Semantic-aware LLM-Application Scheduling

Otto Whitee3

December 10, 2025

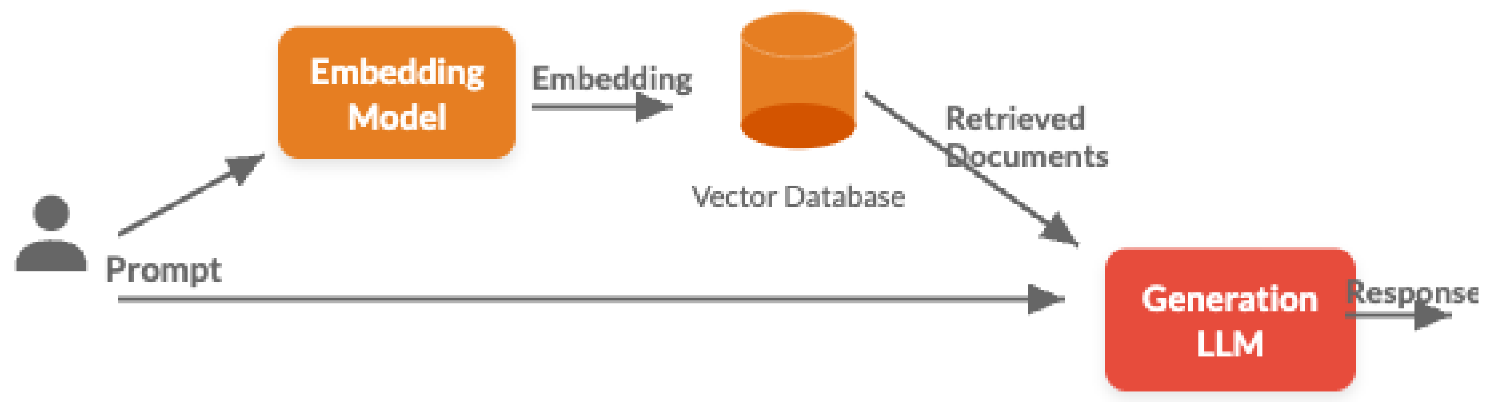
Utilising LLM Applications

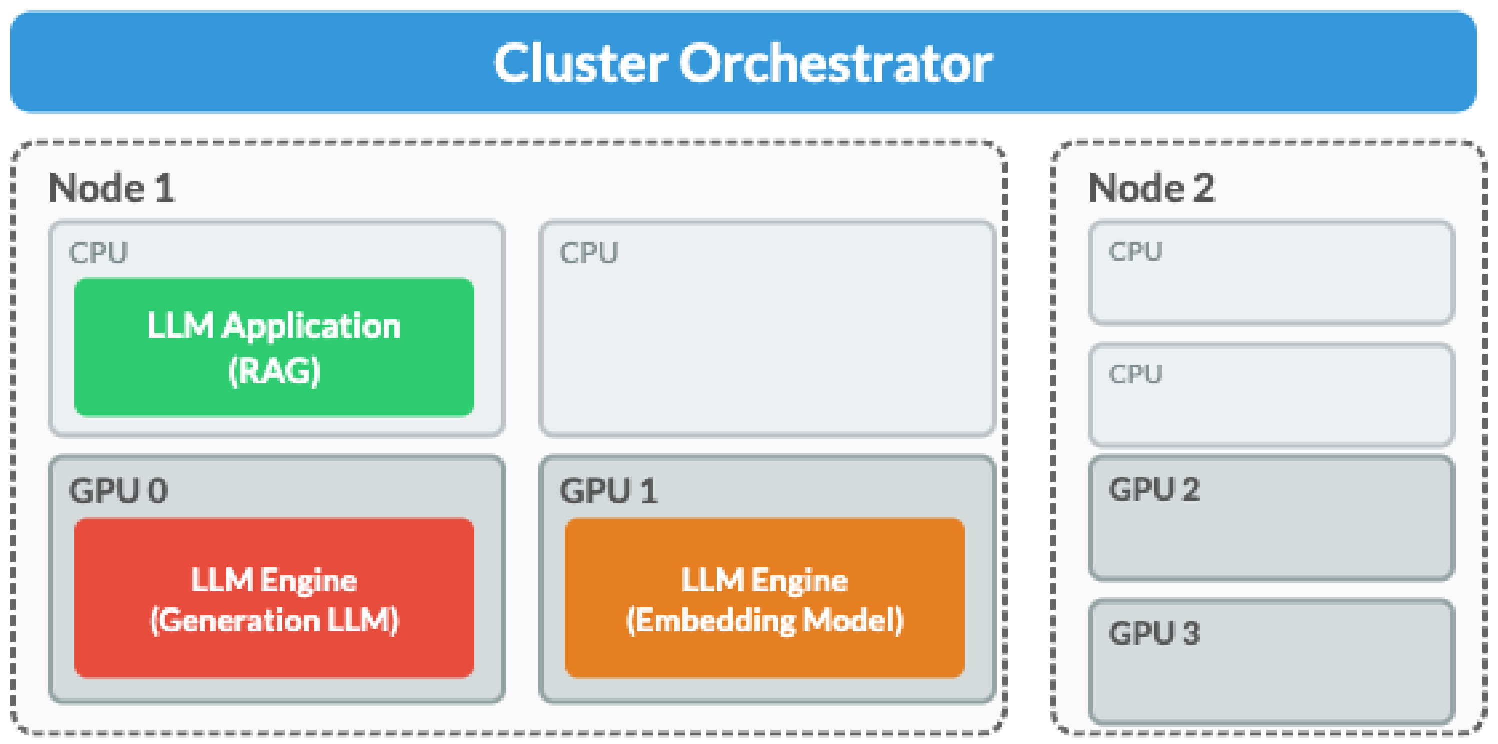
Easier than ever to write

Hard to productionize

LLMs → LLM Applications

- LLM Invocations -> Graphs
- Can't optimise for end-to-end performance
- Lack of Critical Path Awareness
- Unfairness





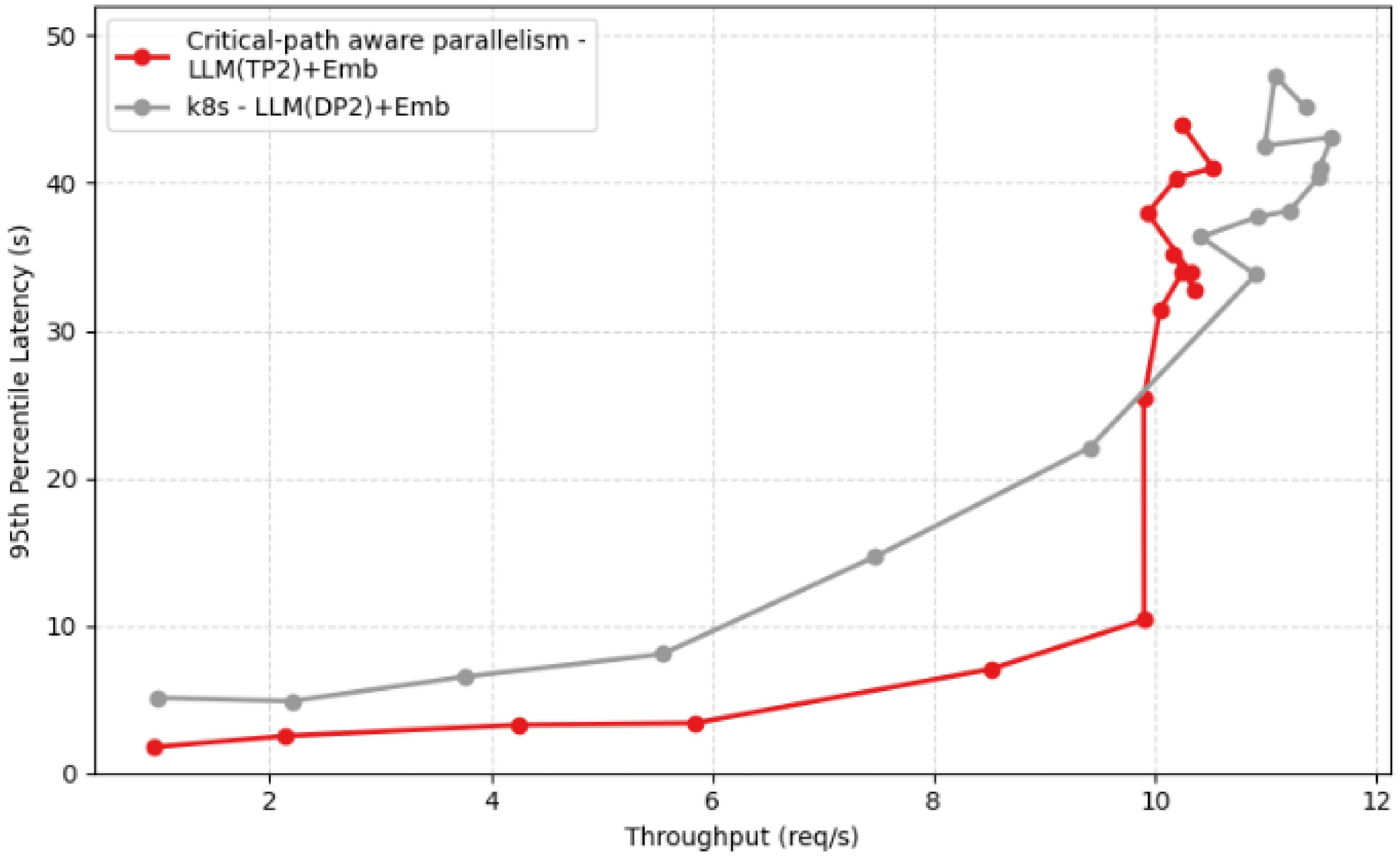
Related Work

System	Level	Multi-Engine	Application-Aware	Scheduling Granularity /Co-location

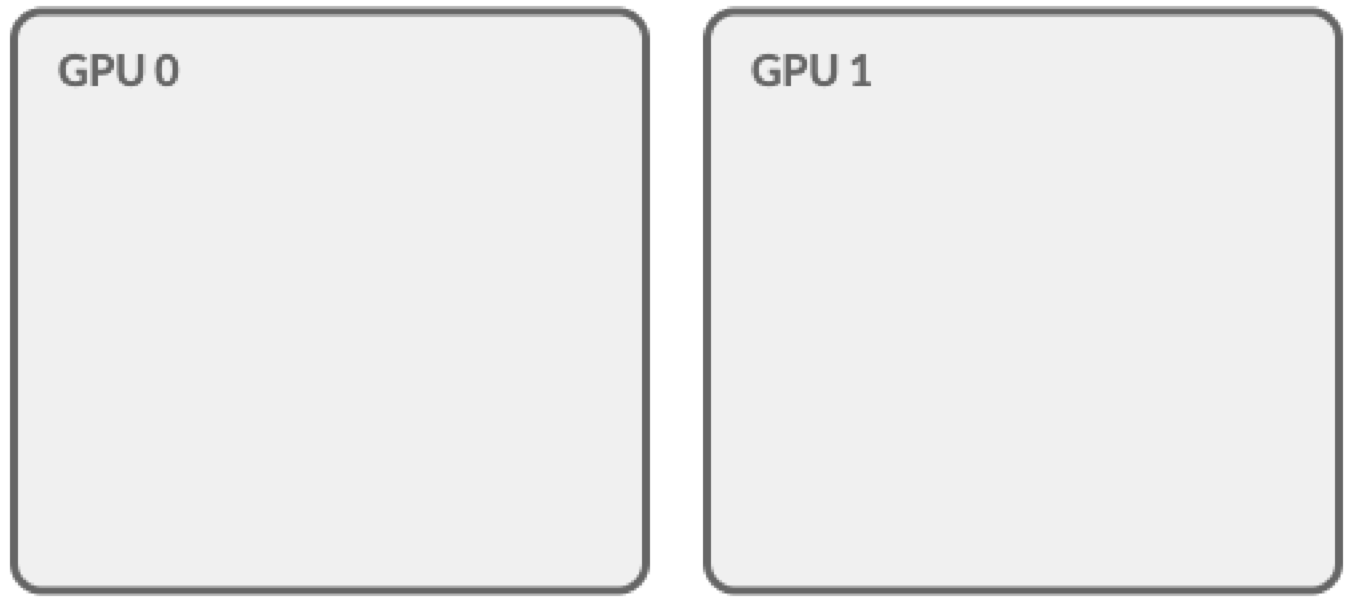
Critical-path Aware Parallelism

Critical-path Aware Parallelism

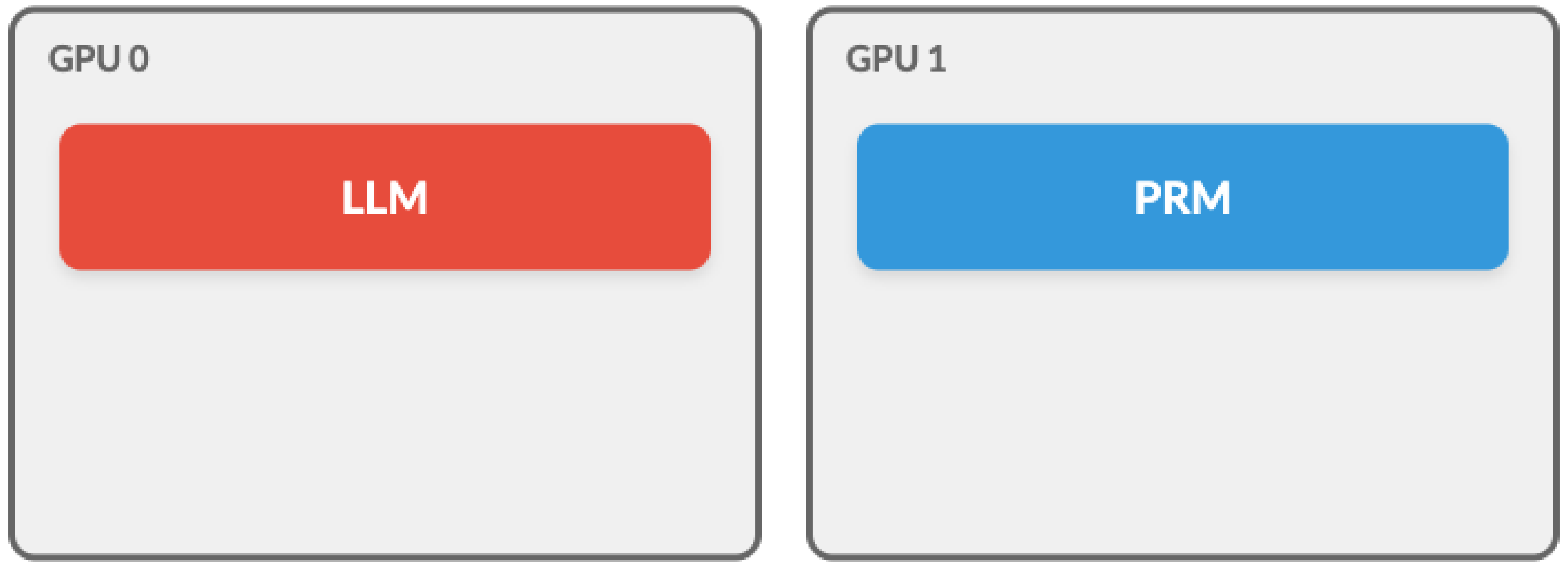
- **2.4x** improvement in latency
- Minor degradation in throughput



Critical-path Aware Co-location

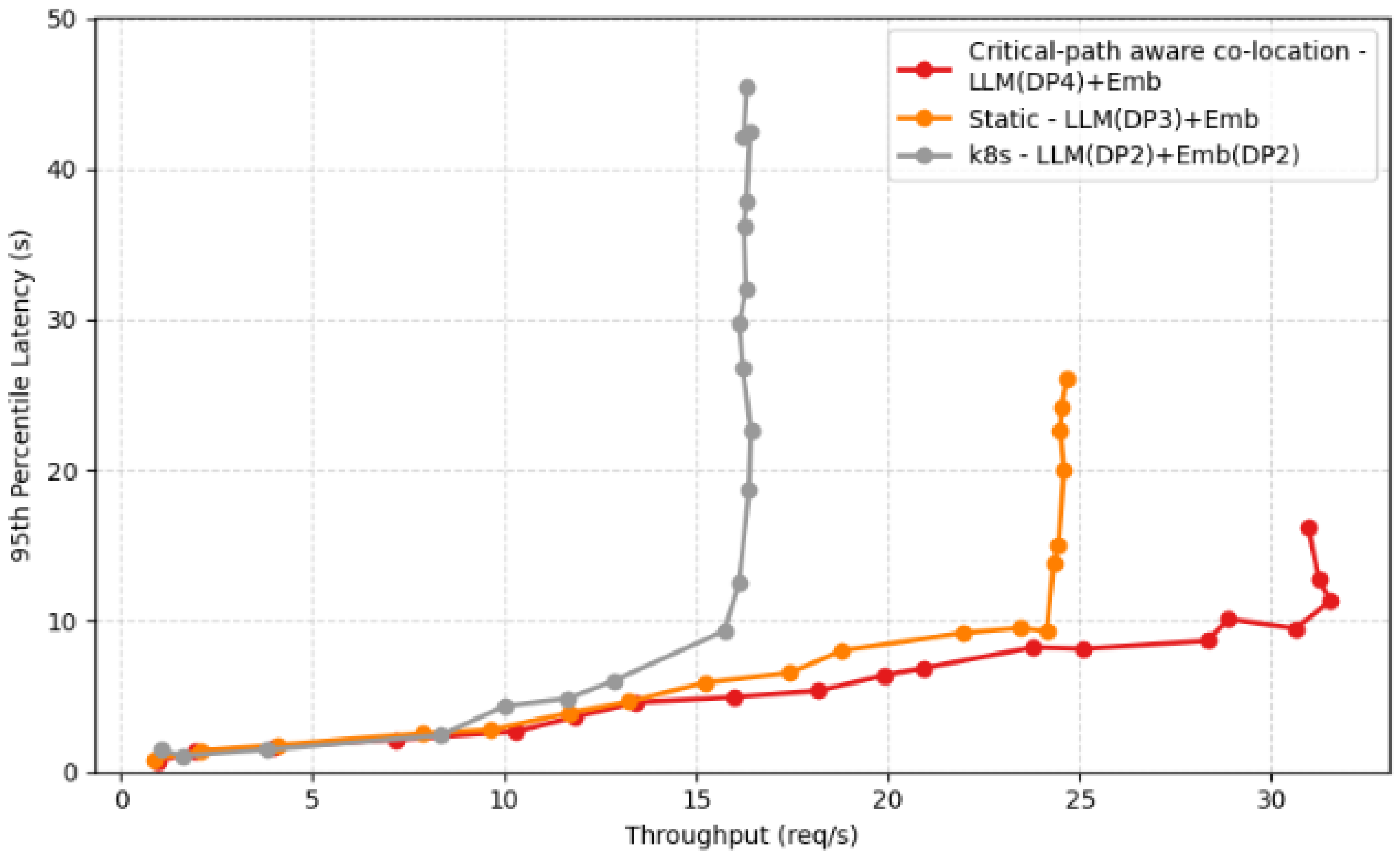


Critical-path Aware Co-location

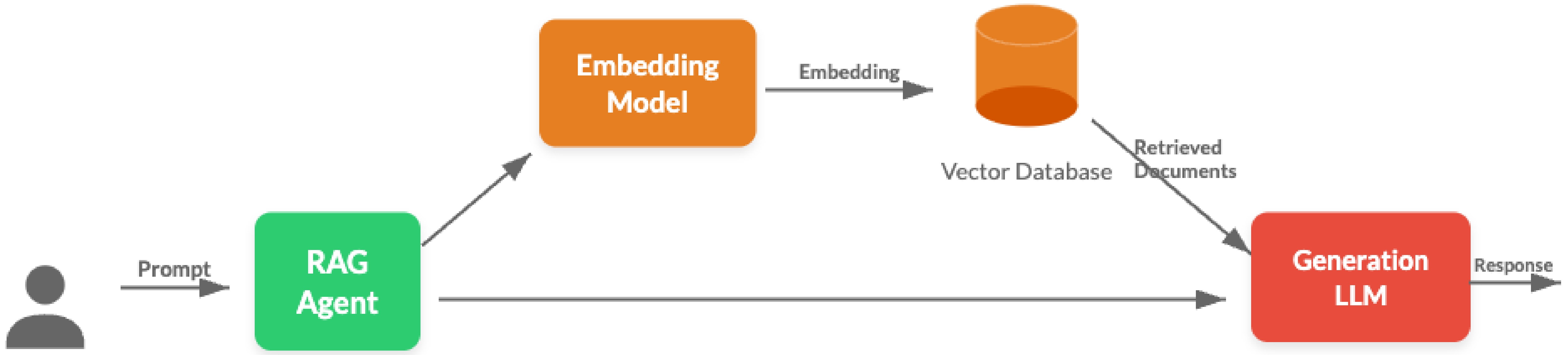


Critical-path Aware Co-location

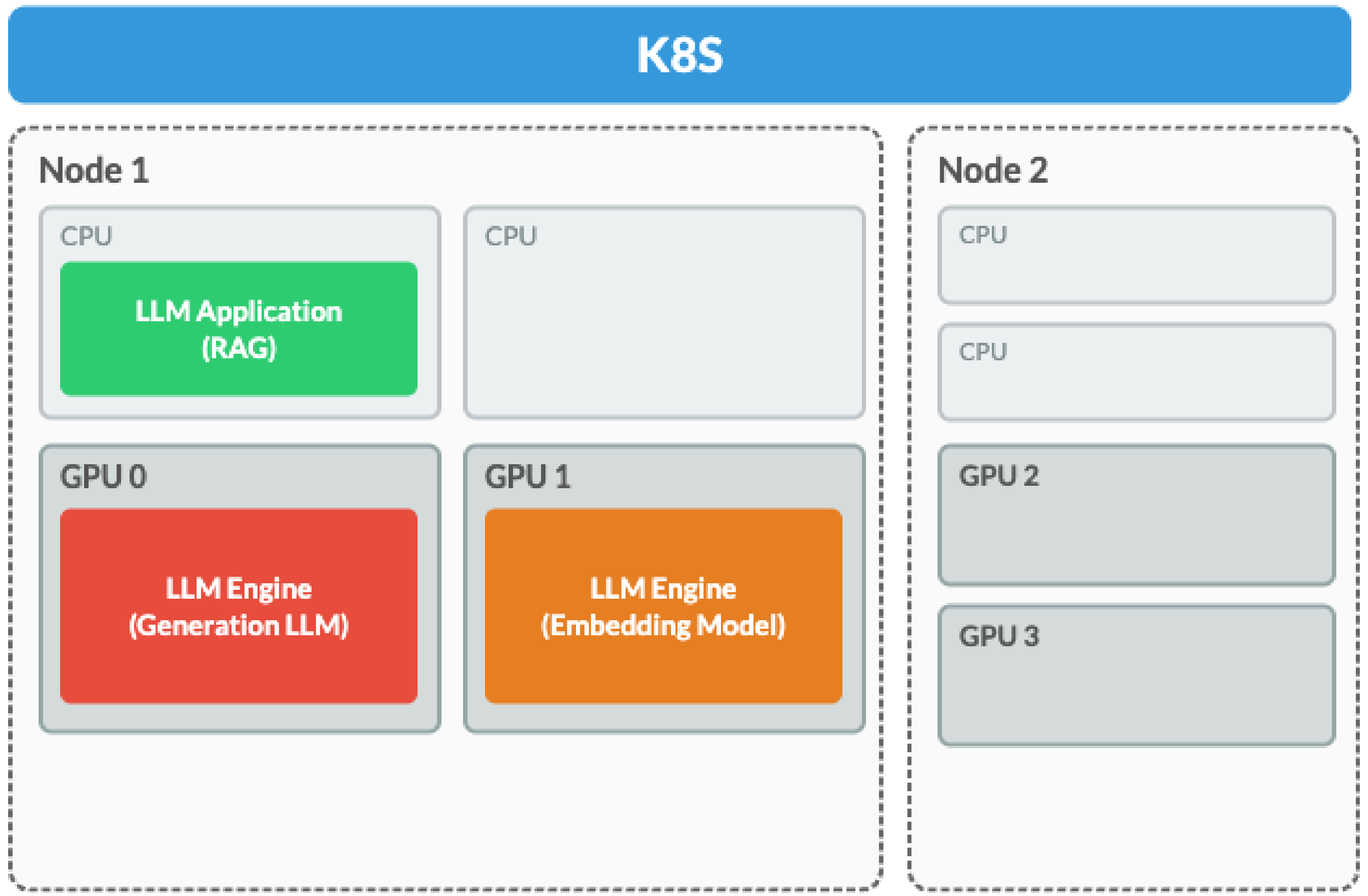
- **2x** throughput over K8S
- **50%** over best manual K8S config
- K8S: data parallelism only



Future Work: Multi-Engine Fairness



Future work: Multi-Engine Fairness


K8S
Node 1
CPU
LLM Application (RAG)
CPU
GPU 0
LLM Engine (Generation LLM)
GPU 1
LLM Engine (Embedding Model)
Node 2
CPU
CPU
GPU 2
GPU 3