A Global In-Memory Cache and Computation Tier for DAOS

J. L. Byrne, C. Crasta, A. Dwaraki, D. Emberson, H. Kuno

S. Lee, S. Singhal, R. A. Rao, S. V. Basri K S, Amitha C

C. Ghosh, R. K. Rajak, S. Ravishankar, P. Shome, L. Evans

S. George, K. Rehm, M.J. Son, T. Kim, S. Hou

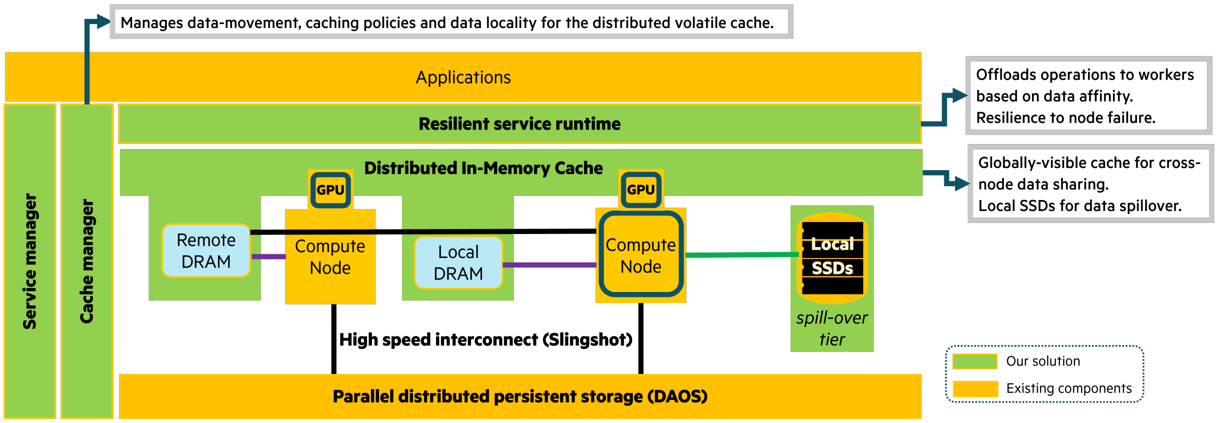
Manages data-movement, caching policies and data locality for the distributed volatile cache.
Applications
Service manager
Cache manager
Resilient service runtime
Offloads operations to workers based on data affinity.
Resilience to node failure.
Distributed In-Memory Cache
Globally-visible cache for cross-node data sharing.
Local SSDs for data spillover.
GPU
GPU
Remote DRAM
Compute Node
Local DRAM
Compute Node
Local SSDs
spill-over tier
High speed interconnect (Slingshot)
Parallel distributed persistent storage (DAOS)
🟩 Our solution
🟨 Existing components

Figure 1: TitaniumRattlesnake extends DAOS with a global client-side cache and resilient runtime.

ACM Reference Format:
J. L. Byrne, C. Crasta, A. Dwaraki, D. Emberson, H. Kuno, S. Lee, S. Singhal, R. A. Rao, S. V. Basri K S, Amitha C, C. Ghosh, R. K. Rajak, S. Ravishankar, P. Shome, L. Evans, S. George, K. Rehm, M.J. Son, T. Kim, and S. Hou. 2024. A Global In-Memory Cache and Computation Tier for DAOS. In *PDSW 2024.* ACM, New York, NY, USA, 1 page.

The AI and data analytics communities rely on programming frameworks that differ from HPC programming frameworks both in usage and capabilities. One key difference is that non-HPC programming frameworks offer more support for an interactive usage style than traditional HPC frameworks (e.g., Python plus Ray vs. slurm plus MPI). This raises three challenges for using non-HPC programming frameworks on HPC systems:

1. Some popular programming frameworks (e.g., pandas, networkX) do not naturally support distributing work across multiple compute nodes.
2. Popular frameworks that do support distributing work across multiple compute nodes may involve fixed resource allocations, which means that long-running interactive workloads with varying resource requirements are vulnerable to stranded resources and out-of-resource errors.
3. Traditional failure recovery mechanisms use checkpoint/restart techniques – however, the time needed to take and re-launch from checkpoints is challenging for large jobs.

We address these challenges by developing TitaniumRattlesnake (TR), a solution that enables ordinary programmers interactively using popular programming frameworks like Python to solve huge problems on HPC systems without stranding resources. Sketched in Figure 1, TR augments Distributed Asynchronous Object Storage (DAOS) with a low-latency/high-bandwidth hierarchical distributed cache and a resilient runtime that intercepts calls to popular frameworks and offloads them to worker processes running on the HPC system. Decoupling worker processes from user applications enables more elastic resource allocation, preventing both stranded resources and out-of-resource errors. Furthermore, because the global cache can be queried, the runtime can schedule offloaded work to exploit locality with regard to the distributed cache. Also, when combined with DAOS, the cache can enable high-speed checkpoints and recovery from failure. Together, the cache and runtime extend DAOS with a performant data and computation tier.

TitaniumRattlesnake is a research project that centers on use cases and focus areas based on customer pain points. We are developing prototype code on a 52 node Slingshot-based HPC cluster with CPUs and plan to extend the work to system with GPUs, including investigating how to extend the Cache Manager and Runtime to achieve efficient GPU sharing. We have implemented an initial proof-of-concept and if accepted we would describe our architecture and present results from some of many experiments that we have been running to inform system design decisions.

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 third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Parallel Data Systems Workshop, November 2024, Atlanta, GA, USA
© 2024 Copyright held by the owner/author(s).