Volume II Preliminaries

Deep Dive into SGLang — Volume II Preliminaries

Yin Yang

Foundation Books Project

Where this bridge sits

This is the front matter that opens Volume II: it hands you *one check* to carry into every optimization chapter.

- Volume I followed one online request from admission to cleanup.
- Volume II follows a *smaller* unit: one scheduled prefill or decode step, while the runtime rewires storage, placement, synchronization, execution, and measurement around it.
- A legal tensor shape is not enough — the step can still lose a row, expose KV too early, pick an ineligible kernel, or move the benchmark window.

Goal: for any Volume II change, name the object, its owner, its commit point, and the evidence that makes the result comparable.

Roadmap

The one check

The five axes

Reading map

The one check

One step, four questions

For every Volume II optimization, ask the same four questions:

1. **What** request row or runtime state is being transformed?
2. **Who** owns it?
3. **When** may another component observe it?
4. **What evidence** makes the result comparable?

This is a checklist, not a new notation layer. The machinery changes, but row identity, state ownership, visibility, and evidence boundaries still have to line up.

The carried objects: minimum background

Four objects survive every optimized path:

Request row one scheduled continuation at one logical position.

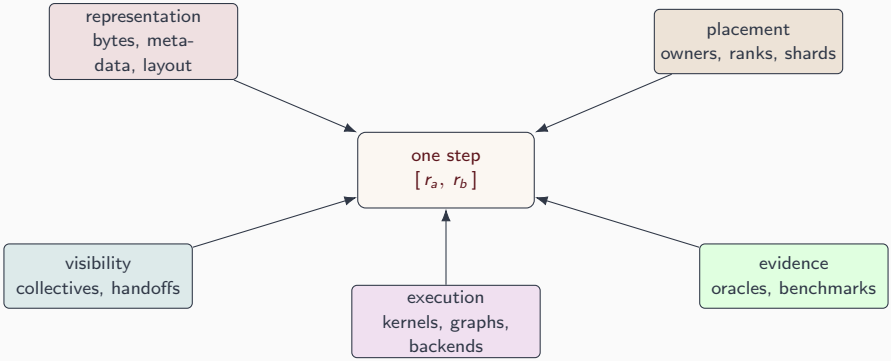
KV state the resident attention history that row may read and extend.

Forward-step boundary where selected rows become model execution metadata.

Output event the user-visible token, finish state, stream update, or non-text result.

Packing, routing, sharding, kernels, and fallbacks may change storage — not row identity or output ownership.

One serving step, five lenses



Five views attach to the *same* scheduled rows. The execution path changes; row identity and output order remain the reference.

The five axes

What each lens can silently break

Axis	Failure that looks valid
Representation	Same-shape bytes decode the *wrong operator* (misread scale grouping, layout, or expert records).
Placement	The split names no owner; local view never rejoins as the request-visible value.
Visibility	A shard, KV page, or routed row is consumed *before* its owner made it visible.
Execution	A kernel, graph bucket, or backend runs outside its envelope, or its fallback drops the row-aligned result.
Evidence	A number is compared under a different window, revision, or claim boundary than the text names.

Each axis gets one section — each states the check, then the failure to rule out.

Reading map

The reading map into Volume II

The bridge names failures; the owning chapters develop the machinery.

Runtime change	Owning chapters
Packed representation & layout	quantization, LoRA, MoE, attention/GEMM kernels.
Placement & visibility	parallelism, collectives, prefill/decode disaggregation, routing.
Fast execution & fallback	kernel design, attention/GEMM, CUDA graphs, hardware backends.
Modal state & non-text steps	multimodal serving, diffusion runtime, diffusion acceleration.
Evidence & extension	benchmarking, correctness testing, extending SGLang.

Use it as a dependency guide — each row specializes the same byte, owner, visibility, fast-path, and evidence failures.

What to carry forward

1. Volume II reasons about **one step**, not one request — the row is the unit of correctness.
2. Four questions gate every change: **object, owner, visibility, evidence**.
3. **Shape is not enough:** metadata, ownership, and commit points decide whether the step is still legal.
4. A **fallback is part of the contract** when it preserves the row-aligned result.
5. A claim is only as large as its **benchmark protocol** and **correctness ledger** allow.

Next: Volume II opens with quantization, LoRA, and MoE — each an optimized path that must still return the same rows this bridge protects.