Programmable Storage for Datasets

Team: Aaron Chu, Ivo Jimenez, Jeff LeFevre, Carlos Maltzahn
Institution: UC Santa Cruz

iris hep — Institute for Research & Innovation in Software for High Energy Physics

Baskin Engineering UC Santa Cruz

CROSS — Center for Research in Open Source Software

Data movement costs

CPU: servers copy data to network, clients copy data from network
Memory: clients store received data
Storage: clients store received data that won't fit into memory
Network: bandwidth *provisioning*

→ Move *computational context* to data instead of data to computational context.

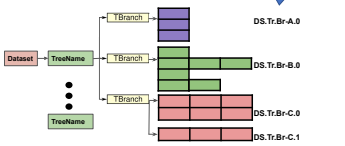
Computational context

Completeness: the actual computation has complete input
Semantics: input is meaningful
Orchestration: data and operations can be mapped to and coordinated within a distributed service, trading off between locality and parallelism.
Robustness: operations can progress in spite of hardware and software failures

→ *Partition* data along semantic boundaries so that operations are mostly local

Dataset semantically mapped to objects

Names are *calculated*, avoiding lookups



Future: Extreme heterogeneity

Disruption: due to end of Dennard Scaling and Moore's Law new devices that undermine traditional software architectures arrive more frequently
Abstraction: software layers isolate applications from disruption but have to minimize assumptions about devices in order to survive
Correctness: correct software takes time to build -- incorrect storage systems lose data.
Performance: software layers have to be able to both hide latency of slow devices and hide bandwidth of fast devices, and transform workloads for diverse device performance profiles for a multi-tiered storage hierarchy spanning multiple orders of magnitude performance differences

→ *Offload* device-sensitive operations to storage systems that manage these devices

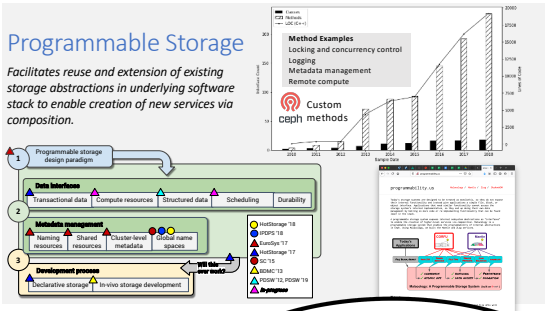
Contact

Jeff LeFevre
Assistant Adjunct Professor
users.soe.ucsc.edu/~jlefevre

Carlos Maltzahn
Adjunct Professor
people.ucsc.edu/carlosm

Programmable Storage

Facilitates reuse and extension of existing storage abstractions in underlying software stack to enable creation of new services via composition.

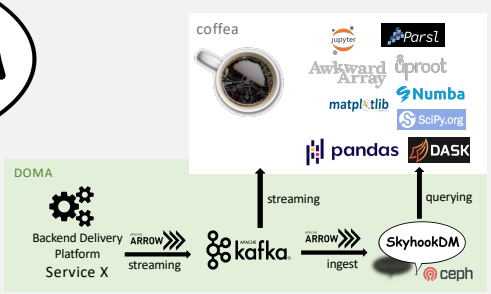


SkyhookDM

Synergies

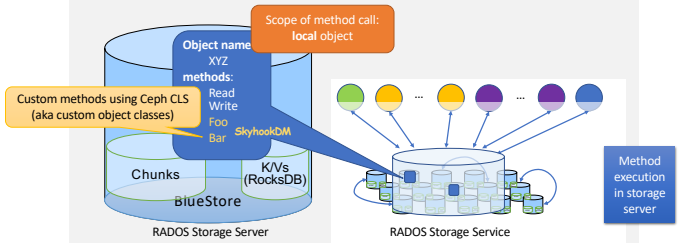
Coffea: Interfacing Coffea processor functions with SkyhookDM
SSL: Using Kubernetes infrastructure for prototyping, testing, and integration
Programmable Storage: SkyhookDM and Ceph are used as research and prototyping platform for software-defined storage services (see programmability.us)
Eusocial Storage: SkyhookDM and Ceph are used for design investigation of collectively acting storage devices (see people.ucsc.edu/carlosm/dev/project/eusocial-storage)
Ceph: widely used, industry-strength storage system, including in the HEP community. Design work uncovers and fixes bugs and other shortcoming
CROSS: The UC Santa Cruz Center for Research in Open Source Software funds part of SkyhookDM development and community work.

Ongoing Integration Effort

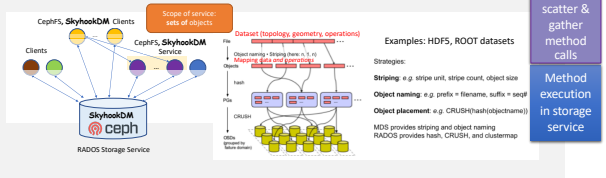


Self-healing storage *service* via P2P architecture

New storage services by individual objects

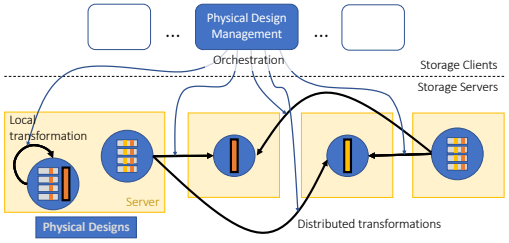


New storage services over sets of objects

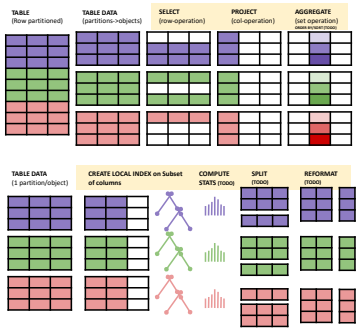


Asynchronous data layout and indexing optimization

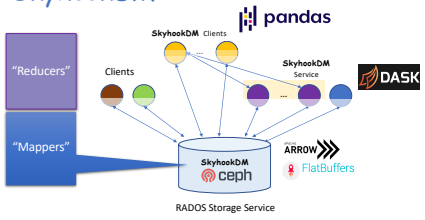
- SkyhookDM is using fast serialization libraries like FlatBuffers and Apache Arrow to store *structured* data.
- Structured data can be stored in different *physical designs* (e.g. with/without indexing, row-oriented vs column-oriented) optimizing for particular access patterns and particular storage devices.
- Physical Design Management is the carefully scheduled orchestration of transformations of physical designs (e.g. indexing, transforming datasets from row to column) balancing beneficial and adverse impacts on production workloads.
- See [DLS+ 19]



SkyhookDM Services



SkyhookDM Architecture



References

[CLM+ 19] Aaron Chu, Jeff LeFevre, Carlos Maltzahn, Aldrin Montana, Peter Alvaro, Dana Robinson, and Quincey Koziol. Skyhookdm: Mapping scientific datasets to programmable storage. In 24th International Conference on Computing in High Energy & Nuclear Physics, Adelaide, Australia, November 4-8 2019.

[DLS+ 19] Kathryn Dahlgren, Jeff LeFevre, Ashay Shirwadkar, Ken Iizawa, Aldrin Montana, Peter Al- varo, and Carlos Maltzahn. Towards physical design management in storage systems. In 4th International Parallel Data Systems Workshop (PDSW 2019, co-located with SC'19), Denver, CO, November 18 2019.

[LM20] Jeff LeFevre and Carlos Maltzahn. Scaling databases and file APIs with programmable Ceph object storage. In 2020 Linux Storage and Filesystems Conference (Vault'20, co-located with FAST'20 and NSDI'20), Santa Clara, CA, February 24-25 2020.

[LWSM19] Jeff LeFevre, Noah Watkins, Michael Sevilla, and Carlos Maltzahn. Skyhook: Programmable storage for databases. In 2019 Linux Storage and Filesystems (Vault'19, co-located with FAST'19), Boston, MA, Februrary 25-26 2019.

[SJW+ 18] Michael A. Sevilla, Ivo Jimenez, Noah Watkins, Jeff LeFevre, Peter Alvaro, Shel Finkelstein, Patrick Donnelly, and Carlos Maltzahn. Cudele: An api and framework for programmable consistency and durability in a global namespace. In IPDPS 2018, Vancouver, BC, Canada, May 21-25 2018.

[SWJ+ 17] Michael A. Sevilla, Noah Watkins, Ivo Jimenez, Peter Alvaro, Shel Finkelstein, Jeff LeFevre, and Carlos Maltzahn. Malacology: A programmable storage system. In EuroSys '17, Belgrade, Serbia, April 23-26 2017

[SJW+ 18] Michael A. Sevilla, Ivo Jimenez, Noah Watkins, Jeff LeFevre, Peter Alvaro, Shel Michael Sevilla, Noah Watkins, Carlos Maltzahn, Ike Nassi, Scott Brandt, Sage Weil, Greg Farnum, and Sam Fineberg. Mantle: A programmable metadata load balancer for the ceph file system. In SC '15, Austin, TX, November 2015.

This project is supported by National Science Foundation under Cooperative Agreement OAC-1836650. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.