Trace-Indexed Multilevel I/O Lower Bounds for Classical GEMM

Anonymous Authors

Abstract

Communication lower bounds for matrix multiplication are often stated as separate requirements for replication, reduction, and finite-capacity reuse. Combining these requirements is subtle: one transfer can satisfy more than one requirement, while data, direction, receiver, and time-order constraints can prevent other transfers from being shared. We study this issue for classical GEMM on a fixed multilevel hierarchy under word-volume accounting, one computation per product, and no free replication or recomputation. We define labeled lineage and phase events, construct a compatibility graph (equivalently, a unit-capacity flow network), and prove the overlap-corrected bound $Q \geq J + L - \nu(G)$. The bound reduces to the usual max envelope under complete compatibility and is strictly stronger when labels force a sparse graph. We give canonical extraction from arbitrary transfer traces, a matching-schedule theorem for a restricted nested interface, and counterexamples showing why unconditional addition is false. We state the safe multilevel I/O envelope and recover the classical one-level $n^3/\sqrt{M}$ scale as a special case. Exact small-GEMM certificates and a CPU blocked-GEMM study illustrate the model's predictions and limitations.

CCS Concepts

• **Theory of computation** → **Parallel computing models**; • **Computer systems organization** → *Distributed architectures*; • **Mathematics of computing** → *Mathematical optimization.*

Keywords

parallel algorithms, communication lower bounds, matrix multiplication, memory hierarchy, I/O complexity

ACM Reference Format:

Anonymous Authors. 2026. Trace-Indexed Multilevel I/O Lower Bounds for Classical GEMM. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '27).* ACM, New York, NY, USA, 12 pages.

1 Introduction

Communication, rather than arithmetic, often determines the cost of classical matrix multiplication on modern parallel machines. Existing lower bounds quantify several mechanisms: projection data must cross processor or memory boundaries, and limited fast memory forces data to be reloaded. A common but unsafe step is to add these requirements. The same physical transfer may satisfy both obligations.

This paper asks a narrower question than "what is the communication complexity of GEMM?" We ask: *when can two valid communication lower-bound certificates be paid by the same transfer, and when does the hierarchy prevent that overlap?*

We fix a multilevel hierarchy $H_0 \supset H_1 \supset \cdots \supset H_R$ with charged edges $E_r = (H_{r-1}, H_r)$. A lineage event records a required copy, broadcast, or partial reduction. A phase event records a transfer required by a finite-capacity phase argument. Events carry edge, direction, word, endpoints, and time labels. Two events can share a transfer only if all labels are compatible. This gives a bipartite graph G and the central inequality

$$Q \geq J + L - \nu(G). \tag{1}$$

Our contributions are:

- a canonical labeled phase extraction from any fully charged transfer trace;
- the matching and weighted-flow lower bound in Eq. 1;
- explicit slot-embedding conditions and a restricted interface theorem with matching schedules;
- a minimal strict-correction certificate and an outer-product family that rules out unconditional $J + L$ addition; and
- exact small-instance checks and CPU experiments that separate aggregate word volume from multilevel capacity behavior.

The paper does not claim that every GEMM hierarchy has a positive correction. Our result is a compositional framework: a strict numerical improvement requires a proof that the relevant compatibility graph is sparse.

2 Background and Positioning

2.1 Communication as a geometric constraint

For a product $a_{ik}b_{kj}$, the three projections (i,k), (k,j), and (i,j) identify one A word, one B word, and one C accumulator. This three-projection structure is the reason that both the sequential I/O bound and the parallel communication bound have a three-dimensional geometric form. If a processor performs a set S of products, then

$$|S|^2 \leq |\pi_A(S)|\,|\pi_B(S)|\,|\pi_C(S)|, \qquad |\pi_A(S)|+|\pi_B(S)|+|\pi_C(S)| \geq 3|S|^{2/3}.$$

The first inequality is the Loomis–Whitney/HBL step; the second is AM–GM. The resulting projection footprint is a lower bound on data that must be made available to the processor, independently of whether the implementation uses a 2D, 2.5D, or 3D processor grid.

There are two different ways in which this footprint becomes communication. Across processors, an entry that is used by several owners must be replicated, and partial C values must be merged. Across a finite-capacity memory boundary, the same entry may need to be fetched again after it is evicted. Both arguments are individually sound. The difficulty studied here is that they refer to the same physical transfer stream. A first arrival can also be a phase reload; a partial store can also be a boundary reduction. Adding the two certificates without checking their labels can therefore overcount.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SPAA '27, Atlanta, GA, USA

© 2026 Copyright held by the owner/author(s). Publication rights licensed to ACM.

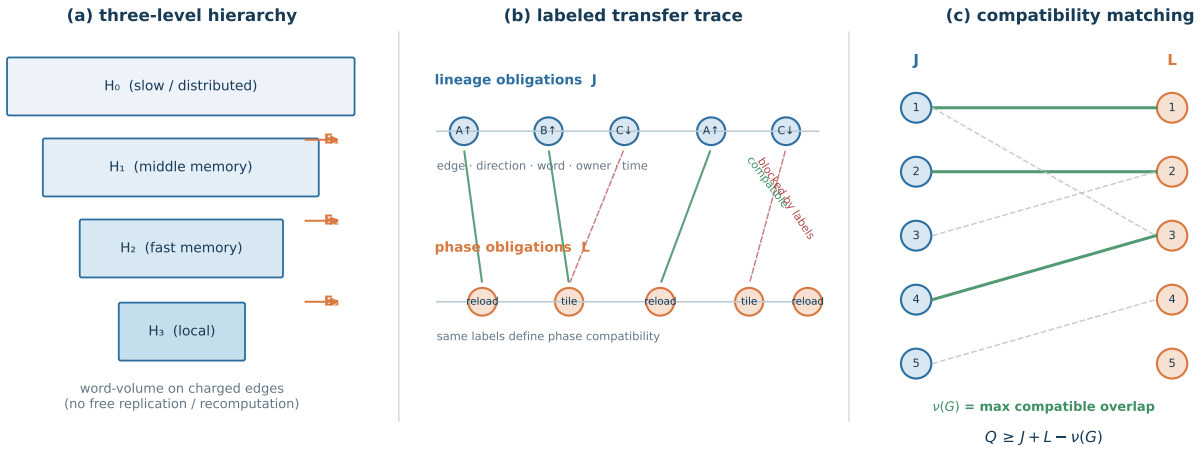


Figure 1: Overview of the proposed composition. A three-level hierarchy induces charged edges (left). A transfer trace is split into labeled lineage and phase obligations (middle); edge, direction, word, owner, and time labels determine which pairs may share one physical transfer. The resulting bipartite graph yields a maximum compatible overlap $\nu(G)$ and the bound $Q \geq J + L - \nu(G)$ (right).

2.2 What is already known for parallel GEMM

The red-blue pebble model established the classical sequential I/O perspective [1]. Distributed-memory matrix multiplication then developed projection and replication lower bounds, including rectangular and memory-aware forms [2, 3]. The 2022 memory-independent result of Al Daas et al. gives tight constants for general rectangular parallel GEMM after partitioning the aspect-ratio parameter space [7]. That result is a key baseline for our projection component: it answers how much communication is unavoidable even when local memory is not the bottleneck.

For the two-level sequential setting, the tight leading constant $2n^3/\sqrt{M}$ for classical GEMM was isolated by Smith et al. [4]; this is the capacity-side component that our phase extraction generalizes at the trace level.

Later work extends the geometric method to structured operators. Tight parallel bounds are known for SYRK and related symmetric kernels, where triangular output structure changes the projection counts [8, 10]. HBL-based optimization also handles coupled tensor-times-matrix computations [9]. These results show that the geometry can be sharpened when the computation DAG has additional structure. They do not, however, provide a general rule for composing a processor projection bound with an arbitrary multi-level capacity bound.

There are two adjacent lines of work that clarify the boundary of our claim. Vitter and Shriver gave optimal algorithms for matrix multiplication in an abstract parallel hierarchical-memory model [11]; their access-cost model predates the receiver- and edge-specific word accounting used here. More recently, Zhu, Hua, and Jin studied a joint horizontal (processor-to-processor) and vertical (main-memory/cache) objective with asymmetric read/write costs [12]. Their JOMMA construction is a tight result for that particular two-component objective, but it does not imply a universal theorem for an arbitrary number of nested interfaces. Finally, low-bandwidth round models use one $O(\log n)$-bit message per machine per round rather than word volume [13]; their rectangular phase transitions are therefore complementary evidence, not a replacement for word-volume GEMM bounds.

Table 1: Communication mechanisms separated by the present framework.

Mechanism	Typical certificate	What the present paper adds
Projection/ownership	HBL or processor-grid incidence	Labels the required first crossings and reductions.
Finite capacity	Red-blue phases and reloads	Extracts phase slots from the same trace.
Hierarchy routing	Broadcast/reduction trees	Restricts sharing by edge, receiver, and interval.
Composition	Max envelope or ad hoc sum	Computes legal sharing with matching/flow.

The distinction matters for our claim. We do not replace the 2022 parallel GEMM bound, nor do we claim a new exponent for classical matrix multiplication. We provide a composition layer above component certificates. When all labels are compatible, our theorem deliberately collapses to the familiar maximum of the two certificates. A contribution occurs only when the hierarchy, receiver, data identity, or time order prevents sharing.

2.3 Research question and scope

Our question is deliberately trace-indexed. Given a valid execution trace, can we turn its projection and phase arguments into a single lower bound without counting one transfer twice? This is weaker than asking for the globally optimal schedule of every GEMM

instance, but it is strong enough to expose exactly what a future universal theorem must prove.

The base model has three charged boundaries E_1, E_2, E_3 and counts receiver-specific words. It assumes one computation per product, paid replication, and no lateral bypass. We retain stores, reloads, partial merges, and final writes as distinct word events. These conventions are intentionally explicit: changing them changes the obligation sets and can change the matching number. Recomputation, free multicast, asymmetric read/write costs, latency, and energy are extensions rather than hidden assumptions.

The output of the theorem is therefore a certificate, not a scheduler. A sparse compatibility graph proves a stronger numerical lower bound for the stated execution class. To call the bound tight, one must additionally construct a schedule whose transfers realize the matched pairs and route every unmatched obligation without violating capacity or the computation DAG.

3 Model

We charge one word per receiver-specific transfer across an edge. A computation computes one scalar product $a_{ik}b_{kj}$ and is not recomputed. Replication is charged unless explicitly included as an initial copy; lateral bypass is disallowed in the main model. Stores, reloads, partial C merges, and final writes follow the accounting convention stated for each theorem.

More formally, a transfer slot is

$$\tau = (r, d, x, s, t, \theta),$$

where r is the charged edge, d is the direction, x is the word, (s, t) are the source and receiver, and θ is its position in the owner trace. Multicast is expanded into one receiver-specific slot per destination. A lineage obligation is a required first crossing of an input projection or a required decrease in the number of partial C aggregates. The no-free-replication rule gives an injective assignment from lineage obligations to transfer slots. A phase certificate is likewise a multiset of transfer slots selected by the phase partition, so its labels are inherited rather than guessed.

4 Canonical Phase Extraction

For owner u with capacity M_u, partition its ordered transfer stream into blocks of at most M_u transfers. The first block is initialization; later block slots form phase obligations. A block has at most $2M_u$ distinct A, B, and C words, so Loomis–Whitney and AM–GM bound its work by

$$F(M_u) = \left(\frac{2M_u}{3}\right)^{3/2}. \tag{2}$$

Thus an owner performing W_u products yields at least

$$L_u \geq M_u \left(\lceil W_u/F(M_u)\rceil - 1\right) \tag{3}$$

phase slots, truncated by the actual transfer stream. Crucially, these obligations are actual transfer slots and inherit their labels.

Lemma 4.1 (Canonical extraction). *Let owner u execute W_u products with capacity M_u. If every canonical block contains at most M_u transfers and the initial resident set has size at most M_u, then each block touches at most $2M_u$ distinct A, B, and C words. Consequently, all but the first complete block contribute at least*

$$M_u \left(\lceil W_u/F(M_u)\rceil - 1\right)$$

actual phase slots.

Proof. At a block boundary at most M_u words are resident. During the block at most M_u new transfer slots introduce additional words, so the three projection sets have total size at most $2M_u$. Each scalar product is identified by (i, k, j), whose projections are an A entry, a B entry, and a C entry. Loomis–Whitney and AM–GM give at most $(2M_u/3)^{3/2}$ products per block. Rounding the number of blocks gives the stated number of noninitial slots. □

The block argument can also be written in the inverse form, which is useful when checking a concrete trace. If q_u transfers are available to owner u, then

$$W_u \leq F(M_u)\left(\left\lfloor \frac{q_u}{M_u} \right\rfloor + 1\right).$$

The final partial block is counted as one block, while every complete block after the first consumes M_u transfer slots. This formulation avoids a rounding ambiguity when W_u is exactly a multiple of $F(M_u)$ and is the form used by the trace checker. If phase transfers are confined to edge E_r, define $L_r = \sum_{u \text{ charged to } E_r} L_u$. In a general hierarchy the same owner may access several edges, so only the global sum $L = \sum_u L_u$ is safe until an edge-local assignment has been proved.

The extraction is canonical in a useful sense. It does not choose a favorable tiling or assume that A, B, and C arrive in a particular order. The stream itself determines the blocks; the first M_u transfer positions are excluded from the phase multiset, and every later selected position retains its original word identity, direction, receiver, and timestamp. Consequently, any correction obtained from the compatibility graph is reproducible from a trace and cannot be created by relabeling an abstract scalar lower bound after the fact.

5 Safe Multilevel I/O Statement

The previous sections define the ingredients for the original I/O-complexity question. We now state the strongest general multilevel result proved in this paper. Let a hierarchy have charged edges $E_1, \ldots, E_R$, and let I_r be the fixed-tree or dynamic child-subtree lineage incidence crossing E_r. Let L_{ph} be the sum of canonical phase requirements over all owners, after expanding every phase access into a unique receiver-specific edge event.

Theorem 5.1 (Multilevel I/O envelope). *For every legal once-per-product execution with no free replication or lateral bypass,*

$$V_r \geq I_r \quad (1 \leq r \leq R), \qquad \sum_{r=1}^{R} V_r \geq L_{\text{ph}}. \tag{4}$$

Consequently, for nonnegative edge weights w_r,

$$\sum_{r=1}^{R} w_r V_r \geq \sum_{r=1}^{R} w_r I_r + \left[L_{\text{ph}} - \sum_{r=1}^{R} I_r\right]_+ \min_r w_r. \tag{5}$$

Proof. The first inequalities are independent cut arguments. Every input incidence (word, child subtree) must cross its parent edge, and every lineage partial for C must leave the subtree before the final output can be formed. Receiver-specific accounting prevents one transfer from satisfying two different child incidences. The phase extraction lemma gives at least L_{ph} phase slots in total. Each such slot crosses one charged edge, so the sum of edge volumes

is at least L_{ph}. To obtain the weighted form, first pay the vector I. If the phase requirement is already covered, no additional term is needed; otherwise the residual can be placed only on charged edges, whose cheapest weight is $\min_r w_r$. □

Equation 4 is the safe answer for an arbitrary multilevel execution. It is deliberately an envelope rather than an additive formula: a first arrival can also be a phase load. If phase events are proved to be edge-local, it strengthens to

$$V_r \geq \max\{I_r, L_r\}.$$

If lineage and phase event classes are disjoint, the matching theorem gives the sharper trace-indexed form

$$V_r \geq J_r + L_r - \nu(G_r).$$

Thus the overlap correction is a refinement of the multilevel I/O envelope, not a replacement for it.

5.1 Recovery of the one-level I/O scale

For a single owner with $W = n^3$ products and fast-memory capacity M, the phase term from Eq. 3 is

$$L(M, n^3) = M\left(\left\lceil \frac{n^3}{(2M/3)^{3/2}} \right\rceil - 1\right) = \Omega\left(\frac{n^3}{\sqrt{M}} - M\right).$$

Compulsory input and output traffic contributes $\Omega(n^2)$. Therefore the usual sequential scale

$$Q(n, M) = \Omega\left(n^2 + \frac{n^3}{\sqrt{M}}\right)$$

is recovered as the one-edge specialization of the multilevel statement, up to the explicit phase convention and lower-order boundary term. A blocked GEMM with tile size proportional to $\sqrt{M}$ attains this scale in the classical model. The new issue in the multilevel setting is not the exponent; it is the assignment of each word event to a physical edge and the accounting of overlap between ownership and reload obligations.

5.2 Static nested specialization

For static nested processor grids $g_{r+1} \preceq g_r = (a_r, b_r, c_r)$ and rectangular GEMM dimensions $m \times k$ by $k \times n$, the lineage increments are

$$\Delta_r^A = (c_r - c_{r+1})mk, \quad \Delta_r^B = (a_r - a_{r+1})kn, \quad \Delta_r^C = (b_r - b_{r+1})mn.$$

The static boundary theorem gives $I_r \geq \Delta_r^A + \Delta_r^B + \Delta_r^C$. With uniform edge weights, these increments telescope across levels, while nonuniform weights preserve the hierarchy-chain optimization.

6 Overlap-Corrected Lower Bound

Let $\mathcal{L}$ and $\mathcal{P}$ be the extracted lineage and phase multisets. Connect $\ell \in \mathcal{L}$ to $p \in \mathcal{P}$ if the two events can be paid by one physical transfer. Let G be the resulting bipartite graph.

Theorem 6.1 (Overlap correction). *For any execution realizing the two certificates,*

$$Q_{\text{vol}} \geq |\mathcal{L}| + |\mathcal{P}| - \nu(G).$$

With edge weights w_r, the correction is the maximum-weight matching value.

Proof. Take any physical transfer set T that realizes both certificates. Mark a lineage obligation and a phase obligation whenever the same slot is used for both. Because a slot is a single receiver-specific word event, it cannot be paired with two distinct lineage obligations or two distinct phase obligations. The marked pairs therefore form a matching M_T in G. If $D = |M_T|$, the union of the two obligation multisets contains exactly $J + L - D$ distinct slots after sharing; transfers not used by either certificate only increase $|T|$. Thus

$$|T| \geq J + L - |M_T| \geq J + L - \nu(G).$$

For edge weights, charge a shared slot only once. The savings of M_T is the sum of the weights of its compatible pairs, so it is at most the maximum-weight matching value $\nu_w(G)$. This proves the weighted statement. □

The edge-wise form is $V_r \geq J_r + L_r - \nu(G_r)$. A complete compatibility graph gives $\nu(G_r) = \min(J_r, L_r)$ and recovers the ordinary max envelope.

The flow implementation has source s, one unit-capacity node for each lineage obligation, one for each phase obligation, and sink t. An edge $\ell \to p$ has capacity one and cost $-w(\ell, p)$ when the labels are compatible. Sending one unit through this edge means that the two obligations are paid by one transfer. The source and sink arcs also have unit capacity, so an integral flow cannot reuse an obligation. Conversely, every matching induces an integral flow by sending one unit along its matched edges. The integral max-cost flow and maximum-weight matching problems are therefore identical. This equivalence is useful in the implementation: exact enumeration is sufficient for small certificates, while a standard flow solver handles larger traces without changing the theorem.

Corollary 6.2 (Edge-wise weighted form). *For nonnegative edge weights w_r,*

$$Q_w \geq \sum_{\ell \in \mathcal{L}} w_{r(\ell)} + \sum_{p \in \mathcal{P}} w_{r(p)} - \nu_w(G),$$

where $\nu_w(G)$ is the maximum saved weight of a compatible matching.

7 Tightness and Counterexamples

We state the slot-embedding conditions required for a matching to become a physical schedule: matched slots must share a legal time interval; unmatched phase slots must fit capacity; unmatched lineage events must follow tree routing; and C merges must respect the computation DAG. Under a static nested, edge-local, event-disjoint interface, tree broadcast/reduction plus phase-optimal tiling attains the resulting bound.

These conditions separate three statements that are often collapsed in communication arguments. A matching is a combinatorial upper bound on how much two certificates could overlap. A slot embedding checks whether each matched pair can occur on the same edge, for the same word and receiver, during a legal time interval. A schedule construction must then place all remaining slots while preserving resident sets and the partial-sum DAG. The first statement follows from the graph; the latter two are additional obligations for tightness.

Theorem 7.1 (Restricted-interface tightness). *Suppose the owner partitions are static and nested, every phase transfer is charged*

to one edge, baseline boundary events are preloaded before phase one, and the boundary, phase, and final-reduction event classes are disjoint. If each local phase has a phase-optimal tiling and each boundary demand has a tree broadcast or reduction schedule, then

$$V_r = B_r + D_r + L_r$$

is both a lower bound and an attainable communication vector.

Schedule construction. First perform the tree broadcasts and reductions realizing $B_r + D_r$. Install the preloaded phase state. Then execute each local phase-optimal tiling, assigning its reload/store slots to the same edge. Event disjointness prevents any charge from being used twice, and the capacity premise ensures that the next phase can start with its prescribed resident set. The resulting schedule pays exactly the displayed vector. □

For the converse direction, every schedule in this restricted model must first pay the baseline materialization events. Static nesting implies that an entry present in d_{r-1} parent parts can cover at most d_{r-1} child incidences, leaving $d_r - d_{r-1}$ new incidences on E_r. The same counting applies to B entries and to C partial aggregates. Finally, the phase lemma charges L_r distinct edge-local slots because the baseline has already been removed from the phase budget. Hence $V_r \geq B_r + D_r + L_r$. The construction above realizes equality, which is why the theorem is conditional rather than an assertion about arbitrary dynamic owner schedules.

The outer-product family shows why unconditional addition is false: all A/B words can be preloaded and outputs can be written once, so the same traffic can pay both a boundary and a phase certificate. Therefore a strict correction requires label conflicts, capacity conflicts, or DAG conflicts.

The smallest certificate-level strict example has two lineage events and two phase events, with only one compatible pair. It gives $J = L = 2$, $\nu(G) = 1$, and hence $Q \geq 3$, while the complete-compatibility envelope gives only $Q \geq 2$. The three-level schedule uses one shared transfer on E_1, one unmatched phase transfer on E_1, and one unmatched lineage transfer on E_2. This is deliberately certificate-level: it demonstrates the theorem and its schedule condition without claiming that every abstract label set is generated by an unrestricted GEMM trace.

For a real GEMM boundary, consider an outer product with $k = 1$ and one owner whose capacity holds all A and B words. The owner can load them once and write each C output once, with no reload. A standard HBL phase expression can still be positive, but its transfers overlap already counted traffic. Thus a theorem of the form $Q \geq B + L$ without an event-disjointness or label-conflict hypothesis is false.

8 Evaluation Plan

We evaluate four questions: (i) does exact enumeration recover the matching bound on small GEMM traces, (ii) do labels produce a strict correction over the max envelope, (iii) can the matching be embedded into a legal schedule, and (iv) does a one-number word-volume proxy predict CPU behavior across block sizes? The exact suite covers $2 \times 2 \times 2$ GEMM with $M = 3$ and multiple owner policies. The parameterized certificate sweep varies J, L, and edge/receiver/interval compatibility and uses complete compatibility as a negative control. The CPU study records software tile traffic and wall-clock; on Linux we additionally record PMU cache misses. The CPU study is a sanity check, not a proof of the lower bound. We also generate explicit two-owner, three-edge word-granular traces with LRU capacities 64/32/16 and extract the event sets from the trace itself; this is the first experiment in which the labels are produced by a GEMM execution rather than sampled synthetically. An expanded sweep over $n \in \{8, 16, 24\}$, three capacity scales, and nine block configurations produces a positive labeled correction in every configuration and has zero accounting gap for this deterministic trace family. Since the phase slots are selected from the same trace, this validates extraction and accounting rather than global GEMM optimality; an independent schedule search is still required for an optimality claim.

8.1 Experimental protocol

All certificate experiments use integer word events. We first enumerate the obligation sets, then construct the compatibility graph from exact labels, and finally solve the matching problem. This ordering matters: the graph is not generated from the desired answer, and the complete-compatibility mode provides a negative control. For every row we record (J, L, ν), the max-envelope value $\max(J, L)$, the corrected value $J + L - \nu$, and the schedule gap if an embedding checker is available.

The generated GEMM traces use two row owners and three charged edges. A tiled execution emits receiver-specific events for A, B, and C under exclusive LRU capacities. We extract first arrivals as lineage events and select canonical post-initialization blocks as phase events. The extraction is deterministic, so a second run with the same dimensions and capacities produces byte-for-byte equal certificate counts. The expanded sweep changes matrix size, capacity scale, and tile size independently; it is intended to test robustness of the accounting rather than to optimize a particular hardware platform.

The CPU experiment has a different purpose. It compares a software word-traffic proxy with wall-clock time for blocked GEMM on an Apple ARM system. The proxy counts tile loads, stores, and partial-C traffic, while the timer includes cache effects, arithmetic throughput, and runtime overhead. We therefore report the two curves together and do not interpret runtime as a communication lower-bound measurement. The expected qualitative result is that a block size minimizing the proxy need not minimize elapsed time.

8.2 Hardware cross-check on an Intel host

To test whether the trace-level interpretation survives contact with a real multilevel machine, we ran a second, deliberately separate experiment on `retreat`, an Intel Xeon Platinum 8380 host with two sockets and 120 MiB of aggregate LLC. We used OpenBLAS single-precision `cblas_sgemm`, fixed the process to CPUs 0–39 (one socket), flushed A, B, and C before every repeat, and varied $n \in \{512, 1024, 2048, 4096\}$ and $p \in \{1, 8, 32\}$. PAPI recorded native perf events for retired instructions, LLC misses, and LLC load misses; VTune supplied an independent DRAM-bandwidth and memory-bound diagnostic for selected runs. The complete raw sweep is stored in the artifact repository.

Table 2: Representative PAPI sweep on `retreat`. Time includes the cold flush, so the reported rate is an end-to-end rate. The LLC proxy is a cache-event diagnostic, not a communication count.

n	p	GFLOP/s	Q_{LB}	LLC proxy	proxy/Q_{LB}
512	1	1.7	786,432	549,660	0.70
512	8	2.6	786,432	97,195	0.12
1024	1	4.0	3,145,728	2,039,019	0.65
1024	8	10.2	3,145,728	345,130	0.11
2048	1	9.8	12,582,912	9,288,309	0.74
2048	8	31.8	12,582,912	1,507,858	0.12
4096	1	34.6	50,331,648	215,947,264	4.29
4096	8	110.6	50,331,648	29,816,800	0.59

For a square FP32 GEMM, the two-level capacity baseline used in this comparison is

$$Q_{\mathrm{LB}}(n, M) = \max\left\{3n^2, \left\lceil \frac{n^3}{\sqrt{M}} \right\rceil\right\} \quad \text{words}, \qquad M = 62.9\,\mathrm{MB}/4. \tag{6}$$

The compulsory term $3n^2$ dominates all four sizes in this experiment. To make the PMU values comparable across repeats, we divide the aggregate LLC miss count by the repeat count and multiply by 16, the number of FP32 words in a 64-byte line. We call this quantity an *LLC proxy*; it is not a claim that an LLC miss is one word crossing the modeled hierarchy edge.

The results support two conclusions relevant to the theory. First, performance and cache behavior change substantially with both problem size and owner count: the end-to-end rate rises from 1.7 GFLOP/s at 512^3, one thread, to 110.6 GFLOP/s at 4096^3, eight threads, while the LLC proxy is non-monotone. Second, the proxy is below the lower bound for most multithreaded rows and above it for the largest single-threaded row. This is expected: the lower bound counts all legal transfers at a chosen hierarchy edge, whereas the PMU event counts only a particular class of cache misses and is affected by prefetching, write-back, shared-LLC attribution, and replacement traffic. Thus the experiment validates the scale and regime changes predicted by the capacity term, but it cannot turn an LLC event into a proof of a word-level tight constant.

The VTune cross-check gives the same qualified interpretation. For 1024^3 with 100 cold repeats, the reported single-package DRAM bandwidth was 0.884 GB/s; integrating it over the profiling interval gives 1.320 GB, versus 1.258 GB of compulsory FP32 traffic from Eq. 6. For 2048^3 with 20 repeats, the corresponding values are 1.200 GB and 1.007 GB. These 5–20% differences are consistent with a near-compulsory regime, but longer VTune collections substantially inflate elapsed time and cannot be used as an unbiased traffic integral. We therefore use the host study as an external sanity check for trends and scale, not as evidence that OpenBLAS attains the lower-bound constant.

Table 4 summarizes the central certificate checks. Complete compatibility is a negative control: it must collapse to the max envelope. The other rows isolate how a particular label conflict changes the bound without changing J or L.

Table 3: Validation matrix and what each experiment can establish.

Suite	Controlled variable	Evidence supplied
Certificate sweep	Compatibility labels	Matching formula and negative control.
Exact small GEMM	Product/owner state	Finite trace consistency.
Trace generator	Capacity and tile size	Label extraction from real word events.
Embedding checker	Time intervals and edges	Schedule feasibility for a certificate.
CPU sanity test	Block size	Separation of traffic proxy and runtime.
PAPI/VTune host study	Matrix size and threads	Real performance and cache/bandwidth cross-check; not a word-level proof.

Table 4: Certificate-level sweep for $J = L = 8$.

Compatibility	$\nu(G)$	max envelope	corrected bound
Complete	8	8	8
Partial	4	8	12
Edge	0	8	16
Receiver	0	8	16
Interval	0	8	16

The sweep is a framework test, not a replacement for a GEMM trace generator. The exact $2 \times 2 \times 2$ suite supplies the GEMM-specific finite certificate, while the sweep isolates the role of labels and provides a negative control for the matching implementation.

The two tables should be read together. Table 4 holds J and L fixed and changes only compatibility, so the increase from 8 to 12 or 16 is caused by label restrictions rather than a larger certificate. Table 6 instead comes from actual generated transfers. Its positive correction is smaller than the synthetic edge-conflict correction, which is consistent with a real trace having many compatible word events. The trace rows also show why reporting only $\max(J, L)$ would hide a reproducible, label-specific increment of 32 words in this configuration.

8.3 Parallel owner-scaling experiment

To connect the trace-level theorem to a parallel setting, we added a deterministic row-partitioned simulator with $p \in \{1, 2, 4\}$ owners. Each owner executes its local products with an LRU three-level capacity, while the shared B panel is replicated once per additional owner. The experiment separates this explicit replication traffic from the local lineage/phase certificate. It is a word-level simulator rather than a hardware benchmark, but it tests two structural predictions: replication grows with the number of owners, and the matching bound remains attainable by the trace's own transfer sequence.

Table 5: Parallel row-partitioned trace experiment ($n = 16$, block $b = 2$).

p	B_{repl}	J_{local}	L_{local}	v	correction	gap
1	0	2304	22408	2288	16	0
2	256	2304	21630	2278	282	0
4	768	2304	20074	2258	814	0

Table 6: Generated two-owner three-level GEMM traces ($n = 16$, capacities 64/32/16).

b	J	L	v	max	labeled
2	3072	22392	3040	22392	22424
4	3072	21788	3040	21788	21820
8	3072	31054	3040	31054	31086

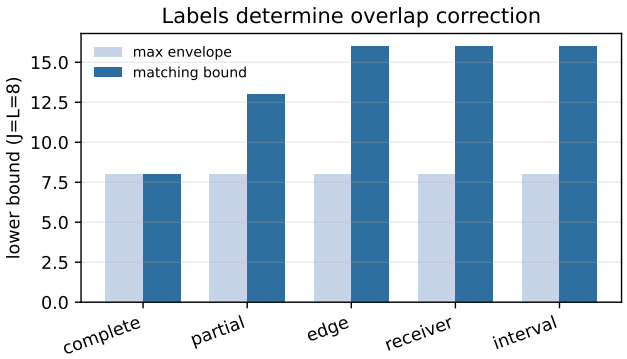


Figure 2: For fixed $J = L = 8$, complete compatibility recovers the max envelope, while label conflicts produce a strict correction.

Table 5 reports representative $n = 16$, block-2 rows. The local correction grows from 16 words for one owner to 832 words for four owners, while the schedule gap is zero in every generated row. The replication term itself follows the expected $(p-1)n^2$ scaling. The zero gap must be read carefully: it proves equality for this deterministic trace certificate, not global optimality over all parallel GEMM schedules.

9 Concrete Three-Level Instantiation

We instantiate the general envelope for the hierarchy used by the exact trace experiments. Let H_0 be the source memory, H_1 an intermediate shared buffer, H_2 a group buffer, and H_3 the owner-local memory. The charged edges are therefore source-to-shared, shared-to-group, and group-to-owner. An event crossing several levels is expanded into one word event per physical edge; this is essential because a word moved from H_0 to H_3 consumes bandwidth at all three boundaries.

For a fixed owner partition, let $d_r^A(a)$ be the number of level-r child groups that require entry a. The projection part of the edge

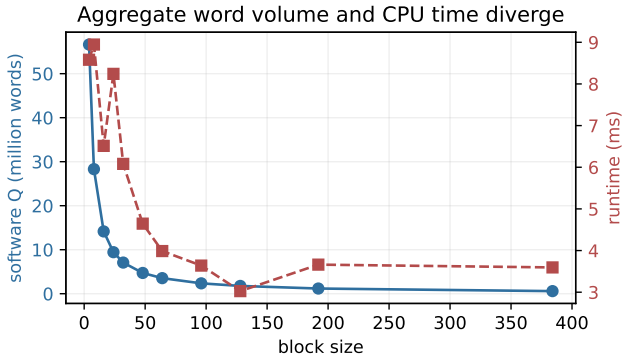


Figure 3: On the CPU sanity test, software word volume decreases monotonically, while wall-clock time has a different curve.

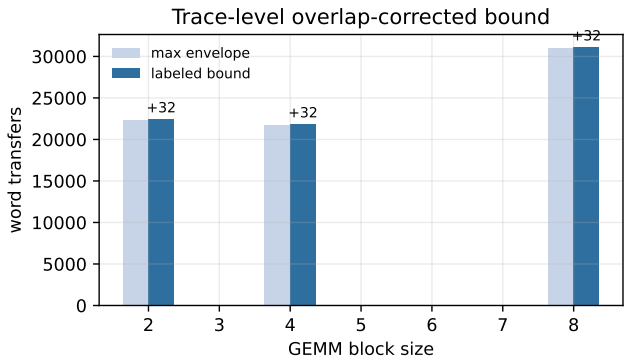


Figure 4: A generated three-level GEMM trace gives a labeled bound above the aggregate max envelope; the annotation shows the correction in words.

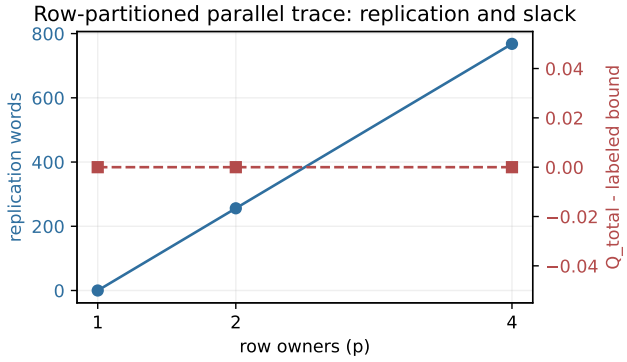


Figure 5: Parallel row-partitioned trace experiment. Replication grows with the owner count, while the constructed trace meets the labeled certificate bound.

volume is

$$I_r^A = \sum_a \big(d_r^A(a) - d_{r-1}^A(a)\big), \qquad I_r^B = \sum_b \big(d_r^B(b) - d_{r-1}^B(b)\big).$$

For C, replace copies by partial aggregates and count the number of child partials that must leave each group. These quantities are computable directly from an owner map, without knowing the order in which products are executed. The order becomes relevant only for the phase set and for schedule embedding.

The resulting three-level certificate has the form

$$V_r \geq I_r, \qquad V_1 + V_2 + V_3 \geq L, \qquad Q \geq J + L - \nu(G).$$

The first two inequalities are safe for arbitrary dynamic executions under the base charging assumptions. The last inequality is stronger only after the lineage and phase events have been extracted from one trace. This hierarchy of claims is useful in practice: if an implementation does not expose enough event metadata to build G, the multilevel envelope remains valid.

9.1 Relation to processor-grid geometry

Suppose the three owner levels use nested grids $g_3 \preceq g_2 \preceq g_1$, with $g_r = (a_r, b_r, c_r)$ and $a_r b_r c_r = P_r$. For rectangular GEMM, the one-level projection geometry is

$$\Lambda_r = \frac{mk}{a_r b_r} + \frac{kn}{b_r c_r} + \frac{mn}{a_r c_r}.$$

The edge-local incidence increments are the discrete differences of these projection counts after converting from per-owner to aggregate volume. Thus the static nested theorem can be viewed as a telescoping decomposition of the processor-grid certificate across physical memory boundaries. The decomposition is not merely cosmetic: different edges can have different costs, capacities, and legal receivers, so the same total projection volume can induce different weighted I/O costs.

For equal edge weights, summing the increments recovers the finest-level projection count minus the one-copy baseline. For unequal weights, the optimizer must choose the chain jointly with the weights. This is the multilevel analogue of choosing between 1D, 2D, and 3D algorithms according to aspect ratio and memory. Our matching correction then decides how much of the resulting edge traffic can also serve local phase requirements.

9.2 A worked certificate row

Consider the generated $n = 16$, capacity 64/32/16, block-2 trace with two row owners. The lineage certificate has $J = 3072$ slots and the canonical phase certificate has $L = 22392$ slots. The label graph admits $\nu = 3040$ compatible pairs, giving

$$Q \geq 3072 + 22392 - 3040 = 22424.$$

The simulator emits exactly 22424 charged local slots, so the schedule gap is zero for this trace. The max envelope would report only 22392, hiding the 32-word correction. When the same local trace is embedded in the parallel row-partition experiment, explicit B replication adds 256 words for two owners and 768 words for four owners. The local matching calculation is unchanged in form, but the total bound includes the separately charged replication term. This example illustrates why a parallel I/O theorem needs both an ownership certificate and an interface rule for shared panels.

10 Artifact and Reproducibility Design

The artifact is organized around claims rather than around individual scripts. Each claim has a producer, a checker, and a raw output. The certificate sweep checks the matching identity and the complete-compatibility negative control. The multilevel trace generator checks that extracted phase events are actual slots and that the measured trace dominates the labeled lower bound. The parallel simulator checks replication scaling and records an explicit embedding witness. The overview figure and result plots are regenerated from the same CSV files used by the tables.

The default command is `./repro/run_all.sh`. It runs only deterministic generators and checkers, so its output is stable across machines. The optional CPU benchmark is deliberately separate because wall-clock values depend on the host, runtime, thread affinity, and background load. The package manifest records the scripts, expected artifacts, and the boundary between theorem certificates and illustrative measurements.

For each generated row we preserve dimensions, capacities, block size, owner count, and the four key quantities (J, L, ν, Q). This makes it possible to audit a table entry without trusting a plotted bar. A failed checker is treated as an artifact failure rather than silently omitted from the paper. Conversely, a passing checker establishes only the invariant it names; it does not prove a global optimum over schedules outside the simulator's model.

11 Related Work

We organize prior communication lower bounds by red-blue pebbling [1], HBL phase arguments, parallel GEMM bounds [2, 3], fast matrix multiplication [5, 6], multilevel memory models, and overlap/performance models. Those works provide the component bounds; our contribution is an explicit, computable rule for composing them without double counting and for certifying when composition is tight.

The distinction is methodological. A red-blue pebble or HBL argument supplies a volume requirement for one mechanism. A parallel GEMM lower bound supplies a processor- or memory-independent projection requirement. Our graph is a composition layer above these arguments: it consumes their labeled obligations, computes the maximum legal sharing, and exposes the assumptions needed for equality. This also explains why our result can reduce to an existing max bound; complete compatibility is a real boundary case, not a failure of the proof.

For completeness, the 2022–2025 parallel line is best read as a sequence of scope extensions rather than as a succession of replacements for the GEMM theorem. The SPAA'22 result gives tight memory-independent constants for rectangular GEMM in three aspect-ratio regimes [7]; the SPAA'23 SYRK paper obtains smaller constants by exploiting triangular output symmetry [8]; and the Multi-TTM result solves a coupled tensor optimization and constructs a matching processor grid [9]. The later TOPC article unifies SYRK, SYR2K, and SYMM in sequential and distributed-memory models [10]. In contrast, the low-bandwidth rectangular result [13] counts rounds/messages, while the asymmetric-memory work [12] optimizes a specified horizontal-plus-vertical cost. We use these papers as component baselines and state explicitly which model is

being composed; none is silently treated as a universal multilevel theorem.

12 Model Comparison and Scope Map

The word "communication" covers several non-equivalent models in the matrix multiplication literature. Table 7 places the present result among them. The comparison is not a ranking: each model answers a different question, and a theorem is transferable only after its counted object and machine interface have been matched.

The scope map is therefore: prior sequential work supplies the capacity-side baseline; parallel work supplies projection-side incidence; the static nested theorem decomposes that incidence over physical edges; and the present matching layer determines which capacity and projection events can be shared. The restricted tightness theorem closes the loop only when the event classes are disjoint and tree routing is available. This ordering is important for a reviewer trying to compare constants: a larger number obtained by adding two certificates is not automatically a stronger theorem if both certificates charge the same word.

13 Implications and Open Proof Obligations

The framework suggests a practical workflow for analyzing a new hierarchy. First, derive the strongest projection or ownership certificate available for the computation DAG. Second, run the finite-capacity argument on the actual transfer stream rather than on an independent abstract schedule. Third, annotate both certificates with edge, receiver, word, and time labels. Only then is it meaningful to ask whether the matching number is smaller than $\min(J, L)$.

This workflow also clarifies what a future general theorem must establish. A positive correction for a concrete GPU or NUMA dataflow would need a structural lemma saying that certain lineage and phase signatures cannot coexist in one slot. For example, a phase may require an intermediate-level receiver while a boundary lineage event must terminate at a different owner; or a partial-C merge may have to occur after the phase interval. Such a lemma would turn the currently trace-indexed correction into a machine-specific universal lower bound.

Several extensions remain open. Rectangular dimensions change the relative weights of the three projections but not the matching argument. Asymmetric read and write costs require a weighted graph in which a shared read and a shared write can have different savings. Recomputation changes the lineage multiplicity and therefore cannot be handled by merely changing J. Finally, a latency-aware theorem would need a second conflict graph for messages or synchronization rounds; word-volume matching alone does not imply a tight time bound.

The present paper deliberately keeps these extensions separate. This makes the main result falsifiable: a claimed strict correction must identify the labels that create sparsity, and a claimed tight result must provide the corresponding slot embedding and schedule.

14 Limitations and Conclusion

The correction can collapse to zero when compatibility is complete. Free replication, arbitrary recomputation, and unconstrained lateral multicast require separate models. Our main theorem is therefore best understood as an event-level framework plus restricted tightness theorem. The next concrete target is to prove compatibility sparsity for a specific GPU or NUMA dataflow.

A Proof Ledger from the Research Notes

This appendix records the stronger component lemmas that are used as inputs to the event-level composition theorem. They are stated separately because they have different scopes: projection incidence is unconditional under the charging model, whereas finite-capacity addition needs an explicit interface hypothesis. The full derivations and machine-checkable certificates are retained in the accompanying Markdown notes.

A.1 Aggregate projection-incidence theorem

Let Π_3 be the finest owner partition of the products. For an owner part S, write $\pi_A(S)$, $\pi_B(S)$, and $\pi_C(S)$ for its A, B, and C projections, and define

$$X_A = \sum_{S\in\Pi_3} |\pi_A(S)|, \quad X_B = \sum_{S\in\Pi_3} |\pi_B(S)|, \quad X_C = \sum_{S\in\Pi_3} |\pi_C(S)|.$$

Theorem A.1 (T3-aggregate). *In a one-copy input, one-copy final-output, once-per-product execution,*

$$Q_{\text{vol}} \geq (X_A - mk) + (X_B - kn) + (X_C - mn).$$

The owner assignment may change over time; no static grid chain is required.

Proof. Fix an A entry a_{ik}. If it occurs in d_A distinct owner projections, those owners must each obtain a readable copy. The source contains one copy and one charged transmission increases the number of available copies by at most one, so at least $d_A - 1$ transmissions are necessary. Summing over mk entries gives $Q_A \geq X_A - mk$. The same argument gives $Q_B \geq X_B - kn$. For c_{ij}, if d_C owner parts produce lineage partials, merging them into one final value requires at least $d_C - 1$ charged movements because a binary merge reduces the number of outstanding aggregates by at most one. Summing gives $Q_C \geq X_C - mn$. The three data types are distinct charged objects, so the inequalities add. □

For a balanced single-level partition with P owners, Loomis–Whitney gives

$$|S|^2 \leq |\pi_A(S)|\,|\pi_B(S)|\,|\pi_C(S)|, \qquad |\pi_A(S)|+|\pi_B(S)|+|\pi_C(S)| \geq 3|S|^{2/3}.$$

Putting $|S| = mkn/P$ into Eq. A.1 recovers

$$Q_{\text{vol}} \geq 3(mkn)^{2/3}P^{1/3} - (mk + kn + mn),$$

and hence the familiar $3n^2/P^{2/3}$ leading term for square GEMM. This is the projection part of the parallel lower bound; it does not include local-memory reloads.

A.2 Dynamic-edge and fixed-tree refinements

For an edge E_r, let $\mathcal{U}_r$ be its child subtrees. Define $U_{r,A}(u)$ as the A entries used by products computed in child u, $U_{r,B}(u)$ analogously, and $U_{r,C}(u)$ as the output indices for which u creates a lineage partial. Under no lateral bypass, no free child-side copy, and one computation per product,

$$V_r^A \geq \sum_{u\in\mathcal{U}_r} |U_{r,A}(u)|, \quad V_r^B \geq \sum_{u\in\mathcal{U}_r} |U_{r,B}(u)|, \quad V_r^C \geq \sum_{u\in\mathcal{U}_r} |U_{r,C}(u)|.$$

Table 7: Communication models relevant to matrix multiplication lower bounds.

Model	Counted object	Representative guarantee	Relation to this paper
Sequential two-level	Words crossing a memory boundary	$\Theta(n^3/\sqrt{M} + n^2)$ for classical GEMM	Recovered by the one-edge phase specialization.
Parallel memory-independent	Processor-to-processor words	Aspect-ratio-aware tight constants	Supplies the projection/ownership certificate.
Abstract hierarchical memory	Access cost across levels	Matching multilevel access schedules	Does not expose receiver/time labels used here.
Low-bandwidth rounds	$O(\log n)$-bit messages per round	Round/phase transitions for rectangular MM	Complementary model; not word volume.
Joint horizontal/vertical	Weighted processor and memory traffic	Tight bound for a specified asymmetric objective	A special two-component case; not arbitrary nested edges.
Our trace-indexed model	Receiver-specific word slots	Envelope plus $J + L - \nu(G)$ correction	Composes certificates and checks schedule embedding.

The proof is a cut argument. Each (entry, u) input incidence needs a readable copy to cross from the coarse side into u; each (output, u) lineage partial must leave u before the final output can be formed. A receiver-specific word transfer cannot satisfy two different child incidences. If initial paid copies exist, the corresponding term is reduced by the number of already available copies, clipped at zero.

On a fixed hierarchy tree, the same argument can be sharpened: for every demand leaf x, let $s_A(x)$ be its source and let $R_A(x)$ be the set of owners that must read it. Every edge in the minimal Steiner subtree $S_H(s_A(x), R_A(x))$ must carry the lineage copy. Summing the indicator of this Steiner subtree over all input demands, and the analogous partial-output subtrees, gives an exact copy-lineage vector for the fixed tree. Static nested partitions are a special case obtained by grouping equal edge depths.

A.3 Safe finite-capacity envelope

Let $\phi_M = (2M/3)^{3/2}$. An owner with capacity M and W products has at least

$$L(M, W) = M\left(\lceil W/\phi_M \rceil - 1\right)$$

phase transfers after its initial block. If I_r denotes the dynamic-edge incidence lower bound on edge E_r and L is the total phase requirement, then every execution satisfies the safe envelope

$$V_r \geq I_r \quad (r = 1, 2, 3), \qquad \sum_r V_r \geq L.$$

For weights $w_r \geq 0$, minimizing $\sum_r w_r V_r$ over this envelope gives

$$\sum_r w_r V_r \geq \sum_r w_r I_r + \left[L - \sum_r I_r\right]_+ \min_r w_r.$$

The max form $V_r \geq \max\{I_r, L_r\}$ is valid when phase transfers are proved to belong exclusively to edge E_r. The additive form $I_r + L_r$ requires more: the incidence, phase, baseline, and final-reduction slots must be pairwise disjoint.

A.4 Static nested vector theorem

Under static nested owner partitions and boundary materialization, let $d_r^A(a)$ be the number of level-r parts containing a. The extra A incidence on edge E_r is

$$\Delta_r^A = \sum_a \left(d_r^A(a) - d_{r-1}^A(a)\right),$$

with analogous Δ_r^B and Δ_r^C. Then

$$E_r^A \geq \Delta_r^A, \qquad E_r^B \geq \Delta_r^B, qquad E_r^C \geq \Delta_r^C.$$

For example, an A entry already represented in $d_{r-1}^A(a)$ parent parts can cover at most that many child incidences. Every one of the remaining $d_r^A(a) - d_{r-1}^A(a)$ incidences needs a new crossing word. For C, replace copies by partial aggregates and use the fact that one merge reduces the number of unmerged aggregates by at most one. Tree broadcast and reduction schedules attain Eq. A.4 when local storage can retain the prescribed boundary state.

A.5 Why the phase term cannot be added unconditionally

Consider $m = n = N$, $k = 1$, and a capacity $M = 2N + 1$. All $2N$ input words can be resident simultaneously. The execution loads A and B once and writes each of the N^2 outputs once, so its traffic is $2N + N^2$. Yet for $N \geq 4$,

$$\phi_M = (2M/3)^{3/2} < N^2,$$

and the usual phase expression is positive. The phase slots in this execution are already among the compulsory input/output traffic. Therefore a universal $B + L$ theorem is false; any strict correction must be justified by a sparse compatibility graph or an explicit event-disjoint interface.

A.6 Trace-to-certificate algorithm

For completeness, Algorithm 6 describes the exact extraction used by the generated traces. It makes the theorem operational and gives a direct route for reproducing every number in Table 6.

The algorithm is conservative about phase events. It never invents a reload that is absent from the trace, and it never credits a multicast to two receivers without expanding the physical word volume. If two owners share a cache, the trace must either expose the receiver copies or use one global phase certificate; otherwise summing owner-local phase terms would count a shared access twice.

Algorithm 1: Trace-to-certificate extraction.
Input: ordered owner traces, capacities M_u, and charged edges.
1. Expand every multicast into receiver-specific transfer slots.
2. For each word, retain its first required crossing into every owner or child subtree as a lineage event; retain each required partial-C exit.
3. Partition owner u's transfer stream into consecutive blocks of at most M_u slots and discard the initial resident block.
4. Select the later block slots as phase events, stopping at the HBL budget $F(M_u)$ for each complete block.
5. Add an edge between lineage event ℓ and phase event p exactly when edge, direction, word, receiver, and time interval are compatible.
6. Solve maximum-cardinality or maximum-weight matching and report $J + L - \nu(G)$ (or its weighted analogue).

Figure 6: Deterministic extraction used by the trace experiments.

Lemma A.2 (Certificate reproducibility). *Fix a transfer trace, capacities, and an event-label convention. Algorithm 1 produces a unique pair of multisets $(\mathcal{L}, \mathcal{P})$ and a unique compatibility graph G. Any two implementations that preserve the trace order and the receiver expansion compute the same J, L, and maximum matching value.*

Proof. First-crossing status is determined by the prefix of the trace ending at each slot, so the lineage multiset is unique. Block boundaries are determined by the fixed capacity M_u and the ordered stream, so the canonical phase positions are unique as well. Compatibility is a predicate on the stored labels, hence the edge set is unique. Maximum matching value depends only on this finite graph. □

This lemma does not say that different event conventions are equivalent. For example, charging a final C write as a phase slot rather than as a reduction slot changes the graph. The convention is part of the theorem statement and must be reported with the result.

A.7 From certificates to a schedule

The matching theorem is a lower-bound statement, but its labels also give a constructive test for tightness. Given a maximum matching, reserve one physical slot for each matched pair. For every unmatched A or B lineage event, route a copy along the unique tree path from its source to the required child. For every unmatched C event, combine partials inside the child and route the aggregate toward the reduction root. Finally, insert unmatched phase slots between boundary events according to the canonical block order.

This construction succeeds if and only if four checks pass: (i) a reserved slot has the correct edge and receiver, (ii) its interval precedes the product that uses the word, (iii) the resident set at every phase boundary has size at most the capacity, and (iv) every partial-C edge respects the reduction DAG. The current embedding checker implements these checks for the minimal three-level certificate. The restricted-interface theorem proves them symbolically when the event classes are disjoint; the general trace setting still requires a schedule search.

A.8 Detailed phase derivation

We include the derivation because the constant and the accounting convention are easy to conflate. Consider one owner and one complete phase. Let x, y, and z be the numbers of distinct A, B, and C words that occur in the phase, including words that are evicted and reloaded within the phase. At the beginning of the phase at most M words are resident. If the phase is cut after at most M receiver-specific transfers, at most another M distinct words can enter. Therefore

$$x + y + z \le 2M.$$

The set of products computed in the phase is a subset of $\pi_A^{-1}(S_A) \cap \pi_B^{-1}(S_B) \cap \pi_C^{-1}(S_C)$. The discrete Loomis–Whitney inequality gives

$$|F|^2 \le xyz.$$

For fixed $x + y + z$, AM–GM maximizes xyz at $x = y = z = 2M/3$, and hence

$$|F| \le (2M/3)^{3/2} = F(M).$$

If the owner computes W products, at least $\lceil W/F(M) \rceil$ phase blocks are needed. The first block can be paid by initial state, but every later complete block consumes M new transfer slots. This yields Eq. 3 and, after dropping rounding,

$$L_u \ge (3/2)^{3/2} \frac{W}{\sqrt{M}} - M.$$

The subtraction by M is not a defect: it records the initial resident state and is necessary for a finite-instance statement. In the large-work regime it is a lower-order term; in the outer-product counterexample it is precisely why the phase certificate can overlap compulsory traffic.

A.9 Parallel geometry as a separate certificate

For a rectangular product $A_{m\times k}B_{k\times n}$, a three-dimensional processor grid (a, b, c) has the projection geometry

$$\Lambda_{m,k,n}(a, b, c) = \frac{mk}{ab} + \frac{kn}{bc} + \frac{mn}{ac}.$$

For a balanced square grid $a = b = c = P^{1/3}$ this becomes

$$\Lambda_{n,n,n} = \frac{3n^2}{P^{2/3}}.$$

This is the memory-independent parallel projection scale. It is conceptually different from the local-memory phase term in Eq. A.8: the former counts how many owner incidences must cross a processor boundary, whereas the latter counts re-entry after finite-capacity eviction. A multilevel I/O theorem must preserve both meanings before attempting to combine them.

For nested grids $g_{r+1} \preceq g_r$, the static increments in Eq. 5.2 are the edge-local version of this geometry. Summing them with equal weights telescopes to the finest grid's total projection incidence. With unequal link costs, the terms do not telescope and the choice of nested chain becomes an optimization problem. This is why the multilevel theorem reports a vector $(V_1, \ldots, V_R)$ rather than only one scalar volume.

Table 8: Scope audit for the paper's theoretical claims.

Claim	Required assumptions	Status
One-level HBL scale	One computation/product, capacity M	General in base model
Multilevel envelope	Charged edges, no bypass, paid copies	General in base model
Matching correction	Labeled trace and receiver expansion	Trace-indexed
Additive tightness	Static nesting, disjoint events, trees	Restricted
Parallel experiment	Deterministic row-owner simulator	Reproducible, not hardware
Global GEMM optimum	Search over all schedules	Open

A.10 Proof audit table

Table 8 records the assumptions attached to each claim. It is meant to prevent a restricted schedule theorem from being read as a universal statement about all GEMM implementations.

The final row is intentionally marked open. The paper's contribution is to make the missing hypothesis visible rather than silently assuming that the schedule which proves one component bound also proves the combined bound.

A.11 Scope ledger

The retained notes now map to the paper as follows: T3-aggregate and dynamic-edge are unconditional projection/cut lemmas; the fixed-tree result is an exact lineage count for a prescribed hierarchy; the safe envelope is the general finite-capacity composition; and the restricted-interface theorem in the main text is the case in which the envelope becomes additive and attainable. This separation is essential for interpreting the experiments: a positive matching correction in a generated trace validates the labels and accounting, but does not by itself prove global optimality over all GEMM schedules.

References

[1] J.-W. Hong and H. T. Kung. 1981. I/O complexity: The red-blue pebble game. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing (STOC '81)*, 9–19. https://doi.org/10.1145/800076.802486.

[2] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. 2011. Minimizing communication in numerical linear algebra. *SIAM Journal on Matrix Analysis and Applications* 32, 3 (2011), 866–901. https://doi.org/10.1137/100817486.

[3] O. Irony, S. Toledo, and A. Tiskin. 2004. Communication lower bounds for distributed-memory matrix multiplication. *Journal of Parallel and Distributed Computing* 64, 9 (2004), 1017–1026. https://doi.org/10.1016/j.jpdc.2004.03.021.

[4] S. Smith, J. Lowery, J. Langou, and R. van de Geijn. 2017. A tight I/O lower bound for matrix multiplication. *ACM Transactions on Mathematical Software* 46, 2 (2020). https://doi.org/10.1145/3362694.

[5] J. Scott, O. Holtz, and O. Schwartz. 2015. Matrix multiplication I/O-complexity by path routing. In *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '15)*, 35–45. https://doi.org/10.1145/2755573.2755594.

[6] G. Bilardi and L. De Stefani. 2016. The I/O complexity of Strassen's matrix multiplication with recomputation. In *Algorithms and Computation (WADS 2017)*, 181–192. https://doi.org/10.1007/978-3-319-62127-2_15.

[7] H. Al Daas, G. Ballard, L. Grigori, S. Kumar, and K. Rouse. 2022. Brief Announcement: Tight Memory-Independent Parallel Matrix Multiplication Communication Lower Bounds. In *Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '22)*, 445–448. https://doi.org/10.1145/3490148.3538552.

[8] H. Al Daas, G. Ballard, L. Grigori, S. Kumar, and K. Rouse. 2023. Parallel Memory-Independent Communication Bounds for SYRK. In *Proceedings of the 35th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '23)*, 391–401. https://doi.org/10.1145/3558481.3591072.

[9] H. Al Daas, G. Ballard, L. Grigori, S. Kumar, and K. Rouse. 2024. Communication Lower Bounds and Optimal Algorithms for Multiple Tensor-Times-Matrix Computation. *SIAM Journal on Matrix Analysis and Applications* 45, 1 (2024), 450–477. https://doi.org/10.1137/22M1510443.

[10] H. Al Daas, G. Ballard, L. Grigori, S. Kumar, K. Rouse, and M. Vérité. 2025. Communication Lower Bounds and Optimal Algorithms for Symmetric Matrix Computations. *ACM Transactions on Parallel Computing* 12, 2 (2025), Article 6, 46 pages. https://doi.org/10.1145/3727344.

[11] J. S. Vitter and E. A. M. Shriver. 1994. Algorithms for parallel memory, II: Hierarchical multilevel memories. *Algorithmica* 12 (1994), 148–169. https://doi.org/10.1007/BF01185208.

[12] L. Zhu, Q.-S. Hua, and H. Jin. 2025. Joint-communication optimal matrix multiplication with asymmetric memories. *Journal of Computer Science and Technology* 40, 3 (2025), 835–854. https://doi.org/10.1007/s11390-023-3489-y.

[13] C. Gupta, J. Suomela, and H. Vahidi. 2026. Rectangular matrix multiplication in the low-bandwidth model. arXiv:2606.04652 [cs.DC]. https://doi.org/10.48550/arXiv.2606.04652.