Grappa: enabling next-generation analytics tools via latency-tolerant distributed shared memory

Grappa is a modern take on software distributed shared memory, tailored to exploit parallelism inherent in data-intensive applications to overcome their poor locality and input-dependent load distribution.

Grappa differs from traditional DSMs in three ways:

- Instead of minimizing per-operation latency for performance, Grappa tolerates latency with concurrency (latency-sensitive apps need not apply!)
- Grappa moves computation to data instead of caching data at computation
- Grappa operates at byte granularity rather than page granularity

Latency tolerance has been applied successfully in hardware for nanosecond latencies (e.g., superscalars and GPUs). This project explores the application of this idea at distributed system scales with millisecond latencies.

More info, papers, code:
http://grappa.io

Grappa is a modern take on software distributed shared memory, tailored to exploit parallelism inherent in data-intensive applications to overcome their poor locality and input-dependent load distribution.

Grappa differs from traditional DSMs in three ways:

- Instead of minimizing per-operation latency for performance, Grappa tolerates latency with concurrency (latency-sensitive apps need not apply!)
- Grappa moves computation to data instead of caching data at computation
- Grappa operates at byte granularity rather than page granularity

Latency tolerance has been applied successfully in hardware for nanosecond latencies (e.g., superscalars and GPUs). This project explores the application of this idea at distributed system scales with millisecond latencies.

In-memory MapReduce

~150 lines of code, implemented with forall loop over inputs followed by forall over keys

K-Means computation with 64 nodes on SeaFlow flow cytometry dataset with two different k values, compared with Spark using MEMORY_ONLY fault tolerance

GraphLab-like API

~60 lines of code, implementing:
- Synchronous engine with delta caching
- Random graph partition with no replication

Benchmarks run on 31 nodes using 1.8B edge Friendster social network graph and 1.4B edge Twitter follower dataset, compared with GraphLab using two partitioning strategies

Performance breakdown (or “why is Grappa faster?”)

More efficient network layer, lower serialization cost.