

SOLUTION OF SOME OPTIMISATION PROBLEMS IN THE DEVELOPMENT OF CENTRALISED AND DECENTRALISED DISTRIBUTED SYSTEMS

KHOSHABA O (pzmag2022@gmail.com)

Vinnitsia National Technical University (Ukraine)

Abstract. This paper addresses optimisation problems that arise in the development of centralised and decentralised distributed systems. We unify the treatment of technology workflows and architectural patterns of corporate computer networks by modelling them as open queueing networks with product-form stationary behaviour (Jackson networks). On that basis we formalise throughput ceilings, latency/availability trade-offs, and actionable levers such as feedback reduction, proportional flow splitting, server pooling, and targeted capacity increments at bottlenecks. We further propose an extended architecture notation that separates protocol and infrastructure security, encodes independent failure domains and consensus type, and binds qualitative labels to measurable KPIs. The approach naturally supports both single-criterion optimisation and multi-objective (Pareto) analysis and aligns with current architecture-description standards.

Keywords: distributed systems; Jackson networks; queueing networks; throughput ceiling; architectural patterns; server pooling; flow splitting; Pareto optimisation; standardisation; corporate computer networks.

Problem statement and motivation. The continued evolution of both centralised and decentralised distributed systems depends on rigorous optimisation of (i) operational technology workflows and (ii) architectural patterns of corporate computer networks. In practice, engineers need compact, defensible models that predict how architectural choices - pooling vs localising, adding channels, tuning load-balancers, or redesigning feedback paths - affect throughput, delay and availability. Classical queueing-network theory provides exactly such a lens: open Jackson networks offer product-form stationary behaviour under mild assumptions and enable interpretable capacity-planning results without heavy simulation [1]-[6]. For decentralised systems, additional constraints stem from quorum-based consensus and broadcast patterns that impose communication overheads and latency-throughput trade-offs [11].

Research tasks and scope. The work resolves four tightly coupled tasks.

(i) It unifies the modelling of centralised (pooled) and decentralised (multi-node, quorum-based) deployments as open Jackson networks whose nodes represent processing stages and network hops, so that both data-plane and control-plane costs are captured [1], [5], [9]-[11].

(ii) It translates architectural patterns into queueing parameters and introduces a throughput-ceiling concept that exposes the true bottleneck under realistic feedback and routing.

(iii) It provides closed-form, engineer-friendly rules for flow splitting, feedback reduction and capacity increments, making the expected benefit of each architectural lever explicit and comparable.

(iv) It frames design as a multi-objective optimisation (throughput, latency, cost, reliability), allowing Pareto reasoning when criteria inevitably conflict [7], [8].

These tasks leverage the author's recent studies, which ground the modelling choices and validation scenarios:

Notations of Architectural Models in Blockchain Networks. Defines a compact notation for local/distributed, single-/multi-node blockchain architectures and shows how the security-scalability-decentralisation trade-offs surface across those models; we reuse this notation to bind performance models to architecture descriptors [14].

Multi-Dimensional Resource Evaluation in Blockchain. Argues for resource assessment beyond one-dimensional gas metering by separating compute, memory access and latency

footprints; we adopt this multidimensional view when mapping consensus/control overheads to queueing loads [15].

RATEM: A Resource-Aware Transaction Evaluation Framework. A preprint [16] proposing fairness- and scalability-oriented transaction evaluation aligned with resource metrics; in our model this corresponds to load-aware routing/weighting that equalises effective utilisations across stages.

Assessing the utilisation of computing resources in blockchain networks. Conference theses that classify workloads from scalar to tensor operations and motivate unconventional metrics (FLOPS, memory bandwidth, execution time); that taxonomy informs our treatment of compute-bound stages [17].

Differential Load Modelling and Multi-objective Optimisation of BFT Parameters in Hyperledger Besu. A preprint introducing time-varying arrival-rate modelling and stability conditions for BFT-based permissioned networks; we reflect these constraints in consensus-aware staging and Pareto analysis [18].

Evaluating and Benchmarking Resilient Web3 Architectures for Multimedia. Conference theses outlining end-to-end benchmarks (cost/GB, time-to-first-frame, rebuffering, carbon footprint) for storage/availability/scaling stacks; we use these scenarios as external validation targets [19].

Methodological foundations. We adopt open queueing networks in the Jackson sense, where each processing stage or link is represented as a service node with service rate μ_i , and requests arrive exogenously at rate λ_0 . Let P be the routing matrix with elements p_{ij} (probability a job leaving node i goes to node j). The traffic equations read:

$$\lambda = \lambda_0 \mathbf{e} + \lambda P \iff \lambda(I - P) = \lambda_0 \mathbf{e},$$

where λ is the vector of throughputs per node and $\mathbf{e}$ is the unit row-vector. The visit ratios $v_i := \lambda_i / \lambda_0$ are thus determined by $\mathbf{v}^\top = \mathbf{e}^\top (I - P)^{-1}$. For M/M/1 nodes, the utilisation at node, the utilisation at node i under network input X equals $\rho_i = X v_i / \mu_i$, and stability requires $\rho_i < 1$ for all i . The throughput ceiling (maximal sustainable input rate) follows directly as:

$$X_{\max} = \min_i \frac{\mu_i}{v_i}.$$

Under standard reversibility assumptions, these networks admit a product-form stationary distribution, yielding transparent expressions for steady-state measures (utilisations, queue lengths, mean sojourn times) and enabling interpretable capacity planning without heavy simulation [1]-[6]. For architects, the key benefit is interpretability: bottlenecks surface as the stages whose effective capacity per visit saturates first [5].

Optimising technology workflows. Workflow optimisation can be posed as either a single-criterion problem - maximising sustainable input subject to latency/utilisation bounds - or as a multi-objective problem that balances throughput, mean delay and cost [7], [8]. In the single-criterion setting, monotonicity properties of $X_{\max}$ and convex-optimisation surrogates give practical tuning heuristics for service allocations and routing biases [8]. In the multi-objective setting, Pareto fronts emerge naturally: decreasing tail latency via duplication/retries often increases work-in-system and cost. Our framework places each candidate design on the Pareto frontier, making trade-offs explicit for decision-makers [7].

Optimising architectural patterns. We express familiar patterns in queueing-network language so that their effect on throughput and latency is auditable rather as is:

Server pooling. A single logical queue feeding a worker pool reduces imbalance and waiting relative to disjoint parallel queues; this is a first-line tactic for centralised designs and for “local” stages inside decentralised stacks [5], [9], [10].

Proportional flow splitting. Allocate traffic across parallel branches in proportion to their effective service capacities μ_i , which equalises utilisations (ρ_i) and elevates $X_{\max}$ without new hardware [5].

Feedback reduction. Removing/suppressing backflows due to retries, re-validation or “return for rework” increases the share of requests that exit per pass and raises sustainable input; engineering counterparts include idempotency, deduplication, earlier validation/caching [5].

Targeted capacity increments. Adding service capacity at the active bottleneck yields the highest marginal gain; the model provides a per-channel “return on investment” that ranks where to add workers next.

Consensus-aware staging (decentralised deployments). Collocate compute with state to cut cross-site messages and minimise the set of nodes on the critical quorum; this has first-order impact on both throughput and confirmation latency [11].

Each lever admits compact rules backed by the product-form structure and the $X_{\max}$ expression above, and was validated in the case study.

Extended architecture notation and standardisation. To compare centralised and decentralised designs consistently, we enrich common S–B–D taxonomies (scalability, security, decentralisation) [14] by decoupling protocol from infrastructure security, encoding the number of independent failure domains, and recording the consensus style and quorum rule where applicable. We propose the tuple:

$\langle \text{ProtSec}, \text{InfraSec}, \text{FDomains}, \text{Consensus}, \text{Quorum}, \text{DataPlane}, \text{ControlPlane} \rangle$,

bound to measurable KPIs (sustainable throughput, confirmation-delay percentiles, availability) so that qualitative labels map to operational targets. This notation sits comfortably within established architecture-description practice and helps align performance models with enterprise-architecture processes and repositories [12], [13]. It also supports standardisation of architectural patterns - pooling, proportional splitting, feedback control, and staged consensus - as reusable, testable building blocks with quantified performance envelopes.

Conclusions. We presented a unified treatment of centralised and decentralised distributed systems by modelling technology workflows and corporate network architectures as open Jackson networks with product-form stationary behaviour. This lens yields operationally meaningful throughput ceilings, makes latency/availability trade-offs explicit, and turns common architectural moves - feedback reduction, proportional flow splitting, server pooling, and targeted capacity increments at bottlenecks - into auditable, closed-form levers. We further introduced an extended architecture notation that separates protocol from infrastructure security, encodes independent failure domains and consensus type, and binds qualitative labels to measurable KPIs. The result is a reproducible, standard-aligned pathway from architectural intent to performance outcomes: it supports both single-criterion tuning and multi-objective (Pareto) analysis, integrates naturally with enterprise architecture practice, and provides concise rules of thumb that can be applied without heavy simulation. In short, the approach closes the gap between architectural description and capacity planning, offering actionable guidance for designing and operating resilient, efficient distributed systems.

References:

- [1] J. R. Jackson, “Networks of waiting lines,” *Operations Research*, vol. 5, no. 4, pp. 518–521, 1957.
- [2] F. P. Kelly, *Reversibility and Stochastic Networks*. Chichester, U.K.: Wiley, 1979.
- [3] L. Kleinrock, *Queueing Systems, Vol. I: Theory*. New York, NY, USA: Wiley, 1975.
- [4] R. J. Boucherie and N. M. van Dijk, *Queueing Networks: A Fundamental Approach*. New York, NY, USA: Springer, 2011.
- [5] S. Harchol-Balter, *Performance Modeling and Design of Computer Systems*. Cambridge, U.K.: Cambridge University Press, 2013.
- [6] H. Chen and D. Yao, *Fundamentals of Queueing Networks*. New York, NY, USA: Springer, 2001.
- [7] K. Deb, *Multi-Objective Optimisation Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.
- [8] S. Boyd and L. Vandenberghe, *Convex Optimisation*. Cambridge, U.K.: Cambridge University Press, 2004.
- [9] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1992.

- [10] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. Harlow, U.K.: Pearson, 2011.
- [11] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in *Proc. 3rd Symposium on Operating Systems Design and Implementation (OSDI)*, New Orleans, LA, USA, 1999, pp. 173–186.
- [12] The Open Group, "The TOGAF® Standard, 10th Edition." [Online]. Available: <https://pubs.opengroup.org/togaf-standard> [Accessed: October 17, 2025].
- [13] ISO/IEC/IEEE 42010:2022, *Systems and Software Engineering—Architecture Description*.
- [14] O. Khoshaba, "Notations of Architectural Models in Blockchain Networks," in S. Kotlyk (Ed.), *Digital Transformation and Intelligent Systems: Theory, Models, Practice*. Iowa State University Digital Press, 2025, pp. 347–391. DOI: 10.31274/isudp.2025.197.08.
- [15] O. Khoshaba, "Multi-Dimensional Resource Evaluation in Blockchain," *Computer-Integrated Technologies: Education, Science, Production*, no. 59, 2025, pp. 276–284. DOI: 10.36910/6775-2524-0560-2025-59-35.
- [16] O. Khoshaba, "RATEM: A Resource-Aware Transaction Evaluation Framework for Scalable and Fair Blockchain Systems," Zenodo preprint, DOI: 10.5281/zenodo.15725686, published June 24, 2025. Available: <https://doi.org/10.5281/zenodo.15725686> (accessed: 2025-06-25).
- [17] O. Khoshaba, "Assessing the utilisation of computing resources in blockchain networks: moving to unconventional methods," *Proceedings of the X International Scientific and Practical Conference "Information Technologies in Education, Science and Production (ITONV-2025)"*, Lutsk, Viddil Imidzhu ta Promotsii LNTU, 2025, pp. 421–424.
- [18] O. Khoshaba, "Differential Load Modeling and Multiobjective Optimisation of BFT Consensus Parameters in Hyperledger Besu," Zenodo preprint, DOI: 10.5281/zenodo.16836406, published August 13, 2025. Available: <https://doi.org/10.5281/zenodo.16836406> (accessed: 2025-08-13).
- [19] O. Khoshaba, "Evaluating and Benchmarking Resilient Web3 Architectures for Multimedia: Storage, Data Availability, and Scaling," *Proceedings of the V All-Ukrainian Scientific and Technical Conference of Young Scientists, Postgraduates and Students "Computer Games and Multimedia as an Innovative Approach to Communication"*, Odesa, ONTU Publishing, 2025, pp. 218–222.