Load-balancing on heterogeneous parallel servers

Parimal Parag

Electrical Communication Engineering
Indian Institute of Science

Acknowledgements








CENTRE FOR
NETWORKED INTELLIGENCE
Indian Institute of Science

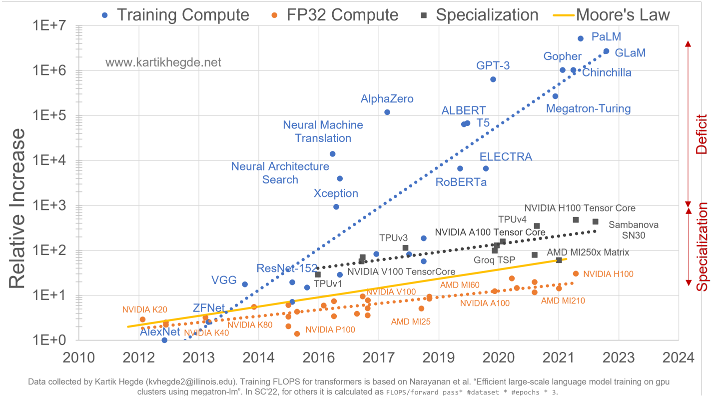
IBM

Qualcomm

NRF National Research Foundation

The growing performance deficit

Deep learning compute demand[1]



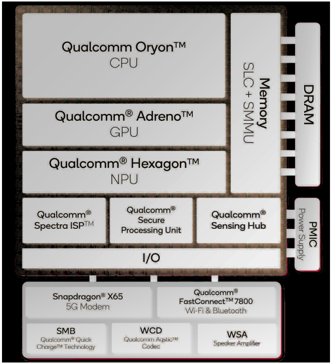
- End of Moore's law and Dennard scaling
- Can no longer keep adding more transistors and increasing core frequencies
- We're accustomed to more and more compute

[1] Image credit: https://kartikhegde.substack.com/p/accelerating-deep-learning-in-the

Heterogeneous Computing[2]

Sources of heterogeneity

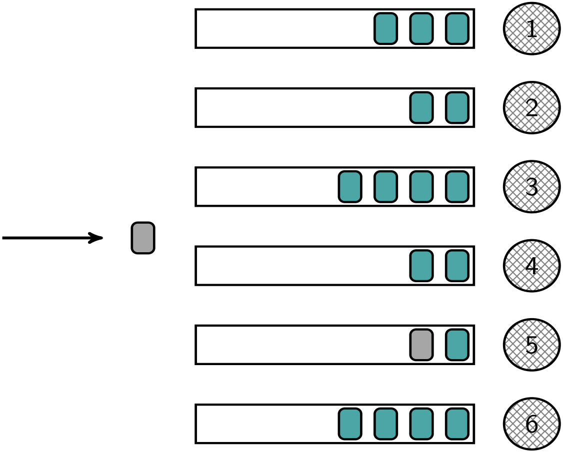
- Different generations of servers and accelerators
- Pooling of all available compute resources (CPUs, GPUs, NPUs)
- Compute resources may be run at different speeds for energy conservation
- Compute cores optimized for different operating regions, to deal with dynamic workloads
- External factors—network bottlenecks, data affinity, etc.



[2] Image: https://www.anandtech.com/show/21445/qualcomm-snapdragon-x-architecture-deep-dive

Load balancing policies—homogeneous servers

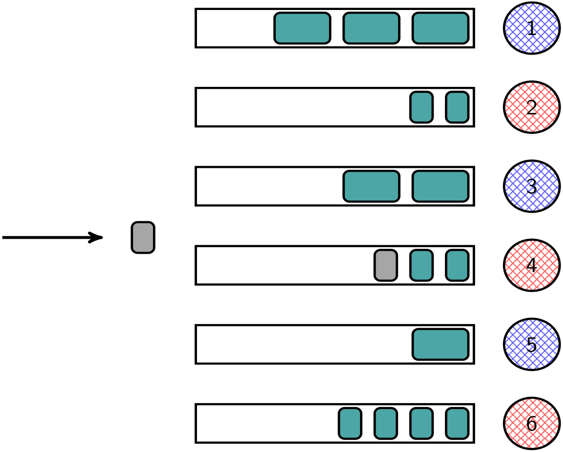
Join shortest queue (1) [3]



[3] W. Winston, "Optimality of the shortest line discipline," J. App. Prob., 14(1), 181–189, Mar 1977.

Load balancing policies—heterogeneous servers

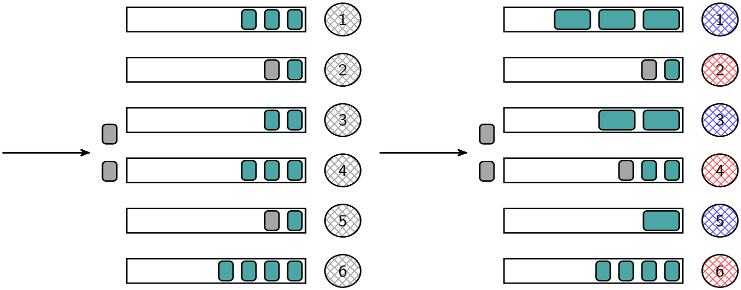
Join smallest workload queue (1) [4]



[4] R. R. Weber, "On the optimal assignment of customers to parallel servers," J. App. Prob., 15(2), 406–413, Jun 1978.

Load balancing policies—parallel processing of subtasks

Join the shortest queue (k)



- Equivalent to (n, k) fork-join system [5]

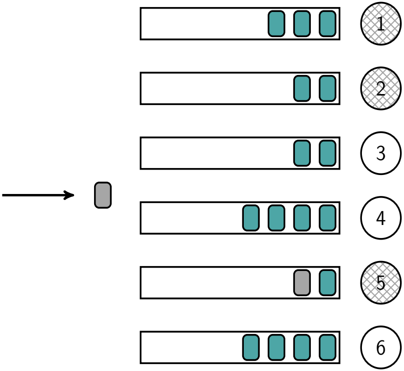
Pro Minimizes the mean task completion time

Con Feedback overhead linearly scaling in the number of servers

[5] A. Badita, P. Parag, and J.-F. Chamberland. Latency analysis for distributed coded storage systems. IEEE Transactions on Information Theory. 65(8):4683–4698, Aug 2019.

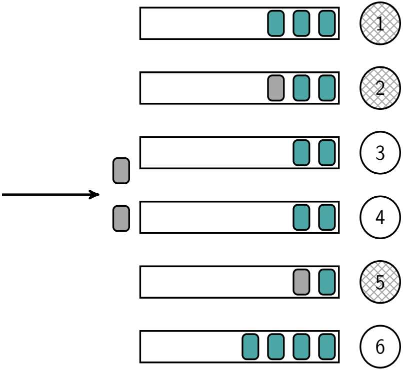
Load balancing policies—low overhead alternative [6]

Power-of-d (1)



- Equivalent to $(d, 1)$ fork-join queue
- When $d = n$, it is JSW

Power-of-d (k)

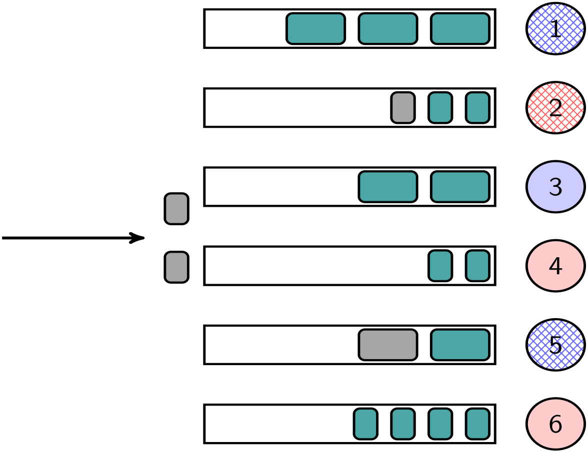


- Equivalent to $(d, 1)$ fork-join queue
- When $d = n$, it is (n, k) fork-join

[6] M. Mitzenmacher, "The power of two choices in randomized load balancing," IEEE Trans. Parallel Distrib. Syst., vol. 12, no. 10, pp. 1094–1104, Oct. 2001.

Heterogeneous servers— low overhead alternative

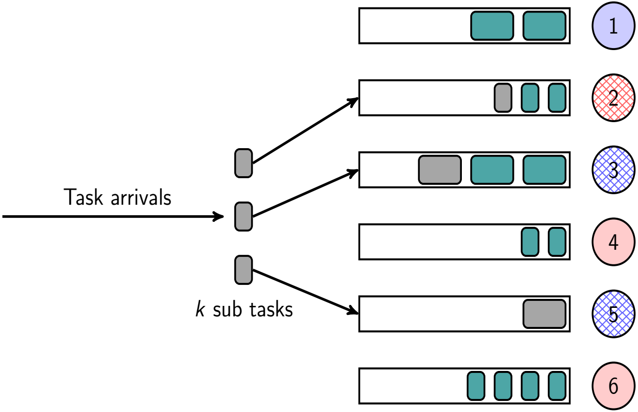
(d, k) fork-join system



- Sample d servers, join k smallest
- Task departs on completion of all k sub-tasks

Heterogeneous servers— zero overhead alternative[7]

(k, k) fork-join system with probabilistic scheduling



Objective Find the optimal slow server selection probability p_s^* that minimizes the mean task completion time

[7] R. Jinan, A. Badita, T. P. Bodas, and P. Parag. Load balancing policies without feedback using timed replicas. *Performance Evaluation*. 162, 102381, Nov 2023.

Related Works

Load balancing strategies in homogeneous system

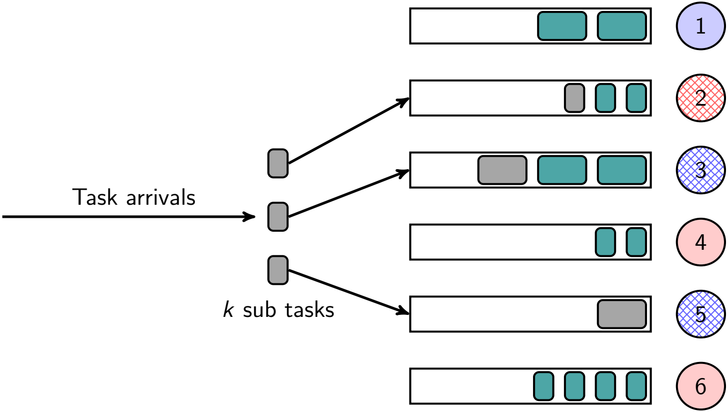
- **Without sub-division of tasks** [M. Mitzenmacher et al., 2001], [U.Ayesta et al., 2019], power-of-d variants
- **With sub-division of tasks** [A.Badita et al., 2019] [R.Jinan et al., 2022],

Load balancing strategies in heterogenous system

- **Without sub-division of tasks**[Der Boor et al., 2021], [Jaleel et al., 2022]: power-of-d variants
- **With sub-division of tasks:** Our work

SystemParameters

Random selection of slow and fast servers

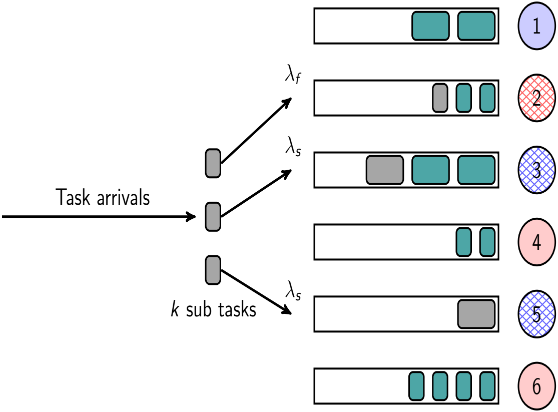


- Probability of selecting k_s slow and $k - k_s$ fast servers for any task is

$$q(k_s) = \binom{k}{k_s} p_s^{k_s} (1 - p_s)^{k-k_s}$$

System Parameters

Arrival rate at individual servers

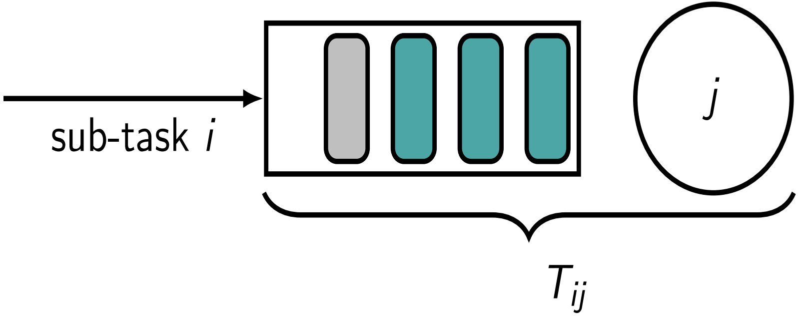


- Arrival at each server is a thinned Poisson process with the arrival rate at slow and fast servers are

$$\lambda_s \triangleq \frac{\lambda n}{k}\left(\frac{kp_s}{nf_s}\right) = \frac{\lambda p_s}{f_s}, \qquad \lambda_f \triangleq \frac{\lambda \bar{p}_s}{\bar{f}_s}.$$

Performance metrics

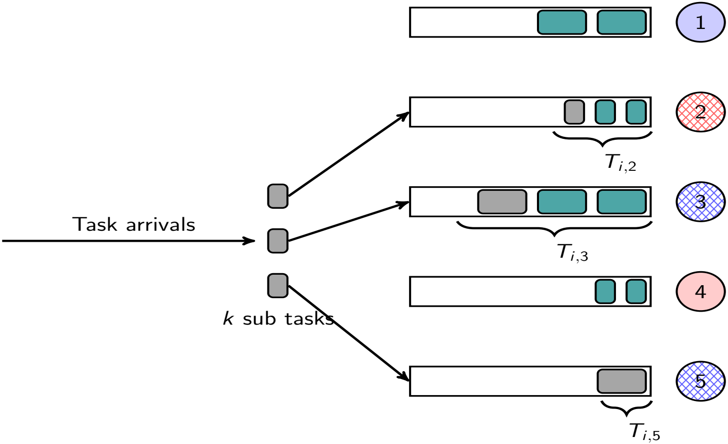
Sub-task completion time



- Sub-task completion time at server j is $T_{i,j} \triangleq W_{i,j} + X_{i,j}$
- For slow server j, $L_s(x) \triangleq \lim_{i\to\infty} P\{T_{i,j} \leqslant x\}$
- For fast server j, $L_f(x) \triangleq \lim_{i\to\infty} P\{T_{i,j} \leqslant x\}$

Performance metrics

Task completion time



$$T_i \triangleq \max_{j \in I^i} T_{i,j}$$

Key contributions

- Establishing asymptotic independence of the stationary workload distribution in a heterogeneous server system with two classes of heterogeneity
- Analytical computation of the limiting mean task completion time for systems with an arbitrarily large number of servers
- A tight closed form approximation for the optimal selection probability

Asymptotic Independence

Theorem

If $\pi^k, \hat{\pi}^k$ are the equilibrium distributions for workloads in the first $k = o(n^{\frac{1}{4}})$ servers of systems $\mathcal{S}$ and $\hat{\mathcal{S}}$ respectively, the total variation distance

$$\lim_{n\to\infty} d_{\mathrm{TV}}(\pi^k, \hat{\pi}^k) = 0$$

Theorem

If asymptotic independence of equilibrium workload at any k queues for a large number of servers holds, then

$$P\left\{T_\infty \leqslant x\right\} = P \cap_{j\in I^\infty} \left\{T_{\infty,j} \leqslant x\right\} = \mathbb{E} \prod_{j\in I^\infty} P\left\{T_{\infty,j} \leqslant x\right\}$$

Therefore, the mean completion time is

$$\mathbb{E}[T] = \sum_{k_s=0}^{k} \binom{k}{k_s} p_s^{k_s}(1-p_s)^{k_s} L_s(x)^{k_s} L_f(x)^{k-k_s} = \int_{x\in\mathbb{R}_+} [1-(p_s L_s(x) + \bar{p}_s L_f(x))^k]dx$$

Asymptotic Independence

When $k(n) = \left\lceil n^{\frac{2}{3}} \right\rceil$

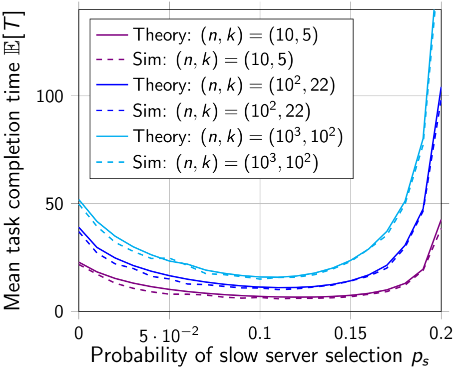


Figure: System with $f_s = 0.5$, $\lambda = 1.2$, $(\mu_s, \mu_f) = (0.5, 2.5)$.

Bound on Mean Task Completion Time

Upper and Lower Bound

- Average smaller than maximum smaller than sum, i.e.

$$\frac{1}{|I^i|}\sum_{j\in I^i} T_{i,j} \leqslant \max_{j\in I^i} T_{i,j} \leqslant \sum_{j\in I^i} T_{i,j}$$

- Mean of sum of sub-task completion times

$$\lim_{i\to\infty} \mathbb{E}\sum_{j\in I^i} T_{i,j} = kp_s \int_{x\in\mathbb{R}_+} \bar{L}_s(x)dx + k\bar{p}_s \int_{x\in\mathbb{R}_+} \bar{L}_f(x)dx$$

- For general service times

$$\lim_{i\to\infty} \mathbb{E}\sum_{j\in I^i} T_{i,j} = \frac{kp_s}{\lambda_s}\left(\rho_s + \frac{\lambda_s^2\mathbb{E}X_s^2}{2(1-\rho_s)}\right) + \frac{k\bar{p}_s}{\lambda_f}\left(\rho_f + \frac{\lambda_f^2\mathbb{E}X_f^2}{2(1-\rho_f)}\right),$$

  where the load $\rho_s = \lambda_s\mathbb{E}X_s < 1$ and $\rho_f = \lambda_f\mathbb{E}X_f < 1$.

Optimal slow server selection probability

Minimizing mean task completion time

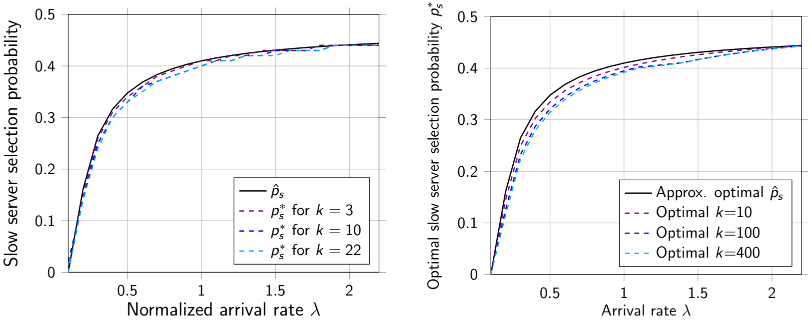
- Find optimal probability p_s^* that minimizes $\mathbb{E}T$ difficult to compute analytically
- Find probability $\hat{p}_s$ that minimizes both the upper and the lower bound
- Approximate p_s^* by $\hat{p}_s$

Exponentially distributed sub-task completion times

Defining $\tau_1 \triangleq \bar{f}_s(\mu_f - \sqrt{\mu_s \mu_f})$,

$$\hat{p}_s = \begin{cases} 0, & \lambda \leqslant \tau_1, \\ \frac{1 - \frac{\tau_1}{\lambda}}{1 + \frac{\bar{f}_s}{f_s}\sqrt{\frac{\mu_f}{\mu_s}}}, & \tau_1 \leqslant \lambda < \mu_s f_s + \mu_f \bar{f}_s. \end{cases}$$

Slow server selection probabilities

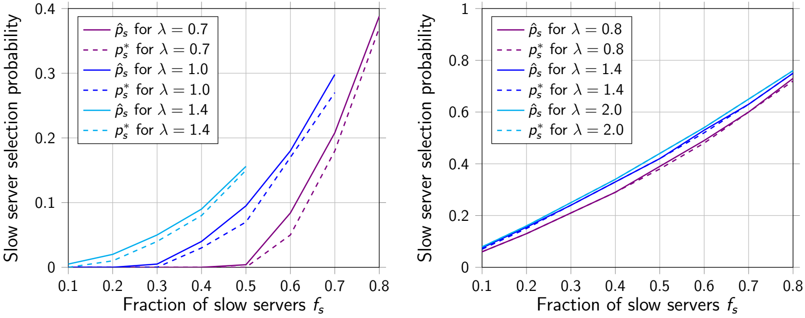


(a) $n = 10^2, k \in \{3, 10, 22\}$.

(b) $n = 10^3, k \in \{10, 100, 400\}$.

Figure: System with $(\mu_s, \mu_f) = (2, 2.5)$.

Variation with the fraction of slow servers f_s



(a) $(\mu_s, \mu_f) = (0.5, 2.5)$. (b) $(\mu_s, \mu_f) = (2.0, 2.5)$.

Figure: System with $n = 10^3$, $k = 10$

Deterministic Scheduling

Deterministic selection

Select k_s slow and $k - k_s$ fast servers

Theorem

The optimal selection probability of slow servers, converges to

$$\lim_{k\to\infty} p_s^* = \frac{k_s^*}{k},$$

where k_s^* is the optimal deterministic selection of slow servers. The optimal probability of choosing ℓ servers converges to

$$\lim_{k\to\infty} q(\ell) = \mathbb{1}_{\{\ell = kp_s^*\}}.$$

Mean number of slow servers

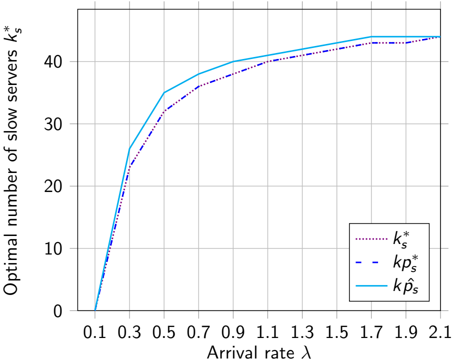
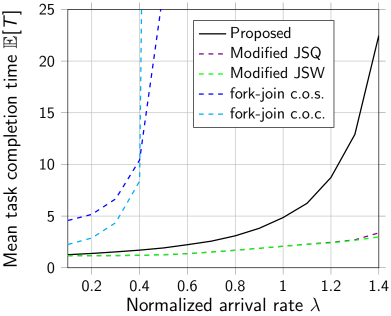
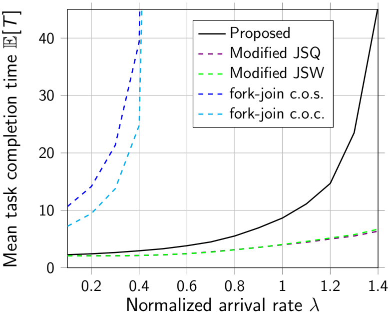


Figure: System with $n = 10^3$, $k = 100$, $f_s = 0.5$,$(\mu_s, \mu_f) = (2, 2.5)$.

Comparison with other Load Balancing Policies



(a) $n = 10^2,\ k = 10$



(b) $n = 10^3,\ k = 10^2$

Figure: System with $f_s = 0.5$, $(\mu_s, \mu_f) = (0.5, 2.5)$

Asymptotic Independence

Theorem

If$\pi^k, \hat{\pi}^k$ are the equilibrium distributions for workloads in the first $k = o(n^{\frac{1}{4}})$ servers of systems $\mathcal{S}$ and $\hat{\mathcal{S}}$ respectively, the total variation distance

$$\lim_{n \to \infty} d_{\mathrm{TV}}(\pi^k, \hat{\pi}^k) = 0.$$

Asymptotic Independence

Proof sketch

We look at three systems with n servers:

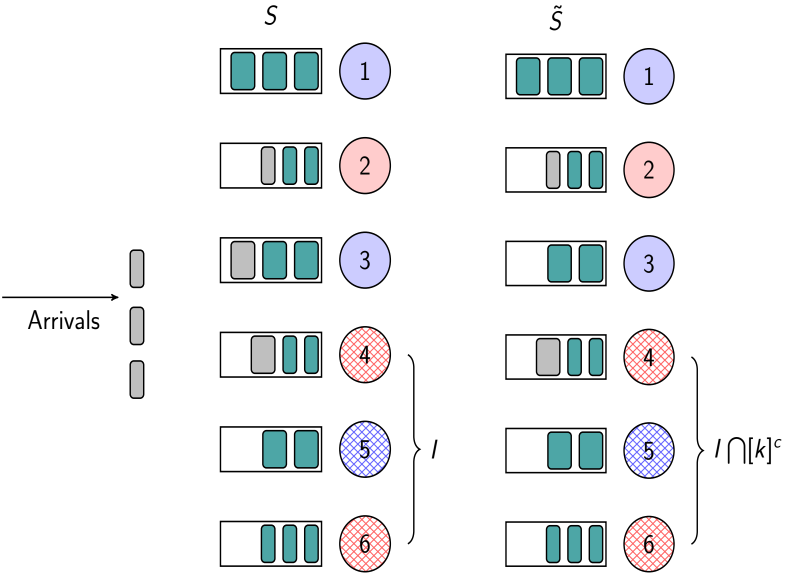
- Original system S under consideration
- Independent system $\hat{S}$, whereall the queues are independent
- Coupled system$\tilde{S}$, where no more than one arrival is allowed in the first k queues

Focus

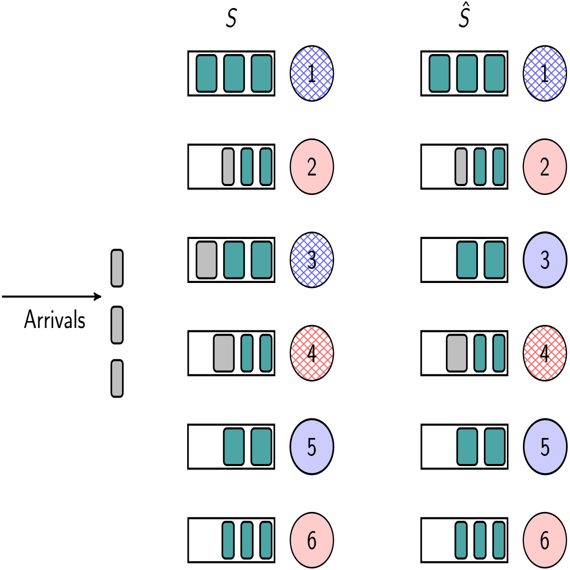
Joint distribution of queues at the first k servers of the original system S, an independent system $\hat{S}$ and a coupled system $\tilde{S}$

$$\begin{aligned} d_{\mathrm{TV}}(\pi^k, \hat{\pi}^k) \leqslant & d_{\mathrm{TV}}(\pi^k, \pi^k_\tau) + d_{\mathrm{TV}}(\pi^k_\tau, \tilde{\pi}^k_\tau) + d_{\mathrm{TV}}(\tilde{\pi}^k_\tau, \tilde{\pi}^k) \\ & + d_{\mathrm{TV}}(\tilde{\pi}^k, \hat{\pi}^k). \end{aligned}$$

Asymptotic Independence—coupled system evolution



Asymptotic Independence—coupled system evolution



Proof steps

- Original and coupled system differ when there is more than one arrival at first k servers
- Coupled system has thinned arrivals since some arrivals to first k servers are dropped
- For a finite time, bound the probability that the workload at first k servers differs between two systems
- At any finite time, workload distribution at first k servers for the original and the coupled systems are close
- For sufficiently large time, the workload distribution is close to stationary distribution
- Workload distribution for coupled and independent systems is close since the load differences are bounded

Conclusion

- We show that the joint distribution of the stationary workload across k queues becomes asymptotically independent as the number of servers, n, grows and $k = o(n^{\frac{1}{4}})$
- We derive an upper and lower bound on the limiting mean response time and identify the selection probability, p_s, that minimizes this bound
- Numerical experiments confirm that the selected probability provides near-optimal performance

How to reduce CPU power?

Energy proportionality

- Push CPUs to sleep.
- Dynamic voltage and frequency scaling (DVFS).

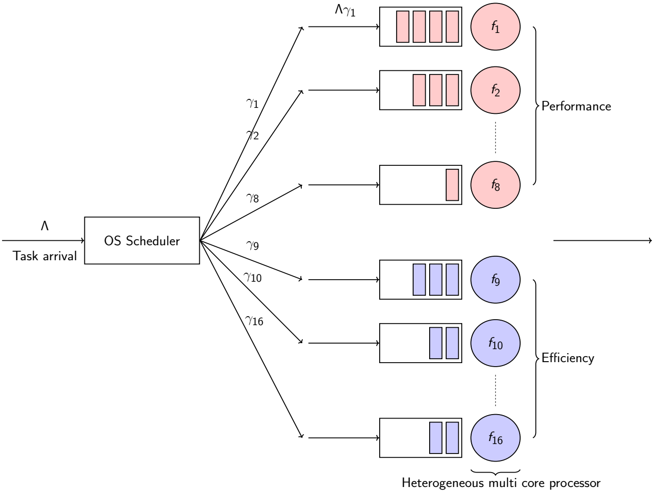
Challenges in Homogeneous multi-cores

- High performance at the cost of high power consumption.
- Power efficient with degraded performance.

Heterogeneous multi-core processor (HMP)

A new architecture consisting of CPUs with heterogeneous cores having different power-performance trade-offs.

Key questions

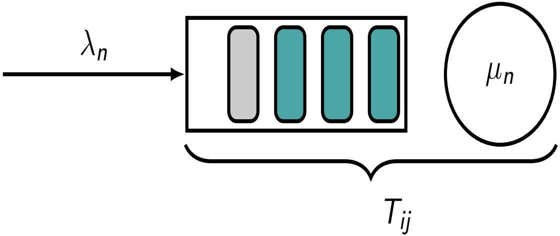


- What is the optimal workload split ?
- What is the operating frequency for all cores?

Key contributions

- Power and performance model for CPUs with heterogeneous cores.
- Problem formulation for the workload splitting and operating frequencies.
- HEMP—*Heterogeneity enabled Energy-Minimizer with Performance constraints*.
- Comparison with Linux frequency governors **(upto 80% reduction in energy-delay product)**.

Problem Formulation



$$\mu_n \triangleq \alpha_{c_n} f_n,$$

$$\rho_n = \frac{\lambda_n}{\mu_n},$$

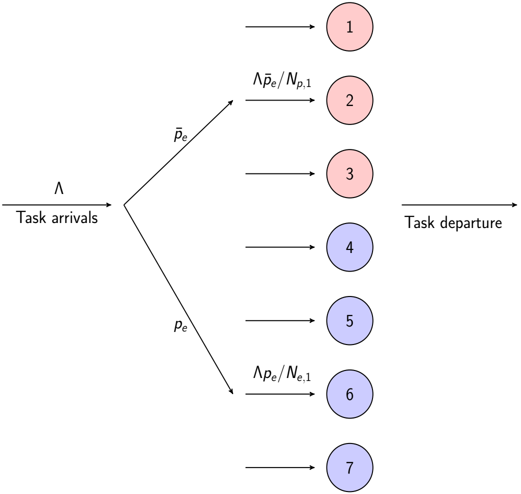
$$\bar{W}_n(c_n, f_n, \gamma_n) = \frac{1}{\mu_n - \lambda_n},$$

$$\bar{P}_n = P_{\text{sta}} + \rho_n P_{\text{dyn}}.$$

$$(\gamma^*, f^*) \triangleq \arg \min_{(\gamma, f) \in A} \sum_{n \in [N]} \bar{P}_n.$$

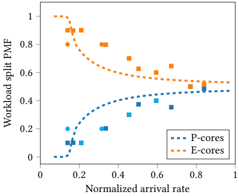
$$\bar{W}_n(c_n, f_n, \gamma_n) \leq w.$$

Optimal selection probability and frequency allocation

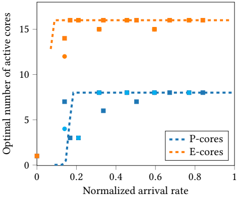


$$f^* \triangleq \frac{1}{\alpha}\Big(\Lambda\gamma + \frac{1}{w}\Big), \qquad \gamma_j^* = \frac{p_e^*}{N_{e,1}^*}\mathbb{1}_{\{j\in E\}} + \frac{1-p_e^*}{N_{p,1}^*}\mathbb{1}_{\{j\in f\}}.$$

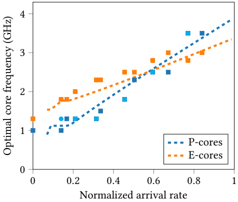
Optimal splitting between classes



Optimal workload split



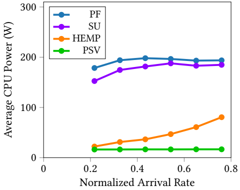
Optimal number of active cores



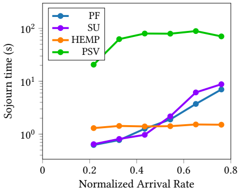
Optimal frequency

- Probabilistic split between two core types
- Constant frequency for all active cores of one type

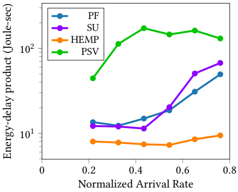
Comparison with Linux frequency governors



Total power



Sojourn time

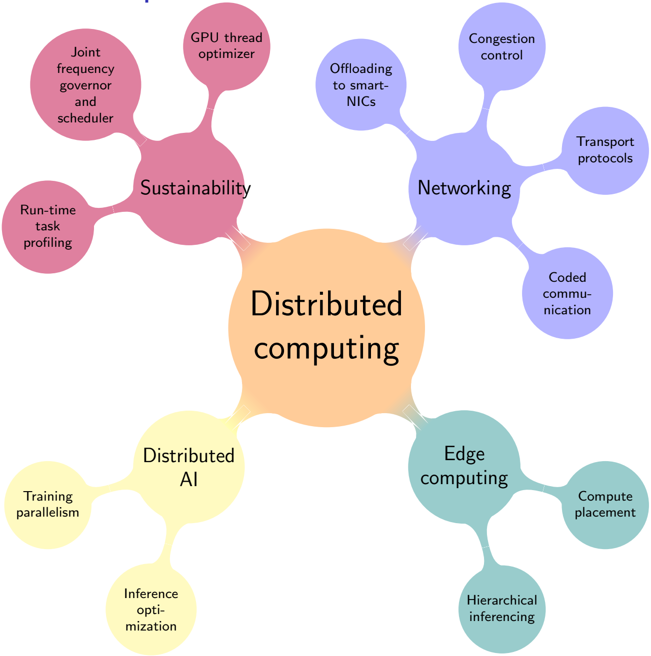


Energy-delay product

TensorFlow Lite workload

PF: Performance **SU**: Schedutil **PSV**: Powersave

Research Landscape



References

- M. Mohanty, G. Gautam, V. Aggarwal, and P. Parag. Analysis of fork-join scheduling on heterogeneous parallel servers. *IEEE/ACM Transactions on Networking*. Jul 2024.
- A. Priya, R. Choudhury, S. Patni, H. Sharma, M. Mohanty, K. Narayanam, U. Devi, P. Moogi, P. Patil, and P. Parag. Energy-minimizing workload splitting and frequency selection for guaranteed performance over heterogeneous cores. *ACM International Conference on Future and Sustainable Energy Systems (e-Energy)*. pp. 308–322, Jun 2024.
- R. Jinan, A. Badita, T. P. Bodas, and P. Parag. Load balancing policies without feedback using timed replicas. *Performance Evaluation*. 162, 102381, Nov 2023.
- R. Jinan, G. Gautam, P. Parag, and V. Aggarwal. Asymptotic analysis of probabilistic scheduling for erasure-coded heterogeneous systems. *ACM SIGMETRICS Performance Evaluation Review*. 50(4):8–10, Mar 2023.
- A. Badita, P. Parag, and J.-F. Chamberland. Latency analysis for distributed coded storage systems. IEEE Transactions on Information Theory. 65(8):4683–4698, Aug 2019.