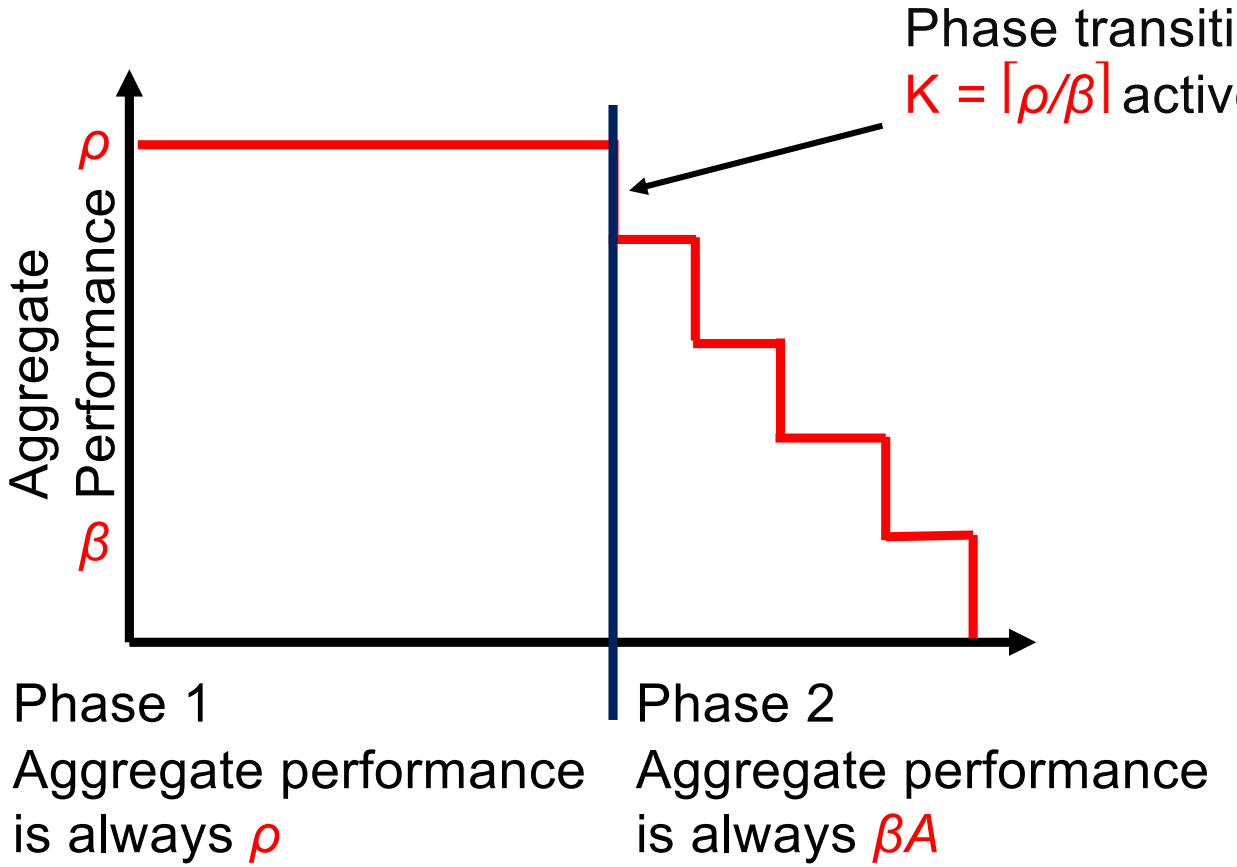
How can we do better? Aggregate Performance!





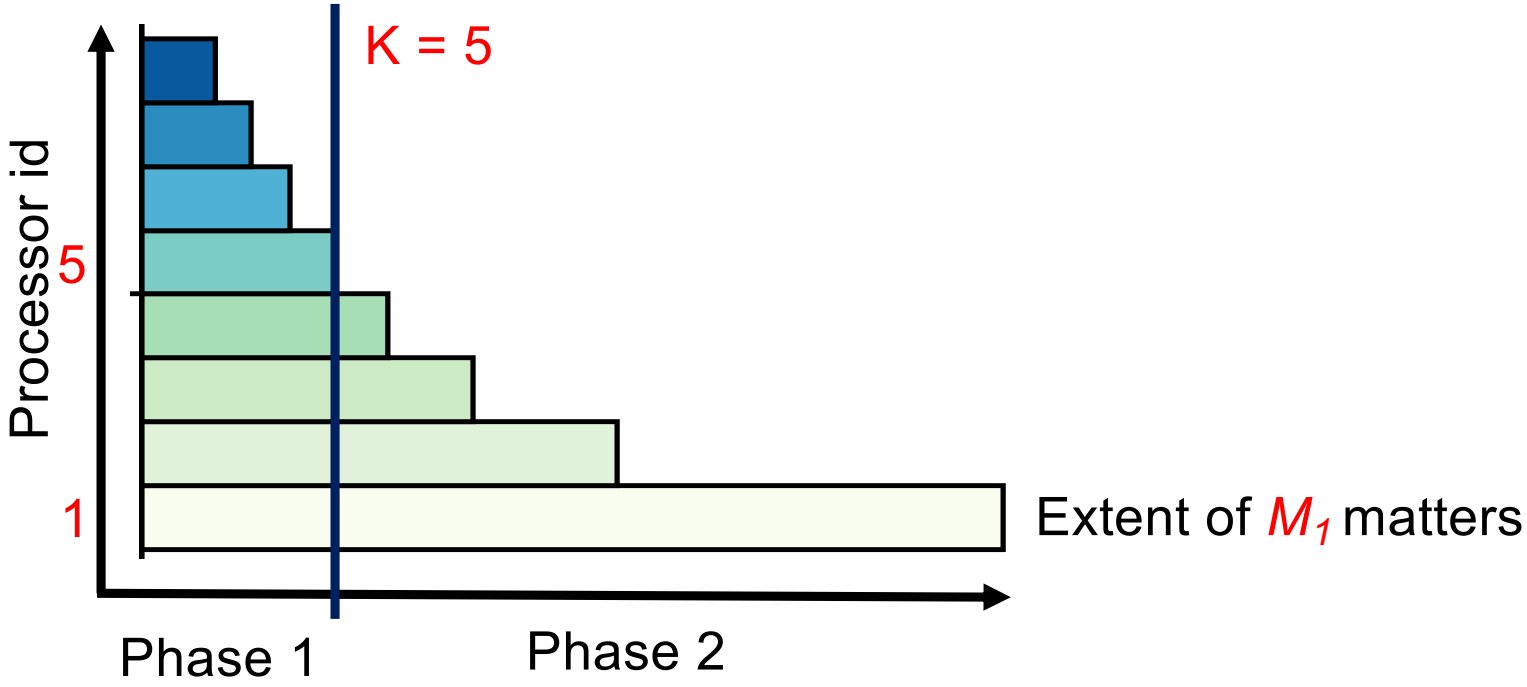
simula

Split Computation in 2 Phases



simula

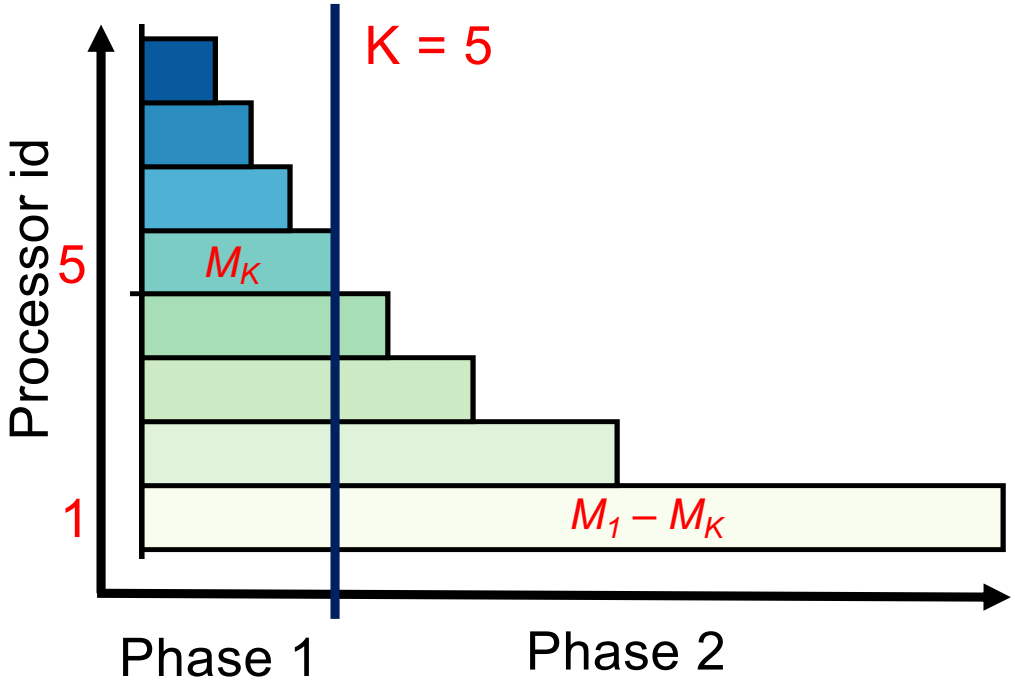
Predicting Phase Duration

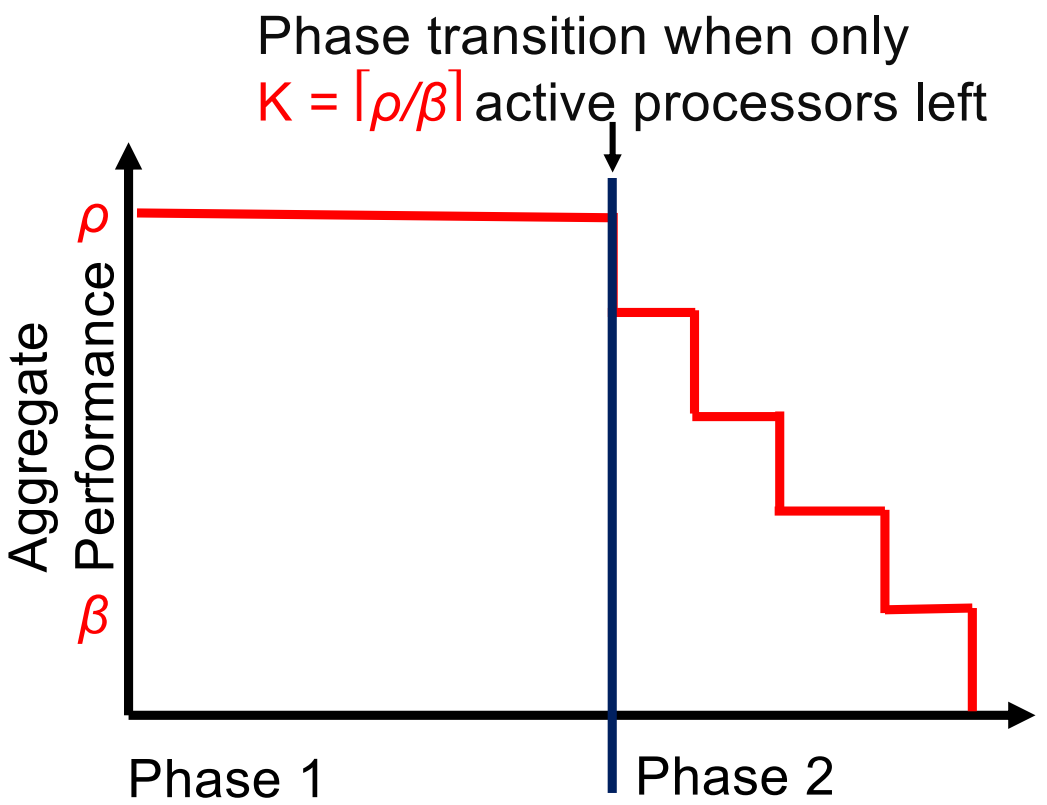


Volume in Phase 1 matters

simula

Predicting Phase Duration





$T_1 = (\sum_{K < i \le P} M_i + K\, M_K) / \rho$ $T_2 = (M_1 - M_K) / \beta$

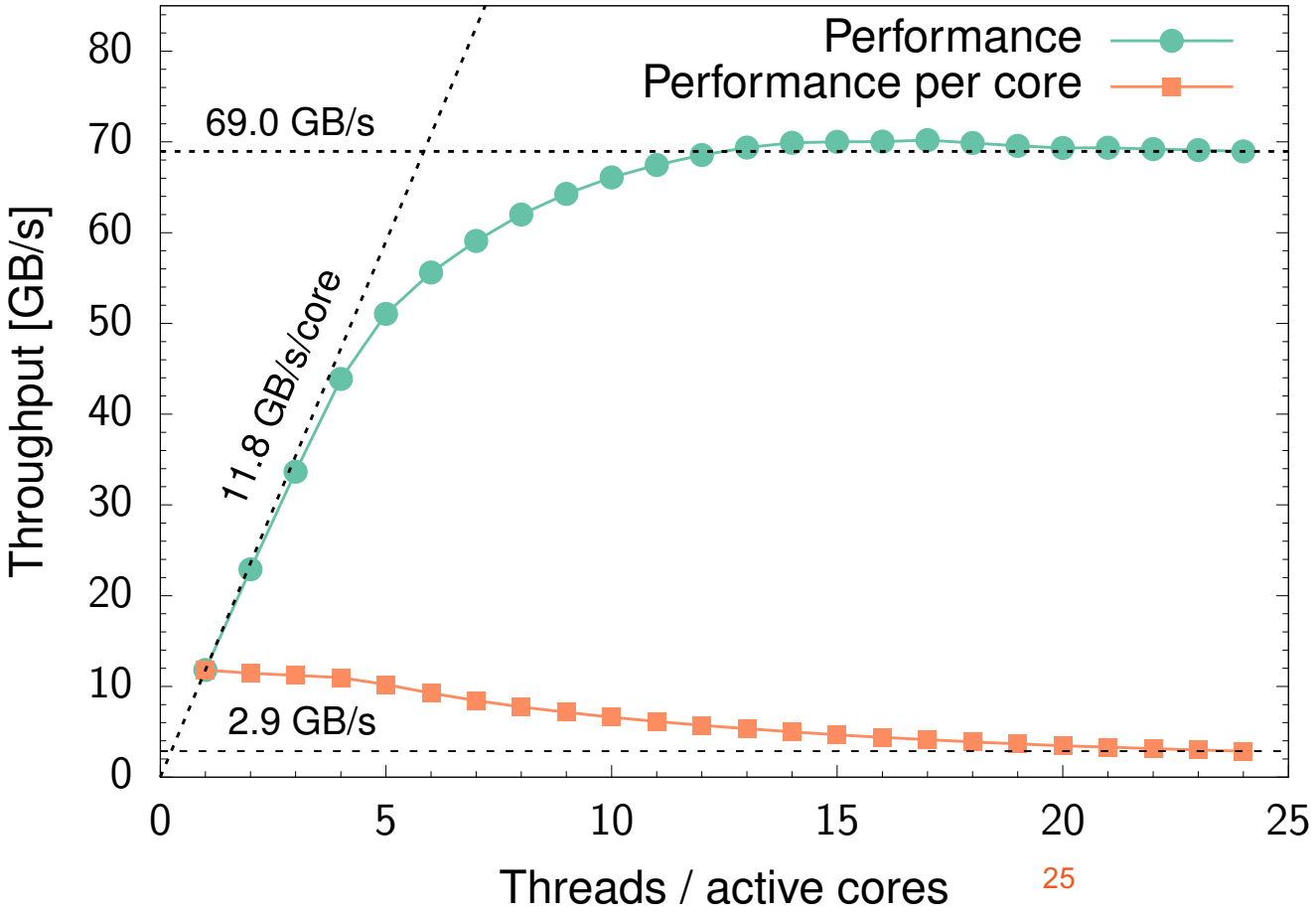
No matter how many processors are active:

- Phase 1 performance will be ρ
- Processor 1 performance will be β in Phase 2

simula

The 2 Phase Model

$$T = (\sum_{K < i \leq P} M_i + K M_K) / \rho + (M_1 - M_K) / \beta$$

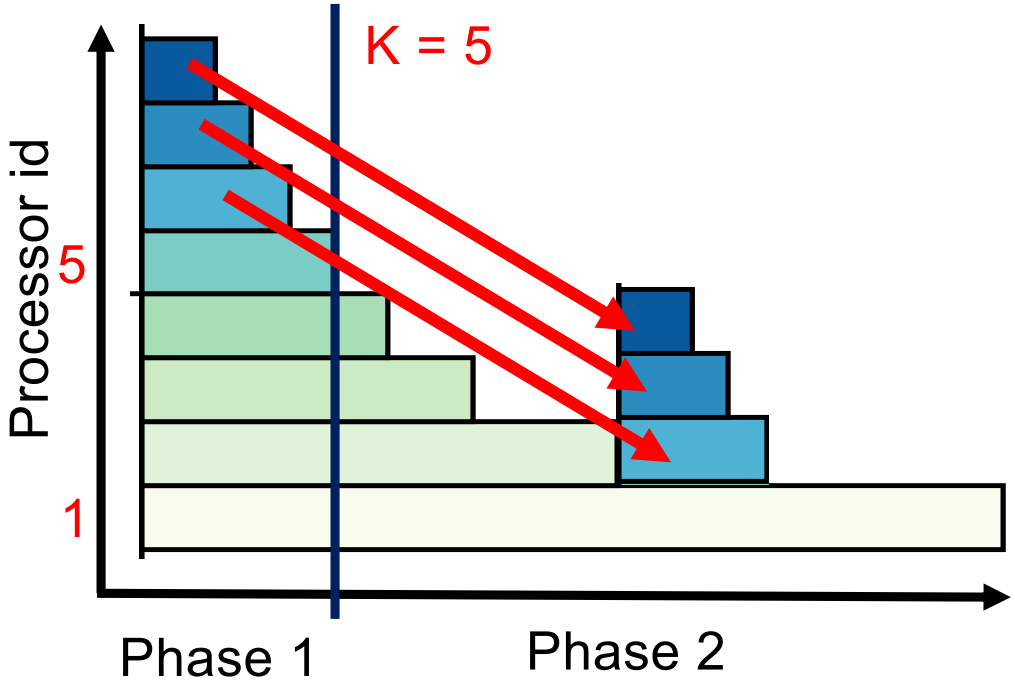


Real processors come close to the model, but phase transition is softer

24-core Intel Xeon Platinum 8168 (Skylake)

simula

Immediate consequence: Fewer Cores can be faster



Extent of M_1 matters:
shorten Phase 1

Volume in Phase 1 matters:
reduce volume by moving it
to Phase 2

simula

Testing the Model: CPU Data

CPU	P	β	ρ	K
AMD Epyc 7302P (Rome)	16	22.83	90.91	5
AMD Epyc 7413 (Milan)	24	31.83	102.58	4
AMD Epyc 7601 (Naples)	32	18.15	85.42	5
AMD Epyc 7763 (Milan)	64	30.93	121.23	4
Intel Xeon Gold 6130	16	13.42	74.74	6
Intel Xeon Platinum 8168	24	11.81	68.96	6
Intel Xeon Platinum 8360Y	36	14.9	158.21	11
Cavium ThunderX2	32	15.51	118.54	8
HiSilicon Kunpeng 920	64	12.35	131.54	11
NVIDIA Grace GH200	72	27.16	316.45	12

AMD: low K value

Intel / ARM

Medium to high K value

simula

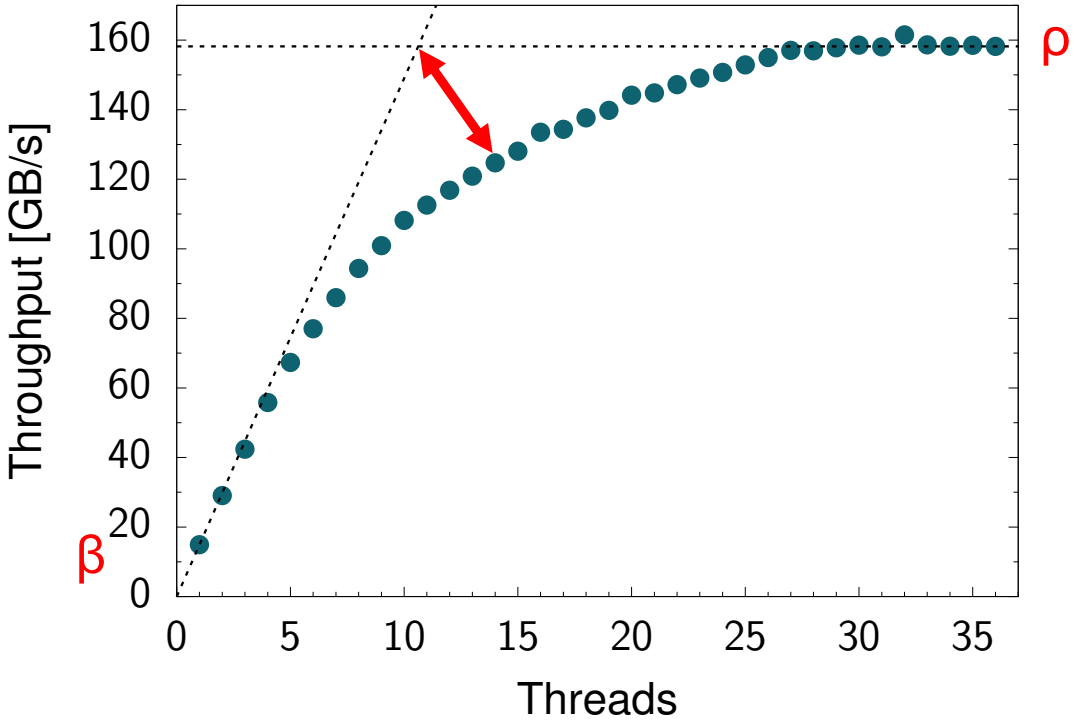
Does it work for Triangular Workloads?

	Measured	Staircase Model		2-Phase Model	
CPU	[GB/s]	[GB/s]	Error	[GB/s]	Error
AMD Epyc 7302P (Rome)	60.24	57.26	-5%	83.16	41%
AMD Epyc 7413 (Milan)	88.21	91.06	3%	100.25	14%
AMD Epyc7601 (Naples)	66.9	72.1	8%	83.02	24%
AMD Epyc7763 (Milan)	127.07	117.41	-8%	120.71	-5%
Intel Xeon Gold 6130	44.1	50.55	15%	63.42	44%
Intel Xeon Platinum 8168	63.98	59.97	-6%	63.61	-1%
Intel Xeon Platinum 8360Y	89.67	129.36	44%	137.77	54%
Cavium Thunder X2	74.09	96.28	30%	108.67	47%
HiSilicon Kunpeng 920	92.46	102.17	10%	125.62	36%
NVIDIA Grace GH200	273.57	309.8	13%	302.77	11%

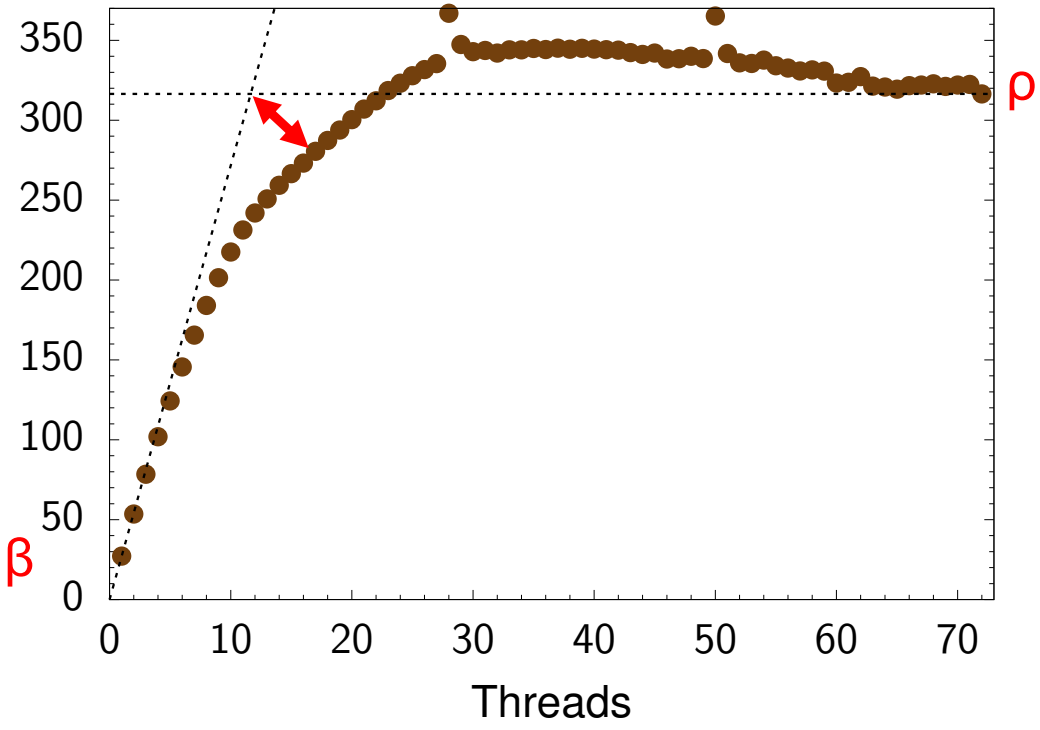
2-Phase is less accurate than Staircase, but much better than simple models

simula

Where does the Difference come from?



Intel Xeon Platinum 8360Y (Ice Lake)



NVIDIA Grace GH200 CPU

CPUs with STREAM results closer the the intersection point can be predicted more accurately.

simula

Does it work for Amdahl Workloads?

CPU	Measured [GB/s]	2-Phase [GB/s]	Model Error
AMD Epyc 7302P (Rome)	33.07	36.49	+10.35%
AMD Epyc 7413 (Milan)	43.52	48.58	+11.62%
AMD Epyc 7601 (Naples)	22.46	29.94	+33.29%
AMD Epyc 7763 (Milan)	47.34	49.28	+4.11%
Intel Xeon Gold 6130	22.07	22.75	+3.10%
Intel Xeon Platinum 8168	20.75	20.16	-2.84%
Intel Xeon Platinum 8360Y	28.14	27.24	-3.22%
Cavium ThunderX2	26.90	27.43	+1.99%
HiSilicon Kunpeng 920	22.21	22.58	+1.67%
NVIDIA Grace GH200	51.84	50.03	-3.49%

Low K value means
AMDs perform well here

- Accurate on all CPUs except Naples
- Identical to Staircase model here

simula

Summary

- Load imbalance causes problems for performance models
- Sample imbalanced workloads are easy to assign
- New model built on Roofline and Max-rate models
- Easy to apply. Uses only P, β, ρ

Summary

- Load imbalance causes problems for performance models
- Sample imbalanced workloads are easy to assign
- New model built on Roofline and Max-rate models
- Easy to apply. Uses only P, β, ρ
- Model is fairly accurate, except for processor-specific behaviour
- Can be used for internode communication (multiple cores, one NIC)
- Main insight: Phase 1 depends on the total, Phase 2 only on M_1
- K and M_K determine phase transition, crucial for performance

Questions?