

The MKA Law and the Milutinović–Katulevskiy Architecture: A Post-Moore Roadmap for AI-Scale, Photonic, and Quantum-Adjacent Computing

Katulevskiy, Daniil; Katulevskaya, Elizaveta;
Katulevskiy, Aleksey; and Milutinović, Veljko

Abstract: Progress beyond conventional transistor scaling increasingly depends on whether gains in devices, memory, interconnects, compilers, and system software survive integration into complete workloads. The Milutinović–Katulevskiy Architecture (MKA) develops a common research framework for this problem. Building on Veljko Milutinović’s Honeycomb architecture and graph-mapping lineage, MKA treats computation, stored state, communication, placement, and physical substrate as one architectural design space. The article formulates a broader system-level principle: a local hardware improvement counts as genuine architectural progress only when its advantage survives data movement, state placement, interfaces and conversion, scheduling and queueing, accuracy, reliability, and service-level constraints at the boundary of completed useful work. The proposed MKA Law operationalizes this principle through the energy–time product of completed workloads under fixed quality and service requirements.

Communication bounds, placement models, uncertainty-aware optimization, and substrate-specific break-even conditions determine when locality, reuse, heterogeneous execution, and changes of physical medium can improve the whole system. An exact reference planner makes one part executable. Across a synthetic planning study, coordinated placement improves feasible plans relative to local and fixed-placement strategies; strong global alternatives recover most of that benefit. Full accounting for movement and setup prevents decisions that look attractive locally but fail system constraints. These results establish a starting point for calibrated hardware, concurrent graph workloads, and experimental implementations.

MKA follows the quantitative architecture tradition developed by John L. Hennessy and David A. Patterson, where workload, compiler, machine, memory, communication, and implementation technology are evaluated together. Oskar Mencer’s dataflow work supplies an implementation precedent for compiler-directed spatial execution. Mor Harchol-Balder’s queueing and server-power research connects execution capacity to delivered service. Dimitris Bertsimas’s robust and prescriptive optimization provides the uncertainty-aware decision layer. Daniela Rus’s graph grammars, decentralized control, and physical co-design motivate search over realizable, controllable structures.

MKA organizes these ideas into six regimes spanning reconfigurable digital fabrics, analog computation, processing near memory, photonic inter-

connects, quantum-dot systems, and entanglement-assisted architectures for quantum-native workloads. Artificial intelligence infrastructure provides an immediate high-pressure application, while scientific simulation, finance, genomics, telecommunications, and other graph-intensive domains broaden the research space.

At datacenter scale, sustained system-level efficiency improvements could translate into major reductions in energy use and operating cost; in science, the same gains could enlarge the computations achievable within finite power and time budgets. MKA therefore proposes more than an individual accelerator design: it establishes a shared architectural language and an experimentally falsifiable research direction intended to carry useful design knowledge across successive generations of computing technology.

Prospectively, the MKA Law can be applied to open-weight LLM serving as a six-vector optimization over compute and precision, resident weights, KV and request state, expert and interconnect routing, batching and queueing, and active capacity with its power and cooling burden. Llama 4 Maverick, GLM-5.3, Kimi K3, and Qwen3.8 expose different resource demands, motivating model-specific operating-point searches. At hyperscale, the economic leverage is substantial: under a 5-GW average AI-serving load, PUE 1.20, and electricity at \$0.07/kWh, every 10% reduction in system-level IT energy per completed request corresponds to approximately \$368 million in annual electricity savings at fixed demand and unchanged PUE. A 30% reduction would exceed \$1.1 billion annually. These figures are sensitivity thresholds, not assumed MKA performance, and illustrate the scale at which experimentally measured system-level gains would translate into economic impact.

Index Terms: Computer architecture, dataflow, energy–delay product, graph placement, heterogeneous computing, large language models, queueing, robust optimization.

1. INTRODUCTION

SELECTING an efficient kernel can leave an inefficient application. Each change of execution site moves an intermediate value, may require a conversion, and can activate resources whose setup must be repaid. In the reference study below,

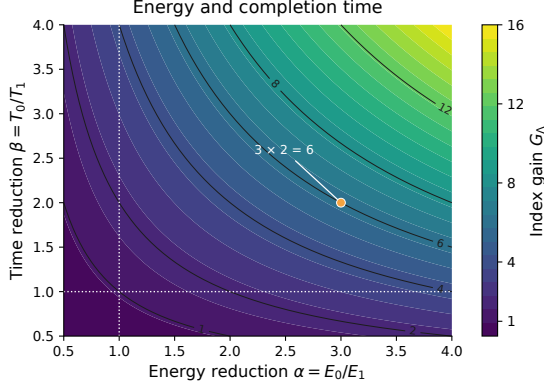


Fig. 1. Index gain $G_A = \alpha\beta$. Dotted lines mark unchanged energy and time; the upper-right region improves both. The marked example gives sixfold gain. MKA impact: simultaneous energy and latency improvements become visible in one completed-work metric.

a planner that omits transfers chooses deadline-violating plans in 189 of 540 cases. The same operations and available sites admit a feasible plan in every case. The difference is the information used to make the placement decision. Workload-based evaluation and computing-energy analysis motivate charging these costs to the complete application [1], [2].

MKA proposes a common description of operations, stored state, physical connections, and service requirements. Honeycomb research supplies the local-organization hypothesis. The research program asks where that organization improves a completed workload and how the decision changes when a computation moves to another substrate. AI serving is an immediate application, but scientific simulation, graph analytics, communication receivers, and numerical finance present related costs.

The proposed *MKA Law* compares energy and normalized completion time at fixed quality. It uses an inverse energy–delay product, with the individual quantities retained. Moore’s integrated-circuit density trend supplied an earlier device-level target [3]; here the engineering target is completed useful work. The underlying multiplicative identity is algebraic. The substantive questions concern the mappings and operating conditions that improve it.

We contribute an executable planning subproblem, an exactness argument, and a complete computational comparison. The planner tracks intermediate ownership and configured sites, making transfer and setup costs part of the choice. The study includes exact single-objective alternatives and cases where joint planning ties them. Its synthetic inputs isolate the decision mechanism; component calibration and general graph mapping remain separate experimental tasks.

The decoder example shows why these decisions matter at scale. At 32K context, each active

sequence needs 10 GiB of KV state. Raising batch size from 16 to 128 lowers modeled reads per token by about 39% while increasing resident block weights and KV state almost fivefold. FlashAttention and PagedAttention demonstrate the practical value of optimizing movement and allocation [4], [5]. The MKA program connects such decisions to substrate interfaces, uncertainty, and the service experienced by users.

Six regimes cover spatial digital execution, analog operations, processing near memory, photonic links, quantum dots, and entanglement-assisted systems. They can coexist, and their experiments have different maturity requirements. We recommend validating a digital mapping on existing hardware first, then testing whichever interface or communication cost remains limiting. The broader program is organized around reusable measurements and operating regions.

2. ARCHITECTURAL FOUNDATIONS

2.1. Honeycomb Placement and the Cost of Regularity

Honeycomb research maps interconnection structures and computational graphs onto regular configurable cells [6], [7]. Hoch’s Purdue study documents an early prototype under Veljko Milutinović’s supervision [8]. V. Milutinović, D. Milutinović, and Furht subsequently developed the mapping patterns and their spatial and timing costs [9].

For one family in that analysis, the required cell count is

$$N_{\text{cells}} = N_{\text{pe}} \left(\frac{N_{\text{pe}}}{2} + 1 + M_c \right), \quad (1)$$

where $N_{\text{pe}} = 2^\ell$ and M_c counts mapped memory cells per processing element. Occupancy includes routing and storage, so it differs from useful processor activity.

Figure 2 shows regularity’s trade-off: fewer global hops require more connectivity. Link length and bandwidth then determine physical latency.

Each Honeycomb-inspired implementation must specify its adjacency, operations, and buffers. Hexagonal geometry alone is ambiguous: tiling vertices have three neighbors, whereas cells joined across their sides have six.

2.2. Hennessy: Evaluate the Compiler and Machine Together

John L. Hennessy’s work supplies the evaluation method. MIPS jointly designed compiler and processor [10]; his VLSI study related architecture to implementation constraints [11]. A graph representation likewise succeeds through the schedules its hardware can execute.

Hennessy and Jouppi studied technology-driven architectural choices [12]; Stanford FLASH extended co-design to memory and communication [13]. These costs remain part of the ma-

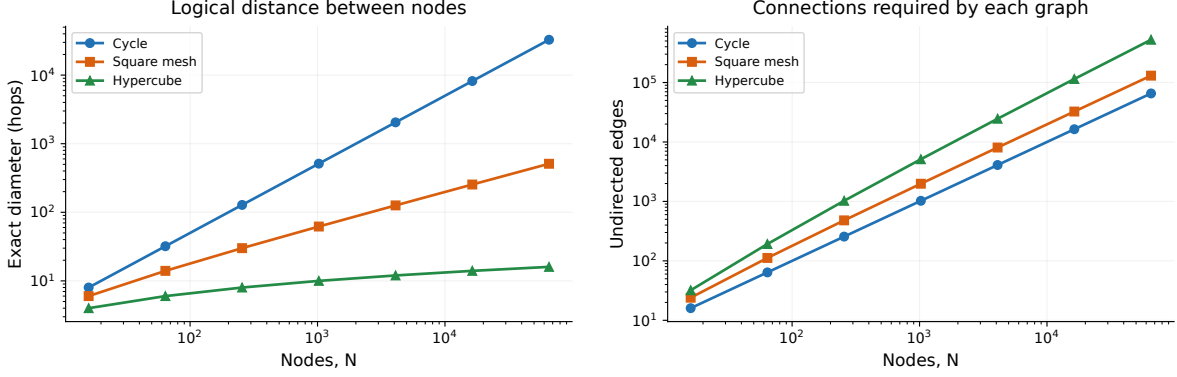


Fig. 2. Exact graph quantities for $N = q^2 = 4^k$. Cycle, square-mesh, and hypercube diameters (left) are $N/2$, $2(q-1)$, and $\log_2 N$; edge counts (right) are N , $2q(q-1)$, and $N \log_2 N/2$. MKA impact: shorter logical paths are valuable only when the added connectivity and physical cost also fit the workload budget.

chine when specialization relocates them. Hennessy and Patterson’s post-Moore account places domain-specific design within this workload-based method [14].

2.3. Mencer: Make Dataflow Executable

Oskar Mencer’s studies of extreme-performance dataflow and predictable reconfigurable systems connect application analysis with resource allocation and bottleneck prediction [15], [16]. Convolutional networks on dataflow engines provide an application example [17].

Buffers and synchronization govern scheduled dataflow. Static pipelines suit regular streams; irregular arrivals need a runtime. Harchol-Balter, Bertsimas, and Rus supply methods for service, uncertainty, and buildable structures.

2.4. Related Tools and the Contribution of the Framework

Timeloop explores DNN mappings and memory hierarchies [18]; MAESTRO analyzes reuse, occupancy, and dataflow cost [19]. Accelerger supplies hierarchical component energy estimates [20], while CiMLoop models interactions between representations, circuits, and compute-in-memory workloads [21]. Halide separates algorithm semantics from storage and execution schedules [22]. Fern further exposes production and consumption patterns to compose black-box subroutines with better locality [23].

The 2024 IRDS *Systems and Architectures* chapter already connects application requirements to heterogeneous computing, memory fabrics, electronic and photonic communication, and whole-stack energy accounting [24]. Its cloud, edge, and cyber-physical categories show that system-level evaluation is an established roadmap concern. MKA specializes that perspective through an explicit workload contract, ownership and interface accounting, and a reproducible planning subproblem. The six regimes organize candidate mechanisms

for that evaluation; the general need for cross-layer design predates this framework.

MKA builds on these established ideas. Its proposed integration couples ownership and interface costs to admissibility, service power, and a substrate research agenda. The reference planner is a small, inspectable specialization of multiobjective path labeling [25]. It exposes exactly when a local decision loses to a joint one. The comparisons below are between fully specified policies in that model; they do not benchmark Timeloop, MAESTRO, Halide, or fabricated hardware. Calibrated cost adapters from such tools are a concrete next integration step.

3. MEASURING COMPLETED WORK

3.1. Workload and Service Requirements

Inference fixes the model, request distribution, and acceptable error; a solver fixes the matrix family and accuracy. The workload contract is

$$\mathcal{C}_W = (\mathcal{D}, \mathcal{Q}, \mathcal{L}, p_{\max}, \mathcal{B}), \quad (2)$$

where $\mathcal{D}$ specifies inputs and demand, $\mathcal{Q}$ output quality, $\mathcal{L}$ timing and throughput, $p_{\max}$ allowed failure probability, and $\mathcal{B}$ the measurement boundary.

The predicate $\mathcal{Q}(A) = 1$ accepts all required accuracy floors and error ceilings. Admissible implementations satisfy

$$\mathcal{A}_W = \{A : \mathcal{Q}(A) = 1, P_{\text{fail}}(A) \leq p_{\max}, A \models \mathcal{L}\}. \quad (3)$$

Margins are fixed before evaluation; a looser quality target defines a different comparison.

A *fixed-work experiment* fixes the job or batch, initial model/data residency, cache and power states, and setup reuse. Measure $T_W = t_{\text{done}} - t_{\text{release}}$ from input release at the declared boundary, before required preparation, until all outputs and required completion checks are available there. Charge setup once across the specified batch.

A *service experiment* uses boundary-arrival and completed-response events per request, including

admission waiting, batching, execution, and delivery. Predeclare the mean or empirical quantile used for T_W , with its estimator. Fix the offered trace, warm-up policy, rejection rules, and treatment of unfinished requests; follow a finite admitted trace through completion or declare a stationary window and censoring rule. Use the same protocol across implementations.

Completed-work survival principle. MKA treats a local improvement as system progress only to the extent that its advantage remains after all changed data movement, state placement, interfaces and conversion, scheduling and queueing, accuracy, reliability, and service constraints are charged within the common boundary around completed useful work. A local speedup or efficiency result that disappears after those terms are included remains a valid component result, but it is not an end-to-end architectural gain. The principle is broader than the scalar MKA Law: it provides a common standard by which device, compiler, runtime, and infrastructure innovations can be compared on the same completed work rather than on incompatible local metrics [1], [2], [26].

3.2. The Proposed MKA Progress Rule

For the protocol's positive energy and time quantities, define

$$D_W = \frac{T_W}{\tau_{\text{ref}}}, \quad \Lambda_W = \frac{1}{E_W D_W} = \frac{\tau_{\text{ref}}}{E_W T_W}, \quad (4)$$

using the same $\tau_{\text{ref}} > 0$ across substrates, conventionally one second. With T_W in seconds and E_W in joules, $D_{\text{eff}} \equiv D_W$ is dimensionless and Λ_W has units J^{-1} . Effective depth is normalized realized time, not hop count. Compute, transfers, conversion, scheduling, and queueing contribute through elapsed time; overlapping durations are not added twice. Publish event timestamps, aggregation code, clock resolution and synchronization error. A single monotonic clock or synchronized clocks must cover both endpoints. The same trace and protocol must reproduce the same D_W ; repeated executions are compared within stated uncertainty.

Gonzalez and Horowitz used the energy–delay product to compare processor energy and performance jointly [27]. Equation (4) is its normalized inverse under a fixed workload contract. The term *MKA Law* denotes the proposed comparison rule; its product identity is algebraic and does not establish an empirical scaling law. For isolated work, $E_W = \bar{P}_W T_W$ weights the denominator as $\bar{P}_W T_W^2$. Energy budgets and deadlines remain constraints.

For two admissible implementations, write

$$\alpha = \frac{E_0}{E_1}, \quad \beta = \frac{T_0}{T_1}, \quad G_\Lambda = \frac{\Lambda_1}{\Lambda_0} = \alpha\beta. \quad (5)$$

The rule accepts $G_\Lambda > 1$ among admissible implementations. The *Compound Gain Regime*

requires $\alpha > 1$ and $\beta > 1$, hence $G_\Lambda > \max(\alpha, \beta)$ regardless of statistical dependence. Throughput and financial value are evaluated separately.

Substitution in Eq. (4) gives $G_\Lambda = (E_0/E_1)(T_0/T_1)$ without an independence assumption. For example, $E_1 = E_0/2$ and $T_1 = 1.2T_0$ give $G_\Lambda = 5/3 > 1$, provided the slower design remains admissible. Simultaneous reductions are sufficient but unnecessary. A 4-J, 4-s design improves the product relative to a 10-J, 2-s baseline but fails a 3-s deadline. Reporting (E_W, T_W) alongside the index preserves this distinction.

For per-request quantities, covariance distinguishes the mean product from the product of means:

$$\mathbb{E}[ET] = \mathbb{E}[E]\mathbb{E}[T] + \text{Cov}(E, T). \quad (6)$$

Service summaries and paired block log ratios serve different estimands; weighting and estimand are fixed before testing.

3.3. Energy Boundaries and Observation Periods

Observed input energy at boundary $\mathcal{B}$ is

$$E_{\text{obs}} = \int_{t_0}^{t_1} P_{\mathcal{B}}(t) dt. \quad (7)$$

For a finite job, $E_W = E_{\text{obs}}$; for a trace followed through completion, $E_W = E_{\text{obs}}/N_{\text{done}}$. Stationary service uses long-run energy per completion. Both alternatives include the same required hosts, optical sources, and board, server, or facility boundary. Programming and calibration are allocated over actual reuse, including preparation before timing.

Disjoint component accounting counts each cost once. Required static power is included; idle-subtracted measurements additionally describe incremental activity.

If fraction f_E of baseline energy improves by factor a_E and all other terms stay unchanged, Amdahl-style accounting gives [26]

$$\alpha_{\text{sys}} = \left(1 - f_E + \frac{f_E}{a_E}\right)^{-1}. \quad (8)$$

For $f_E = 0.2$, $a_E = 10$, the gain is about 1.22. The time analogue requires unchanged, non-overlapping phase fractions; changed critical paths or queues require new timing analysis.

4. A DECODER AND THE LIMITS OF BATCHING

4.1. Arithmetic and State in One Block

Prefill reuses weights across input positions. Decode produces one token per active request from stored keys and values [28]; at low batch, weight reads can dominate the small amount of arithmetic. Each step reads prior KV state, appends new keys and values, and feeds the sampled token into the next step. Finite traces unroll this recurrence.

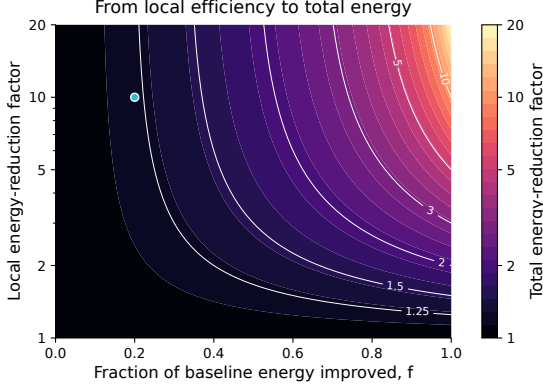


Fig. 3. Total energy gain when fraction f improves and the remainder stays unchanged. At $f = 0.2$, a local factor of 10 gives $1.22\times$ overall. MKA impact: impressive local efficiency becomes important only when it changes a meaningful share of full-task energy.

Count a multiply-add as two FLOPs. Let d be hidden width, f feed-forward width, n_h query heads, and $d_h = d/n_h$. With n_{kv} key/value heads, $d_{kv} = n_{kv}d_h$. GQA shares KV heads across query groups; MQA uses one [29], [30]. Variant comparisons require matched quality.

A GQA block with three-matrix SwiGLU [31] has, excluding biases and normalization,

$$N_{\text{block}} = 2d^2 + 2dd_{kv} + 3df. \quad (9)$$

The terms count query/output, key/value, and SwiGLU matrices. Fixed token arithmetic per layer is $F_0 = 2N_{\text{block}}$; a two-matrix feed-forward layer replaces $3df$ with $2df$.

4.2. Prefill, Decode, and Causality

Score formation and weighted-value accumulation cost approximately $4d$ FLOPs per query-key position pair, giving

$$F_{\text{prefill,layer}} \simeq sF_0 + 4dN_{\text{pair}},$$

$$N_{\text{pair}} = \begin{cases} s^2, & \text{full attention,} \\ s(s+1)/2, & \text{causal attention.} \end{cases} \quad (10)$$

These are useful causal operations; square-then-mask execution or tile padding may add work. FlashAttention reduces memory traffic by tiling exact attention [4].

A decoded token attending to s positions, including itself, requires

$$F_{\text{decode,layer}} \simeq F_0 + 4sd. \quad (11)$$

A full-model count adds softmax, normalization, positional encoding, gating, and $2dV_{\text{vocab}}$ FLOPs for a vocabulary of size V_{vocab} .

Equating the two terms in Eq. (11) gives an arithmetic crossover

$$s_{\times} = d + d_{kv} + \frac{3}{2}f. \quad (12)$$

Table 1
32K DECODER TRAFFIC AND RESIDENT BLOCK WEIGHTS PLUS KV CACHES, BEFORE WORKSPACE AND NON-BLOCK COMPONENTS.

Batch	Read GB/token	FLOP/byte	Resident GiB
1	147.64	1.51	137.5
16	19.29	11.55	287.5
128	11.81	18.87	1407.5

Equal arithmetic contributions do not determine the bandwidth bottleneck, which depends on traffic and hardware.

4.3. The 32K-Context Example

Use $L = 80$, $d = 8192$, $f = 28672$, $n_h = 64$, $d_h = 128$, and $n_{kv} = 8$, representative Llama-like dimensions [32]. The analytical workload assumes 32K context and two bytes per weight and KV element; checkpoint context support and quality still require validation.

Equation (9) yields 68.451 billion block parameters and 136.902 GB of weights, exactly 127.5 GiB. Arithmetic and traffic use this block-only scope; embeddings, output head, and normalization are additional costs.

At $s = 32,768$, fixed arithmetic contributes 136.902 GFLOP and attention 85.899, totaling 222.801 GFLOP per token. Attention's share is 38.6%; crossover occurs at $s_{\times} = 52,224$ (Fig. 4).

For one sequence, KV storage is

$$V_{KV} = 2Lsd_{kv}b_{kv}, \quad (13)$$

where b_{kv} is bytes per element. Per-sequence storage is 10 GiB for GQA, 80 GiB for 64-head MHA, and 1.25 GiB for MQA at fixed remaining dimensions. Appending one token adds 320 KiB across 80 layers before alignment and allocator overhead.

Assume B live sequences, one block-weight read per step, and one full KV read per sequence. With $W_b = 136.902$ GB and $K_b = 10.7374$ GB,

$$V_{\text{read}}(B) \simeq \frac{W_b}{B} + K_b, \quad I(B) = \frac{222.801 \times 10^9}{V_{\text{read}}(B)}. \quad (14)$$

V_{read} is in bytes. Each KV load is reused across its query group; rereads, appends, activations, and output-head traffic add to this estimate.

Weight and KV reads are equal at $B = 12.75$. Further batching mainly reduces the smaller term, approaching 20.75 FLOP/byte while resident memory continues growing (Fig. 5, Table 1).

Batch 128 requires over 1.37 TiB of modeled state. Sharding supplies memory at a communication cost. Quantization, shared prefixes, cache residency, and alternative attention can reduce storage or traffic, subject to quality and execution tests.

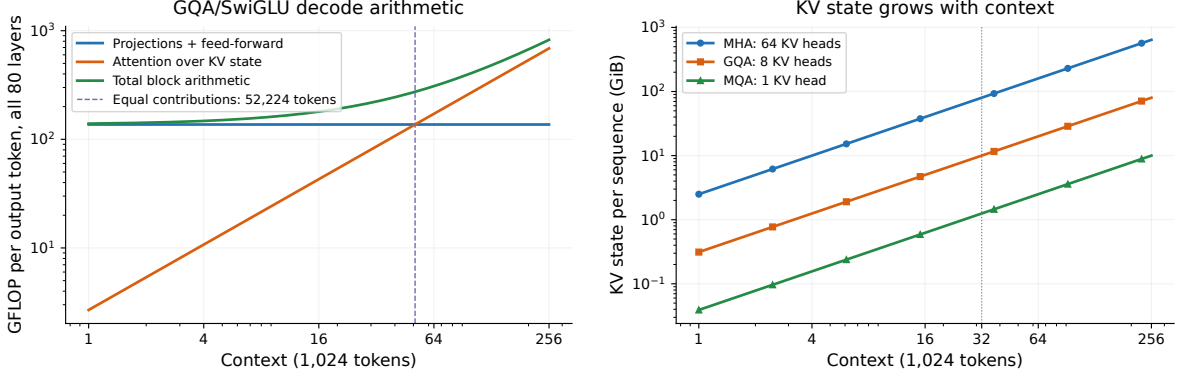


Fig. 4. Decoder with $L = 80$, $d = 8192$, $f = 28672$. Left: GQA/SwiGLU block arithmetic, excluding vocabulary head and elementwise work. Right: KV capacity under three attention layouts at fixed layer and head dimensions. MKA impact: model architecture can move the dominant optimization target from arithmetic to resident state.

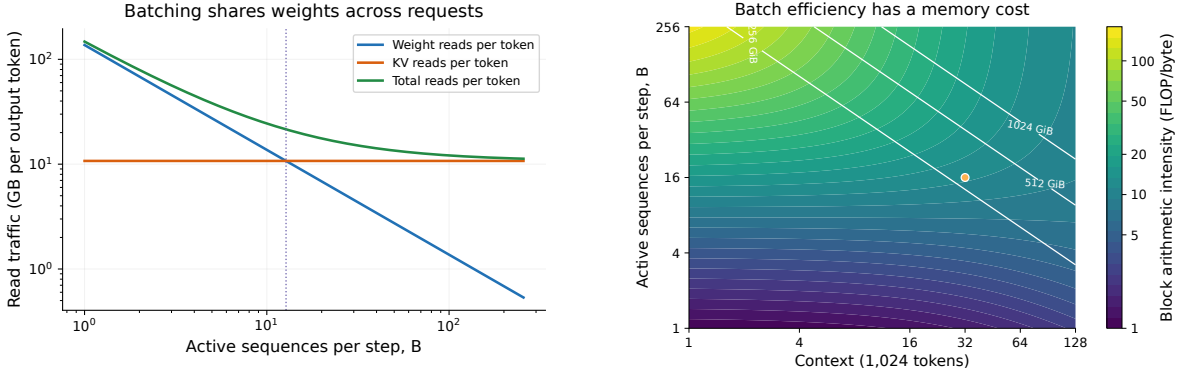


Fig. 5. Weight and KV reads at 32K context, excluding appends, MoE dispatch/return traffic, and scheduling. Batching amortizes weights; each sequence retains its KV reads. Table 1 gives capacity. MKA impact: once weight reuse saturates, KV placement becomes the next system-level target.

Fig. 6. GQA intensity across context and batch. White contours mark resident weights plus KV at 256, 512, and 1024 GiB. At 32K, batch 16 needs 287.5 GiB before workspace and non-block components. MKA impact: the feasible operating point is jointly set by context, batch, and memory capacity.

4.4. Can the Useful Batch Fit?

Let M_{dev} be available device memory, V_{work} reserved workspace, W resident weights, and K KV bytes per sequence. The capacity limit is

$$B_{\text{cap}} = \left\lfloor \frac{M_{\text{dev}} - W - V_{\text{work}}}{K} \right\rfloor. \quad (15)$$

A positive batch needs $M_{\text{dev}} \geq W + V_{\text{work}} + K$; fragmentation, duplication, and placement constraints further reduce capacity.

For target intensity I_* (for example, useful compute capacity divided by memory bandwidth) and F FLOPs per token, $F/(W/B + K) \geq I_*$ requires

$$F > I_* K, \quad B \geq B_{\text{ridge}} = \frac{W}{F/I_* - K}. \quad (16)$$

The target requires $\lceil B_{\text{ridge}} \rceil \leq B_{\text{cap}}$ as well as the intensity ceiling. Network and scheduling impose further limits.

At 32K, 50 FLOP/byte is impossible because the ceiling is 20.75. Reaching 10 FLOP/byte needs $B_{\text{ridge}} \simeq 11.86$, hence 12 sequences and 247.5 GiB. A 256-GiB budget fits at most 12

before reserving workspace, leaving 8.5 GiB for workspace and omitted model components.

Below crossover, batching repays weight reads. Long context favors reducing KV bytes or access cost. Evaluate sharding and new substrates where they change feasible batch and service time.

4.5. Parallel Execution and Request Scheduling

For ring all-reduce of m bytes across p participants, step latency τ_l and bandwidth B_l in bytes/s give

$$T_{\text{ring}} \simeq 2(p-1)\tau_l + \frac{2(p-1)m}{pB_l}. \quad (17)$$

This assumes balanced chunks and uncongested ring links; local reduction adds cost. Tensor-parallel execution depends on the bottleneck along the actual route [33], [34].

For MoE [35], expert-parallel execution includes token fan-out, all-to-all dispatch/return where placement requires it, and router-weighted aggregation before the residual update. Record both transfer directions. Let n_j be assignments to expert j

among E_x equal-rate experts and $\bar{n} = (\sum_j n_j)/E_x$. Imbalance is $\max_j n_j/\bar{n}$. Top- k routing totals k times the input-token count; heterogeneous experts require normalization by service rate. Dropping assignments requires a quality check.

For step time $T_{\text{step}}(B)$ and occupied-slot fraction $u(B)$, delivered rate is approximately $Bu(B)/T_{\text{step}}(B)$. Orca's iteration scheduling and PagedAttention address complementary costs [5], [36]. Larger batches reuse weights but may delay arrivals, so batch choice depends on the service objective.

5. MAPPING COMPUTATION AND STATE

5.1. Workload and Hardware Graphs

In workload graph $G_W = (V, E)$, vertices have compute and state demands. Edge e carries b_e bits on n_e occurrences, totaling $v_e = n_e b_e$. For data-dependent work, an instance is a realized trace; expected traffic supports planning rather than an execution-wise bound.

Hardware graph $H = (N, L)$ specifies supported operations and memory per node, and bandwidth B_l (bits/s) and propagation delay τ_l (seconds) per directed link. Placement $\phi : V \rightarrow N$, ordered directed routes ρ_e , and a schedule assign operations and transfers. A route connects the placed producer to its consumer; repeated transfers are counted in n_e . Node service times and transfer durations are in seconds, energy costs in joules, and normalized timing weights use the common divisor τ_{ref} . Precedence paths refer to the workload; physical routes refer to the hardware. Their bounds omit contention and waiting unless modeled explicitly, whereas realized completion and response times include the delays incurred by the executed schedule.

MKA's tuple $A = (G, T, S, M)$ records workload, logical embedding, substrate classes, and resources per node. Execution additionally specifies H , state ownership, and schedule.

Over period H_t , transfer count $N_e(H_t)$ gives rate $N_e(H_t)/H_t$. A stable homogeneous stream with job rate λ has long-run transfer rate $\lambda \mathbb{E}[n_e]$, hence link utilization

$$u_l = \frac{\lambda \sum_{e:l \in \rho_e} \mathbb{E}[n_e] b_e}{B_l}. \quad (18)$$

Persistent $u_l > 1$ is infeasible. Under ordinary stochastic operation, $u_l < 1$ is necessary for stability but may still yield excessive delay.

5.2. What a First Compiler Must Produce

Each persistent object has an owner, size, update rule, and consumers. Remote consumers create scheduled transfers; representation changes create conversions. Execution traces expose the resulting copies.

For $u = F_1(x)$ and $y = F_2(u)$ supported on one substrate, host-mediated calls may send u out

and back. Keeping both operations and u resident avoids that round trip. At fixed operation costs and V_u intermediate bits,

$$\Delta E_{\text{resident}} = V_u(\varepsilon_{\text{out}} + \varepsilon_{\text{in}} - \varepsilon_{\text{local}}) - \Delta E_{\text{state}} - \Delta E_{\text{control}}. \quad (19)$$

Transfer costs are joules per bit; added state and control include keeping u live. Fusion requires supported operations and precision. A host-only nonlinearity still requires the return trip.

Emit placement, routes, buffers, and a replayable schedule. Verify write-before-read ordering, live capacity, links, and deadlock freedom at common operations, precision, and clocks.

5.3. Communication and Execution Lower Bounds

For a fixed routed instance, two immediate communication bounds are

$$T_{\text{prop}} = \max_e \sum_{l \in \rho_e} \tau_l, \\ T_{\text{cong}} = \max_l \frac{\sum_{e:l \in \rho_e} v_e}{B_l}. \quad (20)$$

The Leighton–Maggs–Rao unit-packet schedule scales with congestion plus dilation [37]. Physical packets, buffers, and arbitration add timing constraints.

For required bits V_C and usable directional bandwidth B_C across hardware cut C , $T_{\text{cut}} = \max_C V_C/B_C$. Compute traffic after legal reduction, caching, replication, and multicast: aggregation can eliminate high-level graph transfers before the cut.

Let T_{cp} bound the precedence chain and T_{work} compatible compute demand. Interchangeable processors use total work divided by aggregate rate; heterogeneous operations require separate capacities. Then

$$T_{\text{LB}} = \max\{T_{\text{cp}}, T_{\text{work}}, T_{\text{cut}}, T_{\text{prop}}, T_{\text{cong}}\}, \quad (21)$$

with $T_{\text{realized}} \geq T_{\text{LB}}$. The index uses realized time. The ratio $T_{\text{realized}}/T_{\text{LB}}$ measures remaining modeled headroom; a large value may reflect the schedule or a loose bound.

5.4. Local Regularity Has a Global Cost

Fixed-density two-dimensional lattices with bounded-range links have diameter of order $\sqrt{N}$: local wiring simplicity entails longer global paths.

A balanced side- q hierarchy uses $O(\log q)$ ancestor hops; longer links and upper-level bandwidth carry physical area–time costs [38].

For symmetric traffic $w_{uv} \geq 0$, let $L_G = D_G - W$, with D_G containing weighted degrees. Assignment $x \in \{-1, 1\}^{|V|}$ has cut weight

$$\text{cut}(x) = \frac{1}{4} x^\top L_G x. \quad (22)$$

Spectral relaxations propose partitions [39], but a small isolated component may fail memory or compute balance. Multilevel partitioning and

constrained refinement can enforce those requirements [40].

Matched area or cost budgets, including wires, routers, and global links, isolate placement from added hardware.

For physical hierarchy, logical hop count must be paired with wire provision. The empirical Rent relation $P(B) = KB^r$ describes external terminal demand for partitions of logic blocks [41]. A two-dimensional region's perimeter grows as $\sqrt{B}$ at fixed density. When the fitted $r > 1/2$, terminal demand outgrows that perimeter, motivating extra routing layers, serialization, or longer links. A topology record should include wire span, bisection bandwidth, router area, and static power alongside graph distance.

5.5. A Controlled Example: Local Exchange Versus All-to-All

Consider an even-sided $q \times q$ mesh, $N = q^2$, with bandwidth B_0 per directed neighbor link. Both patterns below use the same compute work and outgoing budget b bits per node per round.

Local exchange sends $b/4$ bits to each interior neighbor; boundary nodes use at most that budget. A vertical bisection has q links and $qb/4$ bits in either direction, giving constant serialization bound $b/(4B_0)$.

Uniform all-to-all divides b among $N - 1$ destinations. Assume distinct required messages prevent aggregation. The $N^2/4$ pairs crossing in one direction give

$$V_C = \frac{N^2 b}{4(N-1)},$$

$$T_{\text{cut}} = \frac{V_C}{qB_0} = \frac{q^3}{4(q^2-1)} \frac{b}{B_0}. \quad (23)$$

This grows as $\sqrt{N} b/(4B_0)$ at unchanged work and per-node payload because cut bandwidth grows only with side length. The model allows divisible payloads and omits headers; finite packets can increase execution time.

For global payload fraction φ and the remainder sent to neighbors,

$$\frac{T_{\text{cut}}}{b/B_0} = \frac{1-\varphi}{4} + \frac{\varphi q^3}{4(q^2-1)}. \quad (24)$$

On a 128×128 mesh, just 10% global payload raises the cut bound $13.70\times$. A locality advantage can disappear well before traffic becomes predominantly global. Sweeping φ at fixed work tests that boundary; express links and algorithmic reductions are charged for their resources.

5.6. Data Movement and the Active Bottleneck

With disjoint endpoint and link costs, the movement energy of a fixed instance is

$$E_{\text{move}} = \sum_e v_e \left(\varepsilon_{\text{src},e} + \sum_{l \in \rho_e} \varepsilon_l + \varepsilon_{\text{dst},e} \right), \quad (25)$$

Each ε is joules per delivered bit. The model charges copies and endpoint conversion as well as repeated reads; same-domain transfers still consume energy.

For F operations and $\mathcal{V}_k$ bytes through resource k , intensity $I_k = F/\mathcal{V}_k$ gives the multi-resource Roofline bound

$$\frac{F}{T} \leq \min\{P_{\text{peak}}, B_1 I_1, \dots, B_m I_m\}, \quad (26)$$

with B_k in bytes/s [42]. Memory, network, and converters have their own traffic boundaries; a shared intensity axis requires explicit relationships between volumes.

Hong and Kung bound classical dense matrix-multiplication transfers by $\Omega(n^3/\sqrt{M})$ with M fast-memory words in a two-level model [43]. Different algorithms need corresponding bounds. Placement lowers required-transfer costs; algorithms can eliminate transfers. The active Roofline ceiling identifies the useful change.

6. AN EXACT REFERENCE PLANNER AND ITS EVALUATION

6.1. A Bounded Problem With Explicit Costs

We implement serial chains of n operations on k available execution sites. Every accepted site implements the same abstract operation. Its support and memory tests include the operation's workspace and both input and output buffers. A value remains at the producing site until consumed or transferred. Inputs begin at site 0; every plan returns its final result there.

A batch repeats one placement R times, sequentially. Setup enables each used site once per batch; changing kernel or operand work belongs in its recurring cost. Kernel and transfer costs recur, and local transfers have positive cost. All policies provision the same sites and incur the same static power P_s . Let d denote accumulated dynamic and setup energy and t batch duration. Total energy is $E = d + P_s t$. This model fixes overlap and power policy so that the placement comparison is unambiguous.

For operation i at site s , the dynamic cost and time are e_{is}, t_{is} . A transfer from previous owner r costs $c_{i-1}^E(r, s), c_{i-1}^T(r, s)$. If M is the set of already configured sites, the cost increment is

$$v_i(r, s) = (e_{is} + c_{i-1}^E(r, s), t_{is} + c_{i-1}^T(r, s)),$$

$$\Delta_i(r, M; s) = Rv_i(r, s) + \mathbf{1}_{s \notin M}(H_s^E, H_s^T). \quad (27)$$

The initial state is $(r, M) = (0, \{0\})$ with label $(d, t) = (0, 0)$. A transition moves to $(s, M \cup \{s\})$. Unsupported or memory-infeasible choices are rejected before labels are propagated.

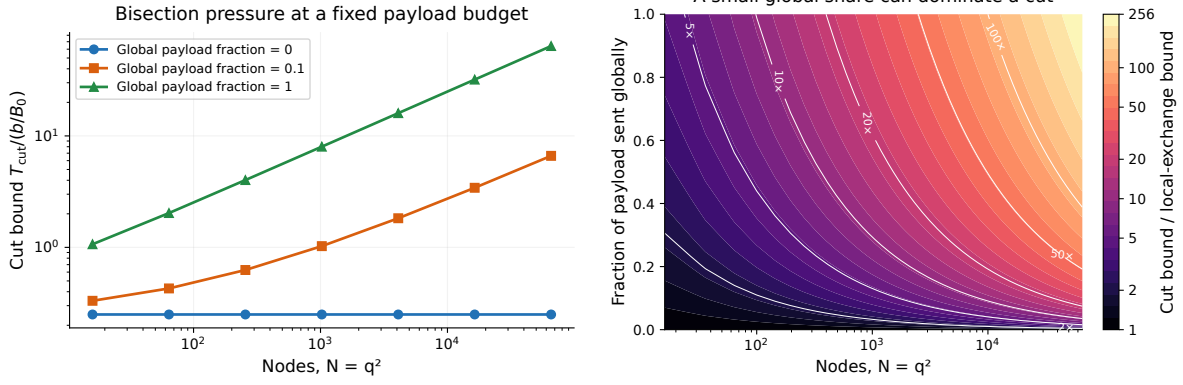


Fig. 7. Bisection bounds at fixed per-node payload and directed-link capacity. Left: local, 10%-global, and uniform all-to-all exchange. Right: the bound relative to local exchange as global fraction grows. Payloads are divisible. MKA impact: even a small global-traffic share can erase locality gains and justify hierarchy or a different interconnect.

6.2. Exactness and the Importance of Retaining State

At each layer and state, retain labels that are nondominated in (d, t) :

$$\mathcal{F}_i(s, M') = \text{ND} \bigcup_{r, M: M' = M \cup \{s\}} \{ \ell + \Delta_i(r, M; s) : \ell \in \mathcal{F}_{i-1}(r, M) \}. \quad (28)$$

After adding the final transfer to site 0, choose the label minimizing $(d + P_s t)t$ subject to the batch deadline. This is a specialization of established multiobjective path methods [25] to the ownership and setup state required here.

Proposition 1 (Exact finite-chain planning). *For the stated serial model with nonnegative additive costs, site-local feasibility, and constant $P_s \geq 0$, the recurrence retains a global energy–time optimum satisfying the deadline whenever one exists.*

Proof. Every legal assignment extends a legal prefix by one allowed transition. Two prefixes in the same layer, owner, and configured-set state admit the same suffixes, with identical suffix costs. If one prefix has no larger d or t , every common completion preserves that ordering. The objective $(d + P_s t)t$ is nondecreasing in both quantities, and a time deadline also preserves dominance. Induction therefore permits each pruning step without losing an optimum. The terminal transfer and final selection complete the argument. $\square$

The configured-set state is essential: two prefixes with the same current owner can have different future setup costs. Pruning across those states would be unsound. General DAGs require additional state for live objects, joins, and resource overlap. The number of nondominated labels can grow exponentially; the recurrence offers exact finite-instance results rather than a polynomial bound for general mapping.

One frontier can also serve several fixed-power operating points. For plans a, b with dynamic energies d_a, d_b and durations t_a, t_b , their product difference, with $\Psi_a(P_s) = (d_a + P_s t_a)t_a$, is

$$\Psi_a(P_s) - \Psi_b(P_s) = d_a t_a - d_b t_b + P_s (t_a^2 - t_b^2). \quad (29)$$

Thus the preferred plan can change as persistent power increases. The retained (d, t) frontier supports that change by rescaling labels, without repeating the search.

6.3. Protocol and Comparison Policies

The released Cartesian suite has four families: uniform site preferences, alternating preferences, irregular support and costs, and tighter local memory. Each uses $n \in \{8, 16, 32\}$, inter-site cost scales $\{1, 4, 16\}$, reuse $R \in \{1, 16, 256\}$, and five deterministic seeds, giving 540 cases. Setup energy is $(0, 1800, 2500)$ and setup time $(0, 180, 250)$ in normalized units; $P_s = 1$. Base operation energy is drawn uniformly from integers 100–900 and duration from 20–140. Site multipliers vary by family. Standard memory capacities are $(10^6, 24, 32)$ storage blocks, reduced to $(10^6, 12, 16)$ in the capacity family. The complete generator, transfer matrices, and every generated instance are included. These normalized costs are synthetic and are not fitted to a particular device.

The deadline is 1.25 times the all-host duration, so a legal comparator always exists. We compare the joint planner with all-host execution, the best feasible fixed site, globally energy-optimal and time-optimal plans, and a greedy planner minimizing the full incremental energy–time product at each step. A stronger comparator chooses the best complete product among the two single-objective optima and the best fixed site. A boundary-blind ablation uses the exact planner with transfer costs omitted during selection, then evaluates the chosen plan under the full model. Every case, including ties and failures, is retained.

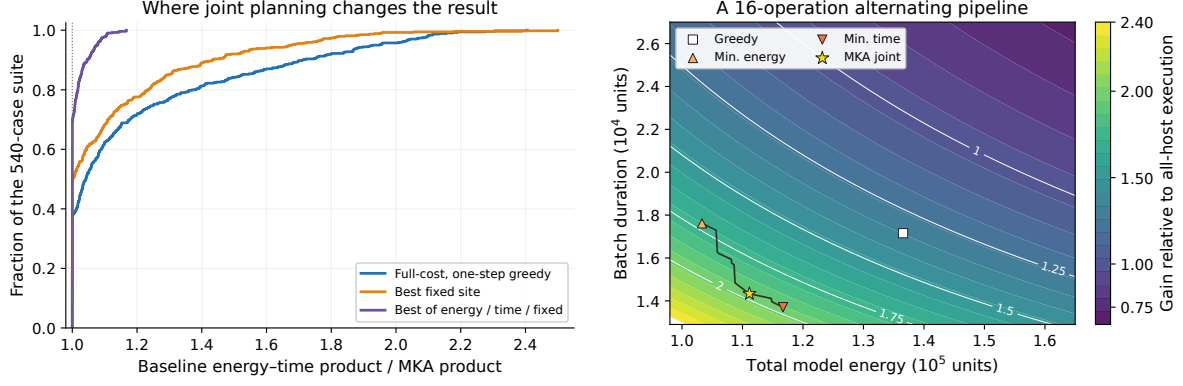


Fig. 8. Computed reference-planner results. Left: gain distributions across all 540 cases; the strong energy/time/fixed comparator often matches the joint optimum. Right: the specified alternating case with $n = 16$, link scale 4, $R = 16$, seed 0. The curve is its feasible energy-time frontier, and the background gives product gain against all-host execution. MKA impact: complete accounting avoids false local optima and recovers feasible energy-time gains.

Table 2
COMPUTED MODEL-LEVEL GAIN: BASELINE PRODUCT DIVIDED BY THE JOINT PLANNER’S PRODUCT. GEOMETRIC MEANS INCLUDE TIES. THE BOUNDARY-BLIND ROW USES ONLY ITS FEASIBLE CASES; DEADLINE FAILURES ARE COUNTED SEPARATELY.

Comparator	Feasible	Geo.	> 1.1	Max.
All host	540	1.428	315	3.502
Best fixed site	540	1.110	172	2.500
Energy optimum	540	1.035	62	1.603
Time optimum	540	1.050	98	1.454
Full-cost greedy	540	1.166	204	2.406
Best energy/time/fixed	540	1.011	12	1.167
Boundary-blind	351	1.174	126	2.360

6.4. Observed Planning Gains

The joint planner improves the product over full-cost greedy in 338 cases, including 204 gains above 10%, and ties it in 202. Its geometric-mean energy and time factors are 1.080 and 1.079, respectively; both improve in 234 cases. Against the best fixed site, both improve in 192 cases. These are placement and amortization gains on the same modeled resources.

The stronger comparator leaves a much smaller gap: 374 ties and 166 strict improvements, with a geometric-mean factor of 1.011. Most of the benefit comes from accounting for transfers and coordinating decisions across operations. Retaining both objectives then resolves a smaller set of trade-offs. The comparison does not establish an advantage over another exact optimizer given the same model and objective.

Boundary-blind selection violates the deadline in 189 cases, or 35%. In the example in Fig. 8, it alternates sites frequently and takes 26,030 time units, exceeding the 24,400-unit deadline. The joint plan uses site 2 for ten operations and site 1 for six,

taking 14,334 units. Its product gain over greedy is 1.470, while the stronger comparator is already within 0.4% of it. A deadline sweep at 1.0, 1.5, and 2.0 times all-host duration produces 261, 144, and 111 boundary-blind failures, respectively; the full results accompany the main grid.

A separate enumerator sums primitive costs for all 3^n assignments on 36 instances with $n \in \{4, 6, 8\}$. Its complete nondominated sets match the recurrence in every case, covering 88,452 assignments before feasibility filtering. All 4,320 selected-plan replays preserve an exact modular-affine test computation; these are ownership and implementation checks, separate from the synthetic cost inputs. The largest retained layer in the main suite contains 518 labels.

6.5. How Much Model Error Would Erase a Gain?

Suppose each plan’s true total energy and time lie within relative bounds $\epsilon_E, \epsilon_T < 1$ around its estimates. For estimated gain $\hat{G}$, the worst-case ratio satisfies

$$G \geq \hat{G} \frac{1 - \epsilon_E}{1 + \epsilon_E} \frac{1 - \epsilon_T}{1 + \epsilon_T}. \quad (30)$$

The result follows by decreasing both numerator quantities and increasing both denominator quantities to their allowed extremes. With separate 10% bounds, a gain must exceed 1.494 to remain above one. Eighty-six greedy comparisons pass that margin. The stronger comparator’s gains do not. Applying this certificate to hardware requires calibrated bounds and a separate robust feasibility check for the deadline and memory constraints.

7. CHOOSING DESIGNS UNDER UNCERTAINTY

7.1. Ranking Feasible Mappings

For placement, routes, schedule, and substrate assignments x under scenario ξ , minimize

$$J(x, \xi) = \log \frac{E(x, \xi)}{E_{\text{ref}}} + \log \frac{T(x, \xi)}{T_{\text{ref}}}. \quad (31)$$

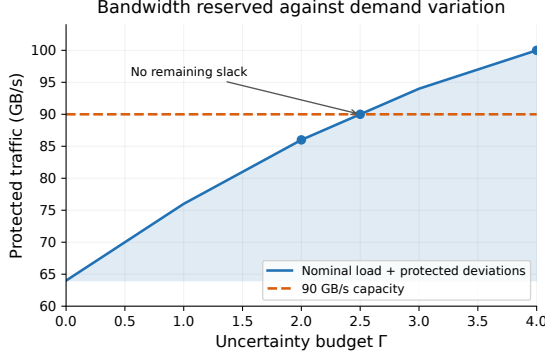


Fig. 9. Budgeted bandwidth protection. Fractional Γ permits a fractional deviation in the next-largest coefficient; capacity 90 GB/s is reached at $\Gamma = 2.5$. MKA impact: uncertainty becomes an explicit capacity-versus-efficiency decision instead of hidden safety margin.

This minimizes the energy–delay product under hard memory, quality, and timing constraints. Monetary cost can be a separate constraint or an explicitly normalized objective.

Group communicating operations, route candidates, and refine after scheduling. Ownership exposes copies; held-out traces test the combinatorial search’s ranking model.

7.2. Bertsimas: Reserve Capacity for Uncertain Demand

Longer contexts or changed routes can overload a trace-fitted placement. Bertsimas and collaborators make protected uncertainty explicit [44], [45]:

$$\begin{aligned} \min_{x \in \mathcal{X}} \quad & \sup_{\xi \in \mathcal{U}} J(x, \xi), \\ \text{subject to} \quad & g_i(x, \xi) \leq 0 \quad \text{for all } \xi \in \mathcal{U}. \end{aligned} \quad (32)$$

The predeclared set $\mathcal{U}$ covers operating variation against which every constraint must hold.

For uncertain coefficients $a_j = \bar{a}_j + \hat{a}_j u_j$, let $\hat{a}_j \geq 0$, $|u_j| \leq 1$, and $\sum_j |u_j| \leq \Gamma$. For nonnegative assignments x_j , the exact robust counterpart of $\sum_j a_j x_j \leq b$ is

$$\begin{aligned} \sum_j \bar{a}_j x_j + \Gamma z + \sum_j p_j &\leq b, \\ z + p_j &\geq \hat{a}_j x_j, \quad z, p_j \geq 0. \end{aligned} \quad (33)$$

The Bertsimas–Sim capacity subproblem remains linear within the combinatorial search [44].

Four flows totaling 64 GB/s have possible increases of 12, 10, 8, and 6 GB/s. Protected loads at $\Gamma = 2, 2.5, 4$ are 86, 90, and 100 GB/s. A 90-GB/s link supports $\Gamma = 2.5$ with no slack; latency generally requires more headroom (Fig. 9).

An overload probability additionally requires distributional assumptions or testing [46]. Evaluate protection’s capacity or latency cost against held-out overload rates; extra reservation has little value once service requirements are met.

The nominal price paid for protection can be recorded as

$$\text{PoR}(\Gamma) = J(x_\Gamma, \bar{\xi}) - J(x_0, \bar{\xi}). \quad (34)$$

It is nonnegative when x_0 is a global nominal optimum and the robust feasible set is contained in the nominal one. Reporting this cost with held-out violations makes the reservation decision interpretable.

Context-dependent decisions [47] can choose batches from the current queue and memory. Tests must exclude future information and capture how policy changes feed back into queueing and observed service.

7.3. Rus: Generate Structures With Explicit Legality Checks

Daniela Rus’s RoboGrammar generates robot designs through graph rules [48]; its multi-objective extension searches design and control jointly [49]. Decentralized reconfiguration links abstract rules to physical mechanisms [50]. We adapt this pattern to computing fabrics.

Rules $\mathcal{R}_S$ generate hardware from seed H_0 . Accepted candidates also pass verification:

$$\mathcal{H}_S^{\text{legal}} = \{H : H_0 \Rightarrow_{\mathcal{R}_S}^* H, \text{Verify}_S(H) = 1\}. \quad (35)$$

Rules enforce ports and local connections; verification checks global timing, routing, thermal feasibility, and deadlock freedom.

Portability preserves required input–output behavior across hardware graphs and schedules. Analog behavior includes error tolerance; quantum interfaces enforce restrictions on copying unknown states.

Low-energy and higher-headroom candidates can both be useful; the contract selects among their operating margins.

A distributed realization also needs local control. A node updates from its own state, observations, and permitted neighbors:

$$s_i(t+1) = F_i(s_i(t), \{s_j(t) : j \in N_i\}, o_i(t)). \quad (36)$$

For a buffered dataflow implementation, credits limit sends to available capacity, ownership identifies the valid copy, and a configuration epoch prevents mixing old and new routes. The verifier must establish these invariants for the actual protocol. Control messages, polling, and barriers consume energy and appear in the execution trace. This protocol remains a target for a distributed backend.

7.4. Reconfiguration Has a Minimum Useful Duration

If switching costs E_{sw} joules and saves $\Delta P > 0$ at unchanged work rate for H seconds, energy improves when

$$H \Delta P > E_{\text{sw}}. \quad (37)$$

Switching costs include state movement and calibration, with timing constraints enforced during interruptions. Frequent switches may consume the entire saving.

Adaptation is evaluated against a validated static baseline on held-out demand, with ownership and legality preserved through each transition.

8. SUBSTRATE CHOICES AND THEIR BREAK-EVEN CONDITIONS

Substrates act on different parts of the graph (Table 3). Analog in-memory arrays combine computation and storage; photonics can connect several compute stages. Recurring interface costs often decide which choice pays.

8.1. A Common Boundary-Cost Test

For a behavior-preserving replacement of U , let S_U be internal energy saved per invocation, $\Delta E_{\partial U}$ the signed change in boundary communication and conversion, H_{per} added recurring overhead, and H_{setup} setup reused R times. Energy improves exactly when

$$S_U > \Delta E_{\partial U} + H_{\text{per}} + \frac{H_{\text{setup}}}{R}. \quad (38)$$

Terms refer to identical completed work within this additive model; unchanged costs cancel. This is an energy break-even test. Timing, accuracy, reliability, and resource constraints still determine admissibility and the full energy–time comparison.

Conversion recurs unless residency or an eliminated crossing avoids it. Programming and calibration amortize over reuse; ordinary same-domain communication remains charged.

With recurring margin $m_E = S_U - \Delta E_{\partial U} - H_{\text{per}} > 0$, reuse must satisfy

$$R > \frac{H_{\text{setup}}}{m_E}. \quad (39)$$

For $m_E \leq 0$ and nonnegative setup, no reuse count gives strict improvement.

At normalized baseline one and setup 12, recurring cost 0.24 first wins at $R = 16$; cost 0.60 needs $R > 30$; cost 1.10 never wins. Separating core cost 0.24 from boundary cost c (Fig. 10), required reuse diverges as $c \rightarrow 0.76$. This threshold follows from the chosen 0.24 core cost; it is not a universal analog limit.

A faster operation may expose another critical path or create a shared-tile queue. Sequential time savings must exceed added transfer, conversion, and waiting; concurrency requires a new schedule.

8.2. Joint Energy and Time Break-Even for a Reusable Batch

Suppose a baseline performs R sequential solves at e_0 joules and t_0 seconds each. A candidate uses e_1, t_1 plus batch setup H_E, H_T .

With identical outputs, no overlap, and setup fully charged,

$$G_\Lambda(R) = \frac{R^2 e_0 t_0}{(H_E + R e_1)(H_T + R t_1)}. \quad (40)$$

Set $a = e_0 t_0 - e_1 t_1$, $b = H_E t_1 + H_T e_1$, and $c = H_E H_T$. For $a > 0$, the product improves when

$$R > R_\Lambda^* = \frac{b + \sqrt{b^2 + 4ac}}{2a}. \quad (41)$$

This solves $aR^2 - bR - c > 0$. If $a \leq 0$ and setup is nonnegative, improvement is impossible. Compound gain additionally needs $e_1 < e_0$, $t_1 < t_0$, and

$$R > \max \left\{ \frac{H_E}{e_0 - e_1}, \frac{H_T}{t_0 - t_1} \right\}. \quad (42)$$

For assumed $e_0 = 1$ J, $t_0 = 1$ s, $e_1 = 0.24$ J, $t_1 = 0.6$ s, $H_E = 12$ J, and $H_T = 4$ s, the first integer product gain is $R = 14$; both axes improve at $R = 16$.

8.3. Stage I: Spatial Digital Execution

A field-programmable gate array (FPGA) or coarse-grained reconfigurable array (CGRA) supports connected stages and local state. With frequency f_i , initiation interval II_i cycles, and replicas u_i , stage i has rate

$$r_i \leq \frac{u_i f_i}{II_i}. \quad (43)$$

The slowest stage bounds a single-rate pipeline. Multirate graphs first normalize to a common output. Replication helps until another stage, memory port, or link limits execution.

At matched operations and precision, charge configuration, fill, transfers, and execution over reuse [11], [16]. Compilation is separate unless on-line. Routing and buffers penalize irregular graphs; setup penalizes small jobs.

8.4. Stage II: Analog Operations and Their Numerical Limits

Ambrogio et al.'s analog chip contains 35 million phase-change-memory devices across 34 tiles and reaches 12.4 TOPS/W chip-sustained performance [51]. Full-system comparisons also include its auxiliary operations and off-chip support.

Abdelaleim, Prakash, Irfanoglu, and Milutinović propose a resistive symmetric positive-definite (SPD) solver [52]. Version 2, dated 24 January 2026, reports analytical derivations and SPICE simulations, including circuit nonidealities. It distinguishes a passive subclass from a general active negative-resistance construction. These are circuit-model results, not measurements of fabricated solver hardware. The claimed size-independent analog phase is conditional on the circuit, matrix spectrum, and permitted component scaling. Full workload evaluation additionally charges programming, input/output, calibration, energy, component

Table 3
SIX SUBSTRATE REGIMES: INTENDED BENEFIT, EVIDENCE AVAILABLE, AND MATCHED SYSTEM-LEVEL TEST. EVIDENCE DESCRIBES THE CITED TECHNOLOGY, NOT A FABRICATED MKA SYSTEM.

	Implementation	Intended benefit	Available evidence	Matched system-level test
I	Reconfigurable digital fabric	Retain operations and intermediate state in a spatial pipeline.	Implemented dataflow precedents; MKA backend proposed.	Same work, precision and resources; include host, routing and configuration.
II	Analog computation	Use summation or relaxation for suitable numerical operations.	Analog-AI chip measurements; cited SPD solver is analytical/SPICE.	Same accuracy; include programming, readout, calibration and settling.
III	Processing near memory	Reduce transfers of stored weights or graph state.	Published PIM hardware precedents.	Saved traffic must exceed peripherals, updates and coordination.
IV	Photonic links	Add capacity at communication cuts.	Measured components and limited complete links.	Same delivered traffic; include source, tuning, interfaces and actual load.
V	Quantum-dot devices	Execute local quantum operations with shared control.	Small-array device demonstrations.	Same quantum task and fidelity; account for control and cooling.
VI	Entanglement-assisted links	Connect quantum states across modules.	Protocol theory and quantum-network precedents.	Same task and fidelity; include failed preparation and classical signaling.

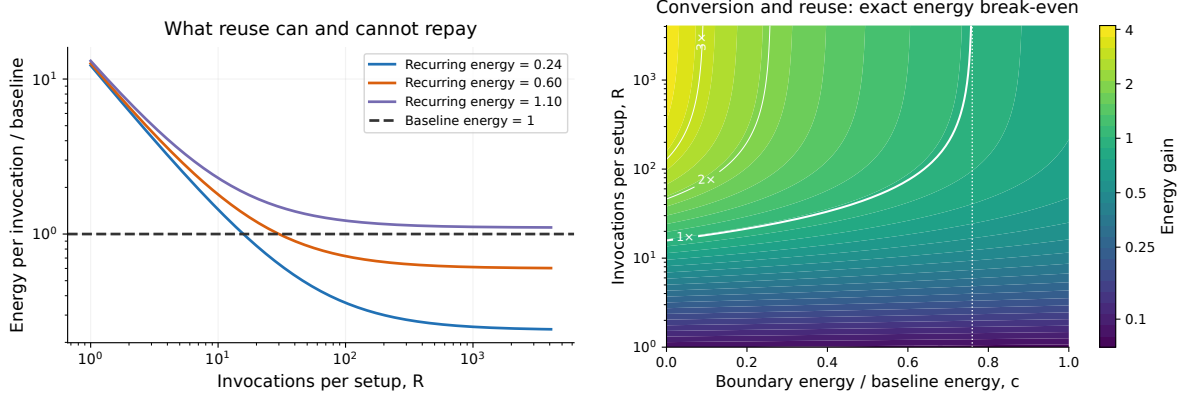


Fig. 10. Left: $E(R) = e_\infty + 12/R$ at three recurring costs. Right: $G_E = 1/(0.24 + c + 12/R)$ with per-invocation boundary cost c . At $c \geq 0.76$, no reuse count gives strict gain. MKA impact: the break-even boundary shows when a new substrate can repay conversion and setup at all.

count, accuracy, and conditioning. Dense instances require quadratically many coefficient settings or equivalent resources. These dependencies motivate a matched-tolerance comparison; they do not establish $O(1)$ end-to-end time for a complete digital-input/digital-output solve, unbounded gains, a computational-complexity-class transition, or universal superiority over digital execution.

The spectral requirement can be seen in an idealized relaxation model, analyzed separately from that circuit proposal:

$$\dot{x} = -\omega K(Ax - b), \quad A \succ 0, K \succ 0. \quad (44)$$

Scaled matrices A, K are dimensionless SPD and $\omega > 0$ has units s^{-1} . At equilibrium $x^* = A^{-1}b$.

With $z = K^{-1/2}(x - x^*)$ and $S = K^{1/2}AK^{1/2}$, $\dot{z} = -\omega Sz$ gives

$$\|z(t)\|_2 \leq e^{-\omega \lambda_{\min}(S)t} \|z(0)\|_2. \quad (45)$$

A sufficient run duration for a relative reduction $\epsilon \in (0, 1)$ is

$$t \geq \frac{\log(1/\epsilon)}{\omega \lambda_{\min}(S)}. \quad (46)$$

Symmetry in transformed coordinates gives this bound; general non-normal dynamics can transiently grow.

The smallest eigenvalue in Eq. (46) can depend on dimension. For $K = I$ and $A_n =$

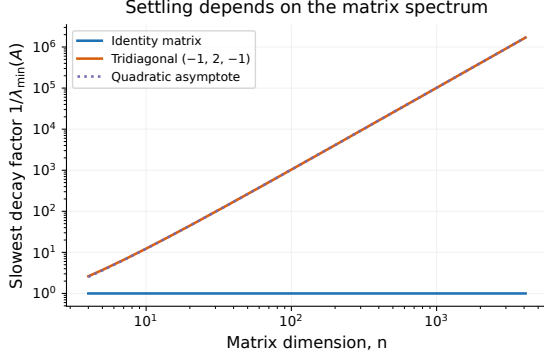


Fig. 11. Relaxation with $K = I$: the plotted factor multiplies $\log(1/\epsilon)/\omega$. The smallest eigenvalue determines dimension dependence. MKA impact: analog advantage depends on the matrix spectrum and requested accuracy, not device count alone.

$\text{tridiag}(-1, 2, -1)$,

$$\lambda_{\min}(A_n) = 4 \sin^2 \frac{\pi}{2(n+1)} \sim \frac{\pi^2}{(n+1)^2}. \quad (47)$$

This family settles quadratically in n ; $A_n = I$ remains constant (Fig. 11). Comparisons include preconditioner construction and realization at fixed tolerance and permitted component scaling.

For $(A + \Delta A)(x + \Delta x) = b + \Delta b$, set $\delta_A = \|\Delta A\|/\|A\|$ and $\delta_b = \|\Delta b\|/\|b\|$. With nonzero b , compatible induced norms, and $\kappa(A)\delta_A < 1$,

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\delta_A}(\delta_A + \delta_b). \quad (48)$$

Conditioning amplifies component error [53]. Residual $r = b - A\tilde{x}$ gives $\|x - \tilde{x}\|/\|x\| \leq \kappa(A)\|r\|/\|b\|$, so report residual and conditioning together. Digital residual checks or refinement add their time and energy.

For a digital-input/digital-output analog solve,

$$E_{\text{analog}} = E_{\text{core}} + E_{\text{in}} + E_{\text{out}} + E_{\text{control}} + \frac{E_{\text{program}} + E_{\text{calibrate}}}{R}. \quad (49)$$

Inputs may use DACs or programmed conductances. Core energy includes static power during settling; readout returns every requested value through its finite interface.

Regularized least squares and covariance systems are suitable tests. Laplacians need a gauge or regularization to become nonsingular SPD; Newton Hessians require a positive-definiteness check.

8.5. Stage III: Processing Near Stored State

Processing-in-memory (PIM), exemplified by Aquabolt-XL [54], saves transfers at the cost of work near storage.

Eliminating ΔV off-chip bytes at baseline cost ϵ_{off} joules/byte gives net saving

$$\Delta E = \Delta V \epsilon_{\text{off}} + \Delta E_{\text{other, saved}} - \Delta E_{\text{added}}. \quad (50)$$

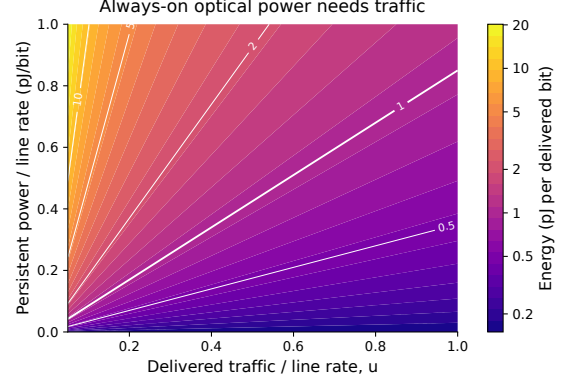


Fig. 12. Optical energy with assumed variable cost 0.15 pJ/bit. The 1-pJ/bit contour gives electrical break-even; persistent power is allocated over delivered bits. MKA impact: photonics becomes valuable only when traffic utilization is high enough to repay persistent optical power.

Added costs include peripherals, updates, conversion, and coherence. Require $\Delta E > 0$ together with bank bandwidth, host coordination, and precision feasibility.

Weights are reused; KV is appended and read by request and head. Expensive writes may favor weights over KV, requiring separate placement.

8.6. Stage IV: Complete Photonic Links

Optics offers bandwidth density and distance-dependent transfer costs [55]. Daudlin et al. report 80-channel transmitter/receiver arrays, 800-Gb/s aggregate capability, and 5.3 Tb/s/mm². Separate front ends consume 50/70 fJ/bit at 10 Gb/s; the complete-link experiment uses four channels at 8 Gb/s [56]. System models retain those distinct measurement scopes.

For line rate R_l , utilization u , variable energy e_{var} per bit, and persistent power P_{on} ,

$$e_{\text{opt}}(u) = e_{\text{var}} + \frac{P_{\text{on}}}{uR_l}. \quad (51)$$

Persistent power covers the active source, receiver bias, tuning, and control. Full-rate front-end energy may contain static bias; gating adds wake-up costs.

Against an electrical comparator with energy $e_{\text{elec}} > e_{\text{var}}$, the energy break-even condition in this model is

$$u > u^* = \frac{P_{\text{on}}}{R_l(e_{\text{elec}} - e_{\text{var}})}. \quad (52)$$

If $u^* \geq 1$, full utilization still fails. At assumed $e_{\text{var}} = 0.15$ pJ/bit, $P_{\text{on}}/R_l = 0.30$ pJ/bit, and a 1-pJ/bit comparator, $u > 0.353$ is required (Fig. 12). Bursts and power gating alter this through wake-up costs.

Target a measured electrical cut bottleneck: a busy package link may repay optics where a local uncongested wire cannot. Include partitioning that avoids cut traffic as an alternative.

8.7. Stages V and VI: Quantum-Native Research Horizons

Semiconductor quantum dots support local quantum states and interactions [57], [58]. Borsoi et al.'s 16-dot germanium array uses shared control with terminal scaling

$$N_{\text{terminals}} = 6\sqrt{g} - 1 \quad (53)$$

for that architecture [59], giving 23 terminals at $g = 16$. Scaling also requires fidelity and cooling measurements.

Shared control limits independently adjustable parameters. Cross-capacitance maps, including rectangular or rank-deficient maps, support calibration. Useful quantum graphs require sufficient control and readout with calibration and cryogenic costs included.

Stage VI connects modules through teleportation using entanglement and classical communication, subject to no-cloning and causality [60]. Generation and management of entanglement are explicit resources [61], [62].

Let two independent links succeed with probability p per synchronous trial of duration τ , storing successes without decoherence. A swap succeeds with probability p_s ; failure restarts both links. Then

$$\mathbb{E}[T_2] = \frac{\tau}{p_s} \frac{3 - 2p}{p(2 - p)}. \quad (54)$$

Independent geometric trials give $\mathbb{E}[\max(X_1, X_2)] = 2/p - 1/[p(2 - p)]$; $1/p_s$ counts swap cycles. For $p \ll 1$, the mean approaches $3\tau/(2pp_s)$. Swap duration, finite memory, and signaling add costs or change the model.

Quantum comparisons fix task and fidelity, then charge preparation failures, control, and transfer time.

9. ACCURACY AND PHYSICAL LIMITS

9.1. Accuracy at the Output

Let δ_v bound error at vertex v , ϵ_v locally introduced error, and L_{vu} sensitivity to predecessor u over the operating region. Then

$$\delta_v \leq \epsilon_v + \sum_{u \in \text{pred}(v)} L_{vu} \delta_u. \quad (55)$$

Sum all DAG paths, including reconvergence, to allocate precision where sensitivity is high.

Bounds must cover reached perturbations. Cycles need convergence or stability analysis, correlated errors need a shared-source model, and nonlinear discrete outputs require application tests.

The Walden relation $P_{\text{ADC}} = F_W 2^{\text{ENOB}} f_s$ links converter power to effective bits, sampling rate, and implementation-dependent F_W [63], [64]. Precision changes require corresponding converter characteristics.

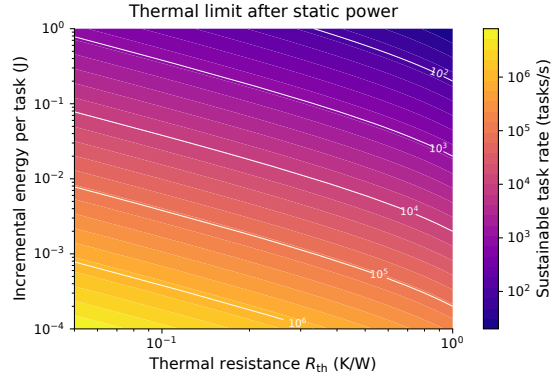


Fig. 13. Thermal rate bound at 40-K allowance and 20-W static power, as resistance and incremental task energy vary. MKA impact: lower energy per task converts directly into sustainable throughput and cooling headroom.

9.2. Failures and Retries

For failure events F_i , the union bound gives

$$\Pr\left(\bigcup_i F_i\right) \leq \sum_i \Pr(F_i). \quad (56)$$

The bound allows dependence; multiplying success probabilities requires independence. Common voltage, temperature, or calibration can violate it. Count faults and retries.

Record numerical checks, communication errors, and retry policy; charge redundancy and retries to energy and time. Quantum error control respects no-cloning.

9.3. Thermal Headroom and Sustainable Throughput

A standard compact thermal model writes

$$C_{\text{th}} \dot{\theta} + G_{\text{th}} \theta = P(t), \quad (57)$$

where θ is rise above ambient. Package-specific resistance-capacitance models support local thermal analysis [65].

For thermal resistance R_{th} , allowed rise ΔT , static power P_s , and incremental task energy e , steady rate r satisfies

$$P_s + re \leq \frac{\Delta T}{R_{\text{th}}}. \quad (58)$$

A positive sustainable rate requires $P_s < \Delta T/R_{\text{th}}$. For the feasible case,

$$r_{\text{max}} = \frac{\Delta T/R_{\text{th}} - P_s}{e}. \quad (59)$$

At assumed $\Delta T = 40$ K, $R_{\text{th}} = 0.5$ K/W, and $P_s = 20$ W, 60 W remains dynamic. A 1-mJ task permits at most 60,000 tasks/s; halving energy doubles this bound at unchanged parameters.

At temperature T , resetting an unbiased unknown bit irreversibly under the usual isothermal assumptions has the Landauer dissipation bound $k_B T \ln 2$ [66]. An implementation erasing N_{irr} such

bits therefore has the conditional floor $N_{\text{itr}} k_B T \ln 2$. The erasure count must be justified for the actual algorithm and implementation. Arithmetic operations, FLOPs, and attention pairs are not automatically irreversible bit erasures. Bennett's reversible-computation result makes this distinction explicit [67]. We infer no universal MKA ceiling or violation of the second law from an arithmetic count. A bound on Λ_W would require both a justified erasure lower bound and a compatible completion-time lower bound.

10. EXECUTION SPEED, SERVICE, AND POWER

10.1. Harchol-Balter's Operational Perspective

Harchol-Balter's queueing research connects execution to waiting, variable jobs, parallelism, and response tails [68], [69]. Placement changes the service process managed by the runtime.

For one FCFS server with Poisson arrival rate λ and independent service requirements X , assume $m_1 = \mathbb{E}[X] > 0$, $m_2 = \mathbb{E}[X^2] < \infty$, and $\rho = \lambda m_1 < 1$. Pollaczek–Khinchine gives

$$\mathbb{E}[T_0] = m_1 + \frac{\lambda m_2}{2(1-\rho)}. \quad (60)$$

Long jobs delay those behind them; equal mean service can therefore yield different waiting times.

Scaling service to X/a , $a > 1$, at unchanged arrivals and policy gives

$$\mathbb{E}[T_a] = \frac{m_1}{a} + \frac{\lambda m_2}{2a(a-\rho)}. \quad (61)$$

Because $a - \rho > 1 - \rho$, $a\mathbb{E}[T_a] < \mathbb{E}[T_0]$ whenever $\lambda m_2 > 0$. Faster service also reduces waiting, amplifying mean response gain.

For exponential service times, this simplifies to

$$\frac{\mathbb{E}[T_0]}{\mathbb{E}[T_a]} = \frac{a-\rho}{1-\rho}. \quad (62)$$

At $\rho = 0.8$, $1.5\times$ acceleration gives $3.5\times$ mean response gain. Batching and multiple servers require their dependency models; tails are separate. Infinite second moment permits infinite mean waiting below saturation.

10.2. Power Cost of Service Acceleration

Harchol-Balter and collaborators jointly evaluate power and response time [70], [71]. At stable fixed admitted rate λ , measured energy per completion is $\bar{E} = \bar{P}/\lambda$, giving

$$\Lambda_{\text{service}} = \frac{\lambda \tau_{\text{ref}}}{\bar{P} \mathbb{E}[T]}. \quad (63)$$

At fixed λ , this minimizes the established power–response-time product [71], allocating observation-period energy per completion.

For power ratio $\pi_P = \bar{P}_1/\bar{P}_0$ at unchanged admitted rate, the M/M/1 case gives

$$G_{\text{service}} = \frac{a-\rho}{\pi_P(1-\rho)}. \quad (64)$$

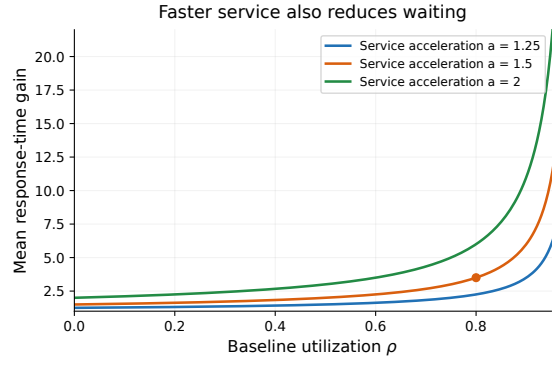


Fig. 14. Exact M/M/1 response gain at fixed arrival rate, uniform service acceleration, and baseline utilization up to 0.95. MKA impact: reducing service time near saturation can create superlinear user-visible latency gains.

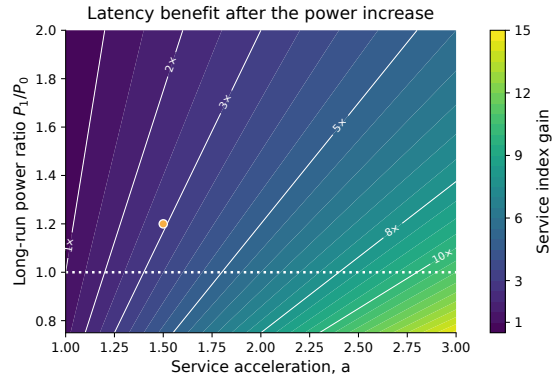


Fig. 15. Service gain at $\rho = 0.8$; dotted line marks unchanged power. At $a = 1.5$, $\pi_P = 1.2$, index gain is $2.92\times$ with higher request energy. MKA impact: the preferred operating point balances power and latency instead of maximizing either quantity alone.

At $\rho = 0.8$, $a = 1.5$, and $\pi_P = 1.2$, the index improves $2.92\times$: response improves $3.5\times$ while request energy rises 20%. A strict energy budget can reject this otherwise favorable trade-off.

10.3. A Latency Target Limits Usable Capacity

For FCFS M/M/1 service rate μ and $\lambda < \mu$, the stationary response tail is

$$\Pr\{T > \ell\} = e^{-(\mu-\lambda)\ell}. \quad (65)$$

Conditioning on the geometric queue length and mixing the resulting Erlang service sums yields rate $\mu - \lambda$ [68].

Requiring $\Pr\{T \leq \ell\} \geq p$ gives

$$\lambda \leq \lambda_{\text{max}} = \mu - \frac{\log[1/(1-p)]}{\ell}. \quad (66)$$

A nonpositive bound admits no positive arrival rate. With $\mu = 100$ requests/s, deadline 100 ms, and $p = 0.99$, capacity is 53.95 requests/s; at $\mu = 150$, it is 103.95. Capacity expansion and fixed-demand energy answer different questions. Batched LLM service needs its dependencies modeled or replayed.

Measured demand and speed–power curves can favor fewer fast servers, more slow servers, or an intermediate allocation [70]. A controller minimizes power subject to the same demand and latency requirements.

10.4. Scheduling and Setup Can Preserve or Erase the Gain

Size-based scheduling and SOAP’s age-based policy analysis can improve response on existing hardware [72], [73]. Record size information and prediction error, then evaluate long jobs through conditional slowdown $\mathbb{E}[T \mid X = x]/x$ and class-specific latency.

Jobs needing m_j simultaneous devices obey $\sum_j \text{active } m_j \leq K_{\text{on}}$ for K_{on} available devices, plus locality, memory, and non-preemption constraints. Multiserver-job scheduling weighs short-job priority against occupying device groups [74].

AutoScale accounts for returning servers to service [75]; deterministic setup analysis shows how its distribution affects waiting [76]. Replaying equal admitted traces while recording power states and response reveals whether device savings lower fleet power.

11. DATACENTER IMPACT AND SCIENTIFIC WORKLOADS

11.1. Datacenter Energy and Operating Costs

For average IT power P_{IT} in kW, annual PUE $\zeta = E_{\text{facility}}/E_{\text{IT}}$, and tariff c_e in dollars/kWh,

$$C_E = 8760 P_{\text{IT}} \zeta c_e. \quad (67)$$

PUE includes cooling and other facility overhead exactly once. Multiplying facility-boundary energy by PUE again would double-count that overhead. Equation (67) concerns electricity expenditure, not total operating cost or revenue. Varying tariffs or carbon intensities require time-resolved energy and rates.

At fixed admitted demand, measured IT energy-per-request gain α_{IT} gives, if average power reduces accordingly,

$$\frac{E_{\text{facility},1}}{E_{\text{facility},0}} = \frac{\zeta_1}{\alpha_{\text{IT}} \zeta_0}. \quad (68)$$

This converts IT gain to facility energy using PUE values ζ_0, ζ_1 . Facility-boundary measurement captures it directly; response-time gains have separate service value.

The fleet example in Section 11.3 gives the monetary sensitivity under one declared set of assumptions. Figure 16 instead shows how power that remains unchanged limits the saving achievable from a local improvement.

Capacity expansion or deferred equipment purchases depend on memory, queueing, demand, and provisioning granularity; use measured capacity and costs. Operational carbon follows energy and grid intensity, with manufacturing impacts accounted separately.

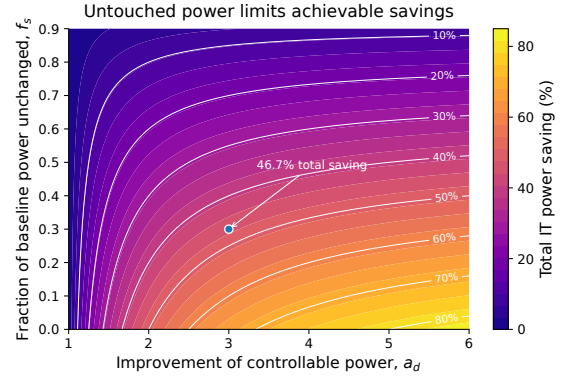


Fig. 16. IT power saving $100(1 - f_s)(1 - 1/a_d)$ percent at fixed completed-work rate. The example $f_s = 0.30$, $a_d = 3$ gives 46.7% saving, not the 66.7% obtained when all power scales. MKA impact: persistent memory, interfaces and idle capacity limit how much local efficiency reaches the electricity bill. All factors are assumed sensitivities.

11.2. The Part of the Power Budget That Remains

If fraction f_s of baseline IT power stays fixed and the rest improves by a_d at unchanged work rate,

$$\frac{\bar{P}_1}{\bar{P}_0} = f_s + \frac{1 - f_s}{a_d}, \quad \alpha_{\text{IT}} = \left(f_s + \frac{1 - f_s}{a_d} \right)^{-1}. \quad (69)$$

This applies Eq. (8) to power. Provisioned memory, interfaces, or other resources may retain their operating load.

For assumed $f_s = 0.30$ and $a_d = 3$, the resulting IT factor is 1.875, or 46.7% less power at the same work rate (Fig. 16). The local threefold factor is a sensitivity input, not a measured MKA result. Consolidation or gating can reduce the fixed fraction within memory and latency constraints; operating policy therefore determines financial value.

11.3. Prospective Six-Vector Optimization of Openly Released LLMs

Open-weight frontier models make model-specific systems optimization especially visible. Using published configurations current in September 2026, we compare Llama 4 Maverick, GLM-5.3, Kimi K3, and Qwen3.8-2.4T-A95B [77]–[80]. Maverick is a 400B/17B-active, 128-expert model with top-1 routing and a 1M-token context. GLM-5.3 uses the GLM-5.2 base architecture at 744B/40B active, 78 layers, 256 routed plus one shared expert, top-8 routing, DeepSeek Sparse Attention, and a 1M context. Kimi K3 scales to 2.8T/104B active, 93 layers, 896 routed plus two shared experts, top-16 routing, 69 KDA and 24 gated-MLA layers, and a 1M context. Qwen3.8-2.4T-A95B uses 2.4T/95B active parameters, 92 layers, 512 experts with 10 routed plus one shared active per token, 69 Gated DeltaNet and 23 full-attention layers, and 262K native context extensible to about 1.01M

tokens. These architectural differences directly change weight residency, context state, expert traffic, batching headroom, and power demand.

For serving model m , define a six-component control vector

$$z_m = (z_c, z_w, z_k, z_r, z_b, z_p), \quad (70)$$

where z_c selects compute kernels and numerical precision, z_w controls weight and expert residency, z_k controls KV and other request-state placement, z_r controls expert routing and interconnect placement, z_b controls batching, admission, and queueing, and z_p controls active capacity, power state, and the associated cooling load. These controls interact. Increasing batch can amortize weights while increasing KV residency and queueing; moving experts can reduce local memory pressure while increasing all-to-all traffic; lower precision can reduce traffic and power while tightening the quality constraint. The MKA operating point is therefore

$$z_m^* \in \arg \max_{z \in \mathcal{Z}_m} \Lambda_m(z) = \arg \min_{z \in \mathcal{Z}_m} E_m(z) T_m(z), \quad (71)$$

subject to the model's quality, memory, reliability, and service-level requirements. For a finite nonempty admissible candidate set with positive costs, a minimizing configuration exists. Finding it requires measured or calibrated $E_m(z)$ and $T_m(z)$; the objective alone is not a search algorithm. The six coupled controls describe an operating-point optimization within the chosen model and hardware configuration, distinct from the six substrate regimes.

Table 4 gives a neutral BF16-equivalent scale view; actual released precisions differ and belong to the compute/precision decision vector. GQA cache follows Eq. (13). For compressed latent attention, a first-order context-dependent state proxy is

$$V_{\text{state}}^{\text{latent}} \simeq L_{\text{cache}} s(r_{\text{KV}} + d_{\text{rope}}) b_{\text{kv}}, \quad (72)$$

where L_{cache} counts layers retaining context-dependent latent state. KDA and Gated DeltaNet recurrent state is context-independent and is accounted separately. These footprints are architecture-derived proxies, not measured HBM traffic.

The contrast is operational. Maverick combines comparatively low active compute with a 1M GQA context, so request state and queueing can dominate. GLM-5.3 enlarges resident weights and expert routing while using compressed sparse-attention state. Kimi K3 pushes expert residency, top-16 routing, active compute, and power much further, while its KDA/MLA hybrid reduces context-dependent cache growth. Qwen3.8 likewise shifts pressure toward very large resident expert state and routing; its DeltaNet/full-attention hybrid moderates native-context KV growth despite 2.4T total

parameters. Figure 17 makes the resulting pressure vectors explicit. These descriptors identify likely bottlenecks to measure. They do not establish a model's optimal configuration or a deployment speedup. The best admissible point depends on the measured request mix, hardware, and operating constraints; different models can sometimes share that point.

The fleet calculation makes the economic sensitivity explicit. With P_{fleet} in kW, fixed completed-request rate and quality, and unchanged PUE and electricity tariff,

$$S_{\text{yr}} = 8760 P_{\text{fleet}} \zeta c_e (1 - \alpha_{\text{IT}}^{-1}). \quad (73)$$

For $P_{\text{fleet}} = 5 \text{ GW} = 5 \times 10^6 \text{ kW}$, $\zeta = 1.20$, and $c_e = 0.07 \text{ USD/kWh}$, every 10% verified reduction in system-level IT energy per completed request corresponds to approximately \$368 million in annual electricity savings at fixed demand. A 30% reduction corresponds to $\alpha_{\text{IT}} = 1/0.70 \simeq 1.43$ and exceeds \$1.1 billion annually. These are sensitivity thresholds rather than assumed MKA performance. Their purpose is to translate measured system-level efficiency into a transparent economic scale: even moderate verified gains can become material when sustained across a hyperscale serving fleet.

The reference planner in Section 6 already exercises a restricted subset of this idea by coordinating location, transfers, setup, and timing. Its feasible improvements show that even partial all-inclusive optimization can change the selected plan. The four-model case study identifies the next deployment-oriented step: calibrate all six vectors from real traces for a chosen model, solve Eq. (71), and verify the predicted energy–time improvement at the same quality and demand.

11.4. Applications Organized by the Cost That Can Change

The framework is intentionally broader than accelerator serving. It applies whenever the cost of moving, synchronizing, or maintaining state is comparable with the cost of the arithmetic performed on that state. The same completed-work principle therefore creates different engineering questions in different fields. In databases and search, a useful optimization may keep an index shard, embedding table, or reranking stage resident instead of moving intermediate results across machines. In molecular simulation and drug discovery, neighbor lists, graph messages, force calculations, and repeated linear-algebra kernels create a mixture of local reuse and global reductions. In robotics and autonomy, perception, sensor fusion, planning, and control form a latency-critical graph in which a locally efficient accelerator is useless if copying or scheduling causes the control deadline to be missed.

Finance and regulated computation combine streaming dataflow with numerical accuracy and re-

Table 4

PUBLISHED ARCHITECTURE SNAPSHOT USED IN THE OPEN-MODEL COMPARISON. BF16-EQUIVALENT WEIGHT FOOTPRINTS USE TWO BYTES PER PARAMETER; CONTEXT-STATE ENTRIES USE THE PUBLISHED ATTENTION ARCHITECTURE AND STATED CACHE/STATE PROXY. THE TABLE SHOWS WHY A SINGLE SERVING RECIPE IS UNLIKELY TO BE OPTIMAL ACROSS MODELS.

Model	Total / active	Context	Layers / attention	Experts / routing	BF16-equivalent reference state	Dominant MKA pressure
Llama 4 Maverick	400B / 17B	1M	48 layers; GQA, 40 Q / 8 KV	128 experts; top-1	800 GB eq.; 192 GiB KV	Long-context state, batching, residency, power gating
GLM-5.3	744B / 40B	1M	78 layers; DSA, latent KV rank 512	256 routed + 1 shared; top-8	1,488 GB eq.; 87.8 GiB state proxy	Expert residency, routing, compressed long-context state
Kimi K3	2.8T / 104B	1M	93 layers; 69 KDA + 24 gated MLA	896 routed + 2 shared; top-16	5,600 GB eq.; 27 GiB latent cache proxy	Extreme expert residency, routing, active compute and power
Qwen3.8-2.4T-A95B	2.4T / 95B	262K native	92 layers; 69 DeltaNet + 23 full attn.	512 experts; 10 routed + 1 shared	4,800 GB eq.; 23 GiB native KV	Massive residency and routing; hybrid attention limits native-context state

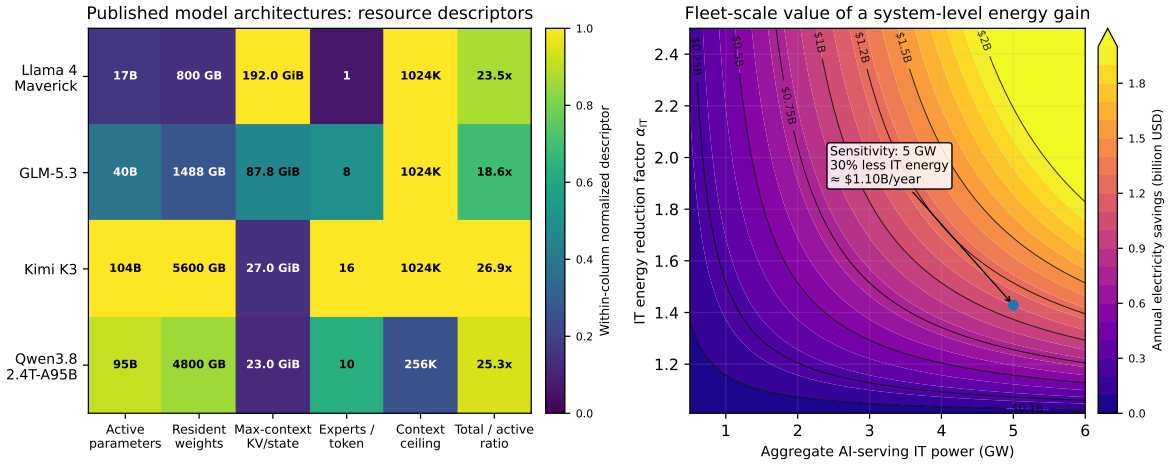


Fig. 17. Left: architecture-derived resource descriptors for four openly released MoE model families; raw values are printed in each cell and colors are normalized within each column. Right: annual electricity savings from Eq. (67) at PUE 1.20 and \$0.07/kWh as aggregate AI-serving power and IT energy reduction vary. At 5 GW, each 10% verified reduction corresponds to about \$368 million per year, while the marked 30% sensitivity point exceeds \$1.1 billion. MKA impact: model-specific resource demands guide a calibrated six-control search, while verified energy gains determine fleet-scale savings.

producible decisions [15]. Pricing and risk aggregation can reuse resident state; covariance updates test whether setup is repaid. For $(\Sigma + \lambda I)w = b$, symmetric positive-semidefinite Σ and $\lambda > 0$ give an SPD test case. The result needs a residual, conditioning estimate, and deterministic replay record. In Monte Carlo risk, a g -fold sample-rate increase reduces sampling error by $1/\sqrt{g}$ at fixed duration for independent finite-variance samples, or finishes a fixed sample count earlier.

Genomics presents less regular locality. Variation graphs represent alternative sequences and structural variation [81]; traversal and alignment combine index lookups, dynamic programming, and sparse state access. For positive total graph

traffic, a useful partition descriptor is

$$\ell_G = 1 - \frac{\sum_{e \in E_{cut}} n_e b_e}{\sum_{e \in E} n_e b_e}. \quad (74)$$

It records the fraction of traffic retained within partitions. High locality must be checked alongside bank balance and the maximum cut load. Read-mapping accuracy and the reference graph remain fixed when placements are compared.

Climate and scientific simulation add phase changes. In a d -dimensional stencil with local side length a and halo width h , the leading halo-to-volume ratio is approximately $2dh/a$ for $h \ll a$. Strong scaling therefore raises communication per local update. Coarse multigrid levels can favor different placements from fine levels. A useful test

follows the complete solve, including convergence, reductions, and remapping.

Telecommunications and edge inference expose hard timing limits. Factor-graph message passing underlies important decoding and inference methods [82]. LDPC decoding, channel estimation, beamforming, and sensor fusion offer distinct precision and locality tests. If I rounds are strictly sequential, $T \geq It_{\text{round,LB}}$; reducing local arithmetic alone leaves the round count unchanged. Evaluation fixes bit-error or estimation quality and includes required host, memory, and radio-interface work.

Table 5 gives representative workloads, controlled comparisons, and end-to-end observables for all nine application areas. A common accounting method makes their results interpretable while preserving each domain's quality and timing requirements.

11.5. Algorithmic Changes and the Implementation Program

The substrate axis is only part of the research space. Sparsity, quantization, fusion, and asynchronous updates can change the work graph itself. Reversible computation trades erasure against space and control [67]; stochastic physical devices provide another route to sampling and optimization [83]. Each change is evaluated at the requested answer quality. This is consistent with the broader post-Moore opportunity in algorithms, software, and architecture [84].

A proposed three-year program has concrete gates. In year one, publish the graph/state schema, calibrated cost adapters, and a digital backend with matched placements and locality sweeps. In year two, test analog reuse and conditioning, PIM update traffic, and optical utilization on complete interfaces. In year three, integrate a multi-node service and one non-AI application, reproduce the result at a second institution, and release both successful and failing operating regions. Quantum experiments proceed through fidelity and preparation-rate gates appropriate to their own tasks.

12. EVALUATION AND OPEN QUESTIONS

12.1. From the Reference Result to a Hardware Claim

Implement Section 5's backend and compare placements on one device. Start with reusable local state, then sweep global payload using Eq. (24) through the point where routing erases the benefit.

Use a strong alternative and an ablation with MKA placement disabled, matching area, memory, and precision. A common legal clock isolates placement; also report each placement's best legal clock. Later sweeps vary analog reuse/conditioning, PIM updates, or optical load at fixed required output.

The *MKA certificate* records Table 6's configuration and measurements, including failed and non-improving runs. Workloads and primary comparisons are selected before evaluation. Existing graph/dataflow systems require the same comparison contract. Calibrate costs on separate runs and test predictions on held-out workloads; the index alone supplies no scaling forecast.

12.2. Paired Gain Estimation and Uncertainty

For independent paired workload blocks, define

$$y_i = \log \frac{E_{0,i}T_{0,i}}{E_{1,i}T_{1,i}}, \quad \hat{G} = \exp(\bar{y}). \quad (75)$$

This equally weighted geometric-mean gain applies to independent paired blocks. A weighted population needs matching weights and variance or resampling. Fix population and estimand before testing.

For approximately normal independent block log ratios, a one-sided lower confidence bound is

$$L = \bar{y} - t_{1-\eta, n-1} \frac{s_y}{\sqrt{n}}. \quad (76)$$

Positive L supports geometric-mean improvement. Block bootstrap is an alternative under its sampling assumptions. Power samples from one thermal run remain dependent; inference needs independent blocks or a time-series model.

Energy and time uncertainty are correlated. With $u = \log(E_0/E_1)$ and $v = \log(T_0/T_1)$,

$$\text{Var}(y) = \text{Var}(u) + \text{Var}(v) + 2 \text{Cov}(u, v). \quad (77)$$

Paired resampling preserves covariance. Report systematic meter uncertainty separately: repetition reduces variability, not calibration bias.

For a planned detectable positive log-gain δ , block standard deviation σ_y , one-sided significance η , and power $1-\beta_s$, the normal approximation gives

$$n \gtrsim \frac{(z_{1-\eta} + z_{1-\beta_s})^2 \sigma_y^2}{\delta^2}. \quad (78)$$

Here z_p is a normal quantile. Pilot data estimate variability; multiple workloads or candidate searches need a declared comparison strategy or held-out tests.

12.3. Quality and Rare-Failure Estimates

Assess actual output errors on matched inputs with a predeclared non-inferiority margin before comparing performance; equal nominal precision can yield different errors.

For n independent Bernoulli trials with no observed failures, the one-sided exact upper bound on failure probability at confidence $1 - \eta$ is

$$p_U = 1 - \eta^{1/n}. \quad (79)$$

Since $\Pr(\text{zero failures}) = (1 - p)^n$, 2,995 independent failure-free trials place the 95% upper bound below 10^{-3} . Correlation invalidates that count; sparse tail observations likewise need adequate replication.

Table 5

EXPANDED CROSS-DOMAIN RESEARCH PROGRAM. REPRESENTATIVE WORKLOADS ILLUSTRATE HOW THE SAME MKA RULE BECOMES A DIFFERENT PLACEMENT, STATE, OR SCHEDULING QUESTION IN EACH DOMAIN. A BENEFIT IS ESTABLISHED ONLY WHEN THE LISTED END-TO-END OBSERVATIONS IMPROVE AT MATCHED OUTPUT QUALITY.

Domain	Representative workloads	Candidate MKA mechanism	Controlled comparison	Decisive observations
AI serving and agents	MoE/long-context serving; RAG; tool-using agents	Weight/expert/KV placement; prefix reuse; routing; batching; fewer tool-boundary transfers	Same model, quality, context/request/tool traces and admitted demand	Bytes/token; HBM and network traffic; tail latency; joules/request; active fleet power
Finance and risk	Monte Carlo; streaming risk; pricing; fraud graphs	Resident aggregation; matrix reuse; streaming pipelines; graph partitioning	Same market trace, sample count, residual/error tolerance, and reporting deadline	Time-to-result; deterministic replay; setup amortization; joules/scenario
Databases, search, recommendation	Vector search; graph queries; joins; embedding lookup; reranking	Co-locate index shards and embedding state; fuse retrieval/rerank stages; cache hot subgraphs	Same corpus/index, recall or ranking quality, query mix, and tail-latency target	Bytes/query; cache hit rate; cross-shard traffic; p99 latency; joules/query
Genomics	Variation-graph mapping; sequence alignment; pangenome queries	Co-located graph partitions, indexes, and dynamic-programming state	Same reference graph, reads, scoring parameters, and mapping accuracy	Cut traffic; bank imbalance; reads/s/J; time-to-analysis
Molecular, materials, drug discovery	Molecular dynamics; GNN potentials; docking; sparse eigensolvers	Neighbor-list locality; resident coordinates/state; phase-specific dense/sparse kernels	Same molecule/system, timestep or convergence criterion, numerical tolerance, and ensemble size	Time/trajectory; neighbor and reduction traffic; energy/step; candidates/day/W
Climate and HPC	Stencils; multigrid; CFD; weather ensembles; sparse solvers	Halo locality; hierarchy; phase-specific placement and remapping	Same problem size, discretization, solver tolerance, and convergence criterion	Time-to-solution; global reductions; halo bytes; remapping cost; energy/solve
Telecom and edge	LDPC; MIMO; beamforming; channel estimation; sensor fusion	Local message passing; DSP/NPU pipelines; precision and state residency	Same channel/SNR trace, deadline, error target, and radio-interface scope	Iterations/rounds; BER or estimation error; missed deadlines; energy including interfaces
Robotics and autonomy	Perception-fusion-planning-control; multi-camera/ LiDAR pipelines	Sensor-local preprocessing; graph partitioning across CPU/GPU/NPU; deadline-aware state placement	Same sensor log, perception metric, route/task, and control deadline	End-to-end control latency; copied bytes; missed-deadline rate; joules/control cycle
Quantum-native tasks	Distributed sensing; modular quantum algorithms; entanglement-assisted computation	Shared control; quantum-state locality; entanglement routing and preparation scheduling	Same quantum task, fidelity, protocol, path length, and classical-control boundary	Preparation success; useful completion rate; fidelity; classical/cryogenic wall power

Table 6

EVALUATION RECORD AND COMPARISON ERRORS PREVENTED.

Item	Required evidence	Error prevented
Work and quality	Input or demand trace; model or algorithm version; quality tolerance; failure definition.	Comparing different useful work or silently relaxing accuracy.
Physical boundary	Included hosts, memory, interfaces, and overhead; calibrated input-power trace.	Moving energy outside the comparison or counting it twice.
Execution	Placement, routes, runtime and power states; release/completion events, clock resolution/error, and timing estimator.	Attributing software changes to devices or using incompatible timing boundaries.
Resources	Memory footprint, effective traffic, link provision, and physical or cost budget.	Comparing unequal hardware resources as a topology improvement.
Statistical decision	Independent paired blocks, uncertainty, primary endpoint, and tested comparisons.	Treating repeated samples or selected configurations as independent evidence.
Reproduction	Source, configurations, numerical inputs, and scripts for all reported results.	Providing plots that cannot be independently reconstructed.

12.4. Figure Provenance and Reproduction

The source package maps every figure to its equation, parameter values, generation script, and numeric data in `FIGURE_GUIDE.md`. Closed-form surfaces use the stated analytical models; planner plots use the complete synthetic suite, including ties and failures. The open-model panel uses architecture descriptors and storage proxies, not measured utilization or a computed six-control optimum. Its colors are normalized separately within each column. Economic surfaces vary as-

sumed savings; external device results retain the measurement scope of the cited study. None of these plots is presented as a measured end-to-end MKA hardware result.

12.5. Scope of the Evidence

The 540-case study measures decisions in a synthetic cost model. Exact enumeration checks the implementation against that model; it does not calibrate the costs. The modular-affine replay tests common operation semantics and ownership

transfers. The planner does not yet schedule concurrent DAGs, implement a Honeycomb physical network, or run a real decoder. These boundaries define the next experiments rather than changing the interpretation of the reported ratios. All plot grids, generated instances, selected paths, failures, and verification code accompany the manuscript.

12.6. Operating Limits and Falsification

Consistent failure to improve the matched-budget energy–time frontier weakens placement. Excess routing or scheduling identifies compiler limits. A substrate fails where full interface and setup costs exceed its acceptable-quality benefit. Evaluate against the original workload criteria.

Conclusions depend on objective, tested workloads, legality checks, uncertainty set, and quantum-task semantics.

Report the first failing case alongside the best: the reuse, memory, or global-traffic threshold should explain the decision.

13. CONCLUSION

MKA proposes that the graph of useful work should remain the organizing object as computing technology changes. Its central claim is broader than the energy–time equation: a component or local hardware improvement becomes architectural progress only when its benefit survives data movement, state placement, interfaces and conversion, scheduling and queueing, accuracy, reliability, and service-level constraints at the boundary of completed useful work. The MKA Law supplies a quantitative objective for this completed-work survival principle, while the workload contract fixes the conditions under which the comparison is meaningful. This extends the Honeycomb direction initiated by Veljko Milutinović, its early development with Dragana Milutinović and B. Soucek, and the later mapping analysis with Borko Furht into a research program spanning physical substrates [6], [7], [9]. The enduring aim is to make architectural knowledge accumulate as devices change.

The reference planner establishes a first computational result within this program. For the stated serial-workflow model, it preserves an optimal admissible plan while accounting for data location, available memory, recurring transfers, and setup. Exhaustive checks verify the recurrence against direct enumeration. The experiments show that coordinated choices can improve both energy and completion time on unchanged modeled resources and recover feasible schedules that incomplete accounting misses. Comparisons with strong global alternatives show that much of the gain comes from representing the costs correctly and coordinating decisions beyond the next operation. These results support a concrete research direction: extend explicit ownership and full-cost planning to branching

workflows, concurrent execution, and calibrated hardware models.

The wider scientific opportunity lies in the interaction between mechanisms. Keeping an intermediate value resident can eliminate a conversion and a transfer, but may consume memory needed by another operation. A faster device can reduce queueing, while its persistent power can alter the preferred schedule. A new interconnect can relieve a communication cut, yet change which partition is most efficient. MKA provides a common formulation in which these effects can be evaluated together. The communication bounds, accuracy conditions, and uncertainty models identify which change is worth implementing and which additional measurement could change the decision.

The six regimes turn this formulation into a long-term experimental agenda. Reconfigurable digital hardware provides an immediate platform for testing placement and state ownership. Analog operations and processing near memory offer ways to reduce suitable numerical and data-access costs; photonic links address transfers that remain expensive after partitioning. Quantum-dot and entanglement-assisted systems extend the agenda to tasks whose useful resources include quantum states and their preparation. Shared workload descriptions and explicit interfaces can make the findings from these experiments reusable. Each implementation contributes knowledge about the conditions under which a physical mechanism improves completed work.

The consequences extend from individual applications to computing infrastructure. Artificial intelligence services expose repeated access to large models and growing request state. Finance, genomics, and scientific simulation add streaming decisions, irregular graphs, and numerical convergence requirements. Telecommunications and edge inference place tight deadlines on iterative and sensor-driven computation. Their differences create complementary tests of the same architectural questions. The facility scenarios show how sustained system-level energy reductions could produce annual savings on a billion-dollar scale across a sufficiently large datacenter fleet under the stated assumptions. For scientific users, improved efficiency can instead support larger simulations, more samples, or faster analysis within the same available resources.

The open-model analysis motivates a joint optimization problem over six coupled serving controls: compute and precision, resident weights, KV and request state, expert and interconnect routing, batching and queueing, and active power with cooling. Llama 4 Maverick, GLM-5.3, Kimi K3, and Qwen3.8 occupy visibly different points in this resource space [77]–[80]. Their different demands motivate model-specific searches on the same

hardware fleet. Equation (71) turns that observation into a reproducible search problem. The fleet sensitivity in Section 11.3 translates verified IT energy reductions into electricity expenditure at unchanged demand, PUE, and tariff. It does not predict a saving from the architecture descriptors alone. The scientific claim is broader: optimizing all six vectors together exposes trade-offs that isolated kernel, memory, network, or scheduler metrics cannot resolve on their own.

This program builds on specific contributions from the researchers whose work we acknowledge. John L. Hennessy's compiler-machine co-design, his technology-architecture analysis with Norman Jouppi, and his quantitative and post-Moore work with David A. Patterson provide the evaluation discipline [1], [10]–[14]. Oskar Mencer's dataflow research provides an implementation precedent for predictable spatial execution [15], [16]. Mor Harchol-Balter's scheduling and server-power research connects execution choices to the service experienced by users [68], [70], [71], [75]. Dimitris Bertsimas's robust and prescriptive optimization informs decisions under uncertain demand [44], [45], [47]; Daniela Rus's graph grammars and decentralized reconfiguration inform the generation and control of realizable structures [48]–[50]. The analog-solver work of Osama Abdelaleim, Arun Prakash, Ayhan Irfanoglu, and Milutinović supplies a specific circuit proposal for the numerical branch of the roadmap [52]. Dan Shechtman's discovery of quasicrystals remains a methodological inspiration for examining structural assumptions. The relevance is not a direct physical analogy between crystallography and computer architecture, but the lesson that a successful prevailing structural model should not be mistaken for the full set of physically meaningful possibilities. MKA carries that attitude into architecture by treating topology, locality, and substrate boundaries as hypotheses to be measured and challenged rather than inherited constraints [85].

The next phase should connect the reference implementation to measured platforms and independent application studies. A shared representation of work and state, calibrated cost adapters, and reproducible comparisons would let researchers improve different parts of the system while testing their effect on the same completed workload. The long-term contribution sought is a cumulative science of architectural choice across substrates, with useful computation per energy and time budget as its common objective. We welcome the community to test and challenge it.

DECLARATIONS AND ACKNOWLEDGEMENTS

No funding was received for this work. Code, synthetic benchmark inputs, selected plans, and complete plot data accompany the manuscript. The authors state that the manuscript is unpublished

and is not under review at another journal or conference.

V.M. originated the Honeycomb research direction and provided senior scientific guidance. A.K. developed the six-regime roadmap and its industrial framing. D.K. led the MKA measurement framework, analytical development, and manuscript preparation. E.K. contributed figure preparation, presentation, and supporting research. The Milutinović and Katulevskiy families contributed equally to this work.

We are especially grateful to Professor John L. Hennessy for the intellectual influence of MIPS, VLSI architecture, technology-architecture analysis with Norman Jouppi, Stanford FLASH, and post-Moore work with David Patterson [10]–[14]. Their quantitative, workload-based approach guides our evaluation.

We thank Professor Mor Harchol-Balter of Carnegie Mellon University for her encouragement and the scheduling, server-farm power, and capacity-management research informing our service analysis [68], [70], [71], [75]. She met with the authors, shared and discussed prior findings, and encouraged them to develop and write this direction.

We acknowledge Professor Dimitris Bertsimas of MIT for robust optimization and contextual decision methods [44], [45], [47], which inform the uncertain-demand and budgeted-capacity analysis.

We acknowledge Professor Daniela Rus of MIT for decentralized reconfiguration, graph grammars, and joint design-control search [48]–[50]. Their application to MKA's generation and verification is developed by the authors.

We thank Oskar Mencer for dataflow research; Borko Furht and Dragana Milutinović for the parent Honeycomb analysis; and Osama Abdelaleim, Arun Prakash, and Ayhan Irfanoglu for the analog-solver work with V.M. Discussions of Dan Shechtman's quasicrystal work, whose discovery challenged long-standing assumptions about permissible order in solids, encouraged us to re-examine inherited structural assumptions in computing and to treat regularity and topology as experimentally testable design choices rather than fixed conventions [85].

Professors Hennessy, Harchol-Balter, Bertsimas, and Rus did not participate in preparing the manuscript and have not reviewed the final text. These acknowledgments recognize intellectual influence and encouragement without implying review or endorsement.

REFERENCES

- [1] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 6th ed. Cambridge, MA, USA: Morgan Kaufmann, 2017, ISBN: 978-0-12-811905-1.
- [2] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *2014 IEEE Int. Solid-State Circuits Conf. Digest of Technical Papers*, pp. 10–14, 2014, doi: 10.1109/ISSCC.2014.6757323.

- [3] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, 1965.
- [4] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 16344–16359.
- [5] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with PagedAttention," in *Proc. 29th ACM Symp. Operating Systems Principles (SOSP)*, 2023, pp. 611–626, doi: 10.1145/3600006.3613165.
- [6] D. Milutinović, V. Milutinović, and B. Soucek, "The Honeycomb Architecture," *Computer*, vol. 20, no. 4, pp. 81–83, Apr. 1987.
- [7] V. Milutinović, "Mapping of neural networks on the honeycomb architecture," *Proc. IEEE*, vol. 77, no. 12, pp. 1875–1878, Dec. 1989.
- [8] J. E. Hoch, *The Honeycomb Architecture: Prototype Analysis and Design*, Purdue University School of Electrical Engineering Technical Report TR-EE-87-28, Dec. 1987.
- [9] V. Milutinović, D. Milutinović, and B. Furht, "On mapping of topological structures and mathematical algorithms onto specialized architectures for graph-based computing," *Journal of Big Data*, vol. 13, art. 34, 2026, doi: 10.1186/s40537-025-01326-z.
- [10] J. L. Hennessy, N. P. Jouppi, S. Przybylski, C. Rowen, T. Gross, F. Baskett, and J. Gill, "MIPS: A microprocessor architecture," in *Proc. 15th Annual Symp. Microprogramming (MICRO)*, 1982, pp. 17–22.
- [11] J. L. Hennessy, "VLSI processor architecture," *IEEE Transactions on Computers*, vol. C-33, no. 12, pp. 1221–1246, Dec. 1984.
- [12] J. L. Hennessy and N. P. Jouppi, "Computer technology and architecture: An evolving interaction," *Computer*, vol. 24, no. 9, pp. 18–29, Sept. 1991, doi: 10.1109/2.84896.
- [13] M. Heinrich, D. Ofelt, M. A. Horowitz, and J. L. Hennessy, "Hardware/software co-design of the Stanford FLASH multiprocessor," *Proceedings of the IEEE*, vol. 85, no. 3, pp. 455–466, Mar. 1997, doi: 10.1109/5.558720.
- [14] J. L. Hennessy and D. A. Patterson, "A new golden age for computer architecture," *Communications of the ACM*, vol. 62, no. 2, pp. 48–60, Feb. 2019, doi: 10.1145/3282307.
- [15] D. Oriato, S. Girdlestone, and O. Mencer, "Dataflow computing in extreme performance conditions," *Advances in Computers*, vol. 96, pp. 105–137, 2015, doi: 10.1016/b.s.adcom.2014.11.002.
- [16] N. Voss, B. Kwaadgras, O. Mencer, W. Luk, and G. Gaydadjiev, "On predictable reconfigurable system design," *ACM Transactions on Architecture and Code Optimization*, vol. 18, no. 2, art. 17, 2021, doi: 10.1145/3436995.
- [17] N. Voss, M. Bacis, O. Mencer, G. Gaydadjiev, and W. Luk, "Convolutional neural networks on dataflow engines," in *Proc. 35th IEEE International Conference on Computer Design*, 2017, pp. 435–438, doi: 10.1109/ICCD.2017.77.
- [18] A. Parashar et al., "Timeloop: A systematic approach to DNN accelerator evaluation," in *IEEE ISPASS*, 2019, pp. 304–315, doi: 10.1109/ISPASS.2019.00042.
- [19] H. Kwon, P. Chatarasi, M. Pellauer, A. Parashar, V. Sarkar, and T. Krishna, "Understanding reuse, performance, and hardware cost of DNN dataflow: A data-centric approach," in *Proc. MICRO*, 2019, pp. 754–768, doi: 10.1145/3352460.3358252.
- [20] Y. N. Wu, J. S. Emer, and V. Sze, "Accelerger: An architecture-level energy estimation methodology for accelerator designs," in *IEEE/ACM ICCAD*, 2019, pp. 1–8, doi: 10.1109/ICCAD45719.2019.8942149.
- [21] T. Andrulis, J. S. Emer, and V. Sze, "CiMLoop: A flexible, accurate, and fast compute-in-memory modeling tool," in *IEEE ISPASS*, 2024, pp. 10–23, doi: 10.1109/ISPASS61541.2024.00012.
- [22] J. Ragan-Kelley, A. Adams, S. Paris, M. Levoy, S. Amarasinghe, and F. Durand, "Decoupling algorithms from schedules for easy optimization of image processing pipelines," *ACM Trans. Graph.*, vol. 31, no. 4, art. 32, 2012, doi: 10.1145/2185520.2185528.
- [23] M. Bansal, D. Sharlet, J. Ragan-Kelley, and S. Amarasinghe, "Lightweight and locality-aware composition of black-box subroutines," *Proc. ACM Program. Lang.*, vol. 9, PLDI, art. 189, pp. 1043–1067, 2025, doi: 10.1145/3729292.
- [24] IEEE, "Systems and Architectures," *International Roadmap for Devices and Systems*, 2024 ed. [Online]. Available: https://irds.ieee.org/images/files/pdf/2024/2024IRDS_SA.pdf.
- [25] E. Q. V. Martins, "On a multicriteria shortest path problem," *European Journal of Operational Research*, vol. 16, no. 2, pp. 236–245, 1984, doi: 10.1016/0377-2217(84)90077-8.
- [26] G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proc. AFIPS Spring Joint Computer Conf.*, 1967, pp. 483–485, doi: 10.1145/1465482.1465560.
- [27] R. Gonzalez and M. Horowitz, "Energy dissipation in general purpose microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 9, pp. 1277–1284, Sept. 1996, doi: 10.1109/4.535411.
- [28] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [29] J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai, "GQA: Training generalized multi-query transformer models from multi-head checkpoints," in *Proc. EMNLP*, 2023, pp. 4895–4901, doi: 10.18653/v1/2023.emnlp-main.298.
- [30] N. Shazeer, "Fast transformer decoding: One write-head is all you need," arXiv:1911.02150, 2019.
- [31] N. Shazeer, "GLU variants improve Transformer," arXiv:2002.05202, 2020.
- [32] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," arXiv:2307.09288, 2023.
- [33] M. Shueybi et al., "Megatron-LM: Training multi-billion parameter language models using model parallelism," arXiv:1909.08053, 2019.
- [34] D. Narayanan et al., "Efficient large-scale language model training on GPU clusters using Megatron-LM," in *Proc. Int. Conf. High Performance Computing, Networking, Storage and Analysis (SC)*, 2021, doi: 10.1145/3458817.3476209.
- [35] W. Fedus, B. Zoph, and N. Shazeer, "Switch Transformers: Scaling to trillion parameter models with simple and efficient sparsity," *JMLR*, vol. 23, no. 120, 2022.
- [36] G.-I. Yu, J. S. Jeong, G.-W. Kim, S. Kim, and B.-G. Chun, "Orca: A distributed serving system for transformer-based generative models," in *16th USENIX Symp. Operating Systems Design and Implementation*, 2022, pp. 521–538.
- [37] F. T. Leighton, B. M. Maggs, and S. B. Rao, "Packet routing and job-shop scheduling in $O(\text{congestion} + \text{dilation})$ steps," *Combinatorica*, vol. 14, no. 2, pp. 167–186, 1994, doi: 10.1007/BF01215349.
- [38] C. D. Thompson, "Area-time complexity for VLSI," in *Proc. 11th ACM Symp. Theory of Computing*, 1979, pp. 81–88, doi: 10.1145/800135.804401.
- [39] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI, USA: American Mathematical Society, 1997.
- [40] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998, doi: 10.1137/S1064827595287997.
- [41] B. S. Landman and R. L. Russo, "On a pin versus block relationship for partitions of logic graphs," *IEEE Trans. Comput.*, vol. C-20, no. 12, pp. 1469–1479, 1971, doi: 10.1109/T-C.1971.223159.
- [42] S. Williams, A. Waterman, and D. A. Patterson, "Roofline: An insightful visual performance model for multicore architectures," *Communications of the ACM*, vol. 52, no. 4, pp. 65–76, 2009, doi: 10.1145/1498765.1498785.
- [43] J.-W. Hong and H. T. Kung, "I/O complexity: The red-blue pebble game," in *Proc. 13th ACM Symp. Theory of Computing*, 1981, pp. 326–333, doi: 10.1145/800076.802486.
- [44] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004, doi: 10.1287/opre.1030.0065.
- [45] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011, doi: 10.1137/080734510.

- [46] D. Bertsimas, D. den Hertog, and J. Pauphilet, "Probabilistic guarantees in robust optimization," *SIAM Journal on Optimization*, vol. 31, no. 4, pp. 2893–2920, 2021, doi: 10.1137/21M1390967.
- [47] D. Bertsimas and N. Kallus, "From predictive to prescriptive analytics," *Management Science*, vol. 66, no. 3, pp. 1025–1044, 2020, doi: 10.1287/mnsc.2018.3253.
- [48] A. Zhao, J. Xu, M. Konaković Luković, J. Hughes, A. Spielberg, D. Rus, and W. Matusik, "RoboGrammar: Graph grammar for terrain-optimized robot design," *ACM Transactions on Graphics*, vol. 39, no. 6, art. 188, pp. 1–16, 2020, doi: 10.1145/3414685.3417831.
- [49] J. Xu, A. Spielberg, A. Zhao, D. Rus, and W. Matusik, "Multi-objective graph heuristic search for terrestrial robot design," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2021, doi: 10.1109/ICRA48506.2021.9561818.
- [50] Z. Butler, K. Kotay, D. Rus, and K. Tomita, "Generic decentralized control for lattice-based self-reconfigurable robots," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 919–937, 2004, doi: 10.1177/0278364904044409.
- [51] S. Ambrogio et al., "An analog-AI chip for energy-efficient speech recognition and transcription," *Nature*, vol. 620, pp. 768–775, 2023, doi: 10.1038/s41586-023-06337-5.
- [52] O. Abdelaleim, A. Prakash, A. Irfanoglu, and V. Milutinović, "Analog solver circuit for linear symmetric positive-definite systems at a complexity independent of matrix size," arXiv:2508.04609v2, 24 Jan. 2026. Preprint, doi: 10.48550/arXiv.2508.04609.
- [53] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. Philadelphia, PA, USA: SIAM, 2002, doi: 10.1137/1.9780898718027.
- [54] J. H. Kim et al., "Aquabolt-XL HBM2-PIM, LPDDR5-PIM with in-memory processing, and AXDIMM with acceleration buffer," *IEEE Micro*, vol. 42, no. 3, pp. 20–30, 2022, doi: 10.1109/MM.2022.3164651.
- [55] D. A. B. Miller, "Attojoule optoelectronics for low-energy information processing and communications," *Journal of Lightwave Technology*, vol. 35, no. 3, pp. 346–396, 2017, doi: 10.1109/JLT.2017.2647779.
- [56] S. Daudlin et al., "Three-dimensional photonic integration for ultra-low-energy, high-bandwidth interchip data links," *Nature Photonics*, vol. 19, pp. 502–509, 2025, doi: 10.1038/s41566-025-01633-0.
- [57] D. Loss and D. P. DiVincenzo, "Quantum computation with quantum dots," *Phys. Rev. A*, vol. 57, pp. 120–126, 1998.
- [58] R. Hanson et al., "Spins in few-electron quantum dots," *Rev. Mod. Phys.*, vol. 79, pp. 1217–1265, 2007.
- [59] F. Borsoi et al., "Shared control of a 16 semiconductor quantum dot crossbar array," *Nature Nanotechnology*, vol. 19, pp. 21–27, 2024, doi: 10.1038/s41565-023-01491-3.
- [60] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters, "Teleporting an unknown quantum state via dual classical and EPR channels," *Phys. Rev. Lett.*, vol. 70, pp. 1895–1899, 1993.
- [61] S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, eaam9288, 2018.
- [62] K. Azuma et al., "Quantum repeaters: From quantum networks to the quantum internet," *Rev. Mod. Phys.*, vol. 95, 045006, 2023.
- [63] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, Apr. 1999, doi: 10.1109/49.761034.
- [64] B. Murmann, "The race for the extra decibel: A brief review of current ADC performance trajectories," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 3, pp. 58–66, 2015, doi: 10.1109/MSSC.2015.2442393.
- [65] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 14, no. 5, pp. 501–513, May 2006, doi: 10.1109/TVLSI.2006.876103.
- [66] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. Res. Dev.*, vol. 5, no. 3, pp. 183–191, 1961.
- [67] C. H. Bennett, "Logical reversibility of computation," *IBM J. Res. Dev.*, vol. 17, no. 6, pp. 525–532, 1973.
- [68] M. Harchol-Balter, *Performance Modeling and Design of Computer Systems: Queueing Theory in Action*. Cambridge, U.K.: Cambridge University Press, 2013, doi: 10.1017/CBO9781139226424.
- [69] M. Harchol-Balter, "Open problems in queueing theory inspired by datacenter computing," *Queueing Systems*, vol. 97, pp. 3–37, 2021, doi: 10.1007/s11134-020-09684-6.
- [70] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," in *Proc. ACM SIGMETRICS/Performance*, 2009, pp. 157–168, doi: 10.1145/1555349.1555368.
- [71] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, vol. 67, no. 11, pp. 1155–1171, 2010, doi: 10.1016/j.peva.2010.08.009.
- [72] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal, "Size-based scheduling to improve Web performance," *ACM Transactions on Computer Systems*, vol. 21, no. 2, pp. 207–233, May 2003, doi: 10.1145/762483.762486.
- [73] Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf, "SOAP: One clean analysis of all age-based scheduling policies," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 1, art. 16, Mar. 2018, doi: 10.1145/3179419.
- [74] I. Grosz, Z. Scully, M. Harchol-Balter, and A. Scheller-Wolf, "Optimal scheduling in the multiserver-job model under heavy traffic," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 3, art. 51, Dec. 2022, doi: 10.1145/3570612.
- [75] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A. Kozuch, "AutoScale: Dynamic, robust capacity management for multi-tier data centers," *ACM Transactions on Computer Systems*, vol. 30, no. 4, art. 14, Nov. 2012, doi: 10.1145/2382553.2382556.
- [76] J. K. Williams, M. Harchol-Balter, and W. Wang, "The M/M/k with deterministic setup times," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 3, art. 56, Dec. 2022, doi: 10.1145/3570617.
- [77] Meta, "Llama 4 Maverick 17B-128E Instruct model card," Apr. 2025. [Online]. Available: <https://huggingface.co/meta-llama/Llama-4-Maverick-17B-128E-Instruct>.
- [78] Z.ai, "GLM-5.3 model card and GLM-5 series architecture," 2026. [Online]. Available: <https://huggingface.co/zai-org/GLM-5.3>; <https://github.com/zai-org/GLM-5>.
- [79] Moonshot AI, "Kimi K3 model card," 2026. [Online]. Available: <https://huggingface.co/moonshotai/Kimi-K3>.
- [80] Qwen Team, "Qwen3.8-2.4T-A95B model card," Aug. 2026. [Online]. Available: <https://huggingface.co/Qwen/Qwen3.8-2.4T-A95B>.
- [81] E. Garrison et al., "Variation graph toolkit improves read mapping by representing genetic variation in the reference," *Nature Biotechnology*, vol. 36, pp. 875–879, 2018, doi: 10.1038/nbt.4227.
- [82] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, 2001, doi: 10.1109/18.910572.
- [83] W. A. Borders et al., "Integer factorization using stochastic magnetic tunnel junctions," *Nature*, vol. 573, pp. 390–393, 2019, doi: 10.1038/s41586-019-1557-9.
- [84] C. E. Leiserson et al., "There's plenty of room at the top: What will drive computer performance after Moore's law?" *Science*, vol. 368, no. 6495, eaam9744, 2020, doi: 10.1126/science.aam9744.
- [85] D. Shechtman, "Quasi-periodic materials: A paradigm shift in crystallography," Nobel Lecture, Stockholm University, 8 Dec. 2011.

AUTHOR BIOGRAPHIES

Daniil Katulevskiy is a computer scientist and systems researcher working on artificial intelligence infrastructure, heterogeneous computing, distributed inference, accelerator performance, edge computing, and post-Moore computer architecture. He graduated in Computer Science from the University of California, San Diego. His technical work includes AI and performance engineering on Qualcomm and NVIDIA computing platforms, with particular emphasis on memory movement, accelerator utilization, model serving, and system-level efficiency. Daniil is the lead author and primary technical contributor to the present formulation of the Milutinović–Katulevskiy Architecture (MKA). His contributions include the completed-work measurement framework and MKA progress criterion, the workload and admissibility model, graph placement and communication analysis, the exact reference-planning formulation, heterogeneous-substrate break-even models, large-model serving analysis, uncertainty-aware validation, and the connection between system-level efficiency and data-center economics. His broader research interests include scalable AI systems, secure and privacy-preserving computing, and new computational architectures that combine hardware, software, communication, and physical constraints within a common systems framework.

Elizaveta Katulevskaya is a researcher with interests in quantitative analysis, artificial intelligence, applied mathematics, and interdisciplinary scientific communication. She has studied through Stanford University Summer Session and Letovo School. Within the MKA research program, Elizaveta contributed to the supporting research base, scientific presentation, visualization, and organization of evidence across multiple computing substrates and application domains. Her work on the project has focused especially on making complex architectural comparisons interpretable across disciplines and on supporting the presentation of quantitative results.

Aleksey Katulevskiy is a researcher and entrepreneur working at the intersection of artificial intelligence, financial infrastructure, digital assets, and emerging technology. He is affiliated with the Polity Research Initiative on Applied Artificial Intelligence in Regulated Web3 Finance, where his work concerns the application of AI and computational infrastructure to regulated financial and digital-asset systems. Within the MKA program, Aleksey was a major contributor to the development of the six-regime substrate roadmap, the industrial and deployment strategy, the cross-sector application framework, and the economic interpretation of architectural efficiency. His broader interests include AI-enabled financial infrastructure, large-scale digital systems, technology commercialization, and the institutional deployment of advanced computing in regulated environments.

Veljko Milutinović is a computer architect whose research spans microprocessor and VLSI design, parallel and dataflow computing, graph-based architectures, and energy-efficient high-performance computation. He received his Ph.D. from the University of Belgrade and spent a substantial part of his early academic career at Purdue University before continuing his research and teaching through later appointments in Europe and the United States. He is a Life Fellow of the IEEE, a member of Academia Europaea, a member of the Serbian Academy of Engineering Sciences, and a Foreign Member of the Montenegrin Academy of Sciences and Arts. His earlier work includes the co-design of pioneering DARPA-supported GaAs computing systems and the development of graph-oriented and dataflow architectures, while his later research has emphasized energy-efficient computing, algorithm-to-hardware mapping, and new paradigms for high-performance computation. Veljko originated the Honeycomb research direction that provides the architectural lineage for MKA. That line includes the 1987 *Computer Honeycomb* architecture paper with Dragana Milutinović and Branislav Souček, his 1989 *Proceedings of the IEEE* work on mapping neural networks onto Honeycomb, James E. Hoch's Purdue prototype analysis conducted under Veljko's supervision, and later work extending Honeycomb concepts to the mapping of graphs and mathematical algorithms. In the present MKA program, Veljko provides the senior scientific direction and the historical architectural foundation from which the six-regime roadmap develops.

Correspondence:
dkatulevskiy@gmail.com
elkatul@stanford.edu
alexkatul73@gmail.com
vm@etf.rs