University of Tsukuba | Center for Computational Sciences

This research is supported by JST/CREST http://postpeta.jst.go.jp/en/
Japan Science and Technology Agency

[JST CREST]
System Software for Post Petascale Data Intensive Science

http://www.hpcs.cs.tsukuba.ac.jp/project/crest-ppfs/en/

Objective Development of **System Software** for Data-intensive Computing to promote Data-intensive Science

Runtime System | **File System Kernel Driver** | **Distributed File System**

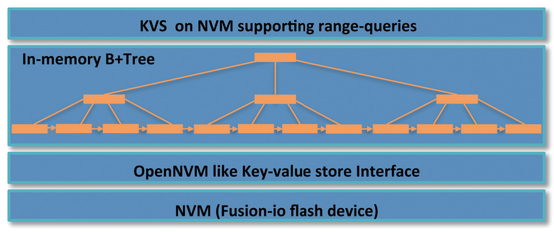
File System Kernel Driver is developed at The University of Electro-Communications

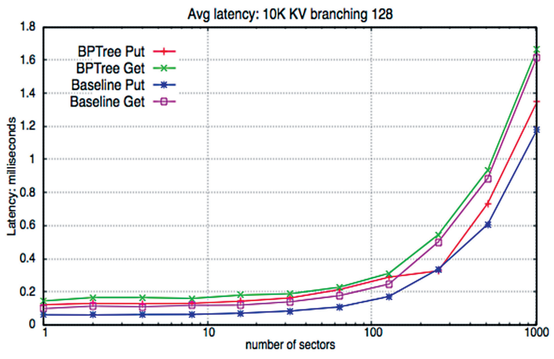
Distributed File System

NVM-BPTree

NVM-BPTree is a Key-Value Stores (KVS) running natively over Non-Volatile-Memory (NVM) like flash supporting range-queries.

- Take advantage of enterprise class NVM new capabilities: atomic writes, direct access to NVM device natively as a KVS,...
  - » *Leverage NVMKV an Open source KVS interface for NVM like flash.*
- Enable range-queries support for KVS running natively on NVM
  - » *Keys stored in a in-memory B+Tree with negligible overhead for key-value pair insertion and retrieval.*
- Provide optional persistence to the BPTree structure and also snapshots



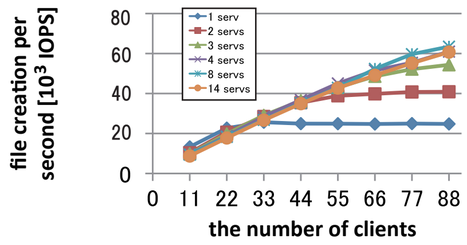


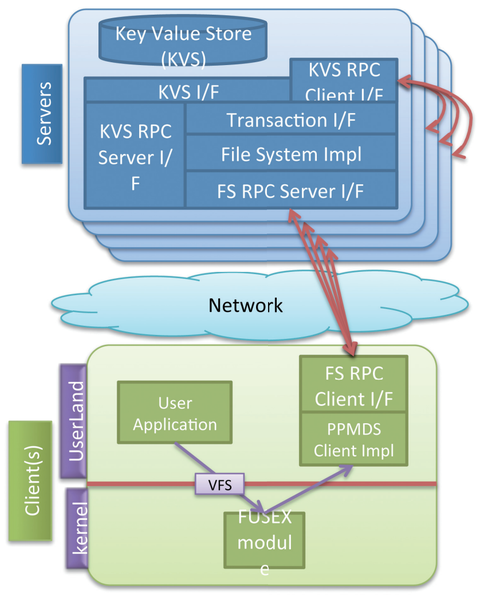
PPMDS: A Distributed Metadata Server

PPMDS is a distributed metadata management system for a distributed file system which targets post-petascale super computers.

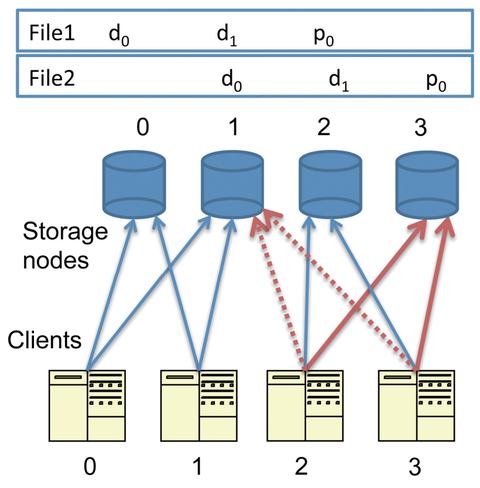
Fine-grained parallelism. The system manages directory namespace efficiently by ordered key-value store. The keys consist of **a pair of a parent inode number and a file name.** The values store metadata. The largest granularity of locking is the key-value pair.

Nonblocking transactions across multiple key-value servers. The system supports it by Dynamic STM to update multi key-value pairs transactionally.

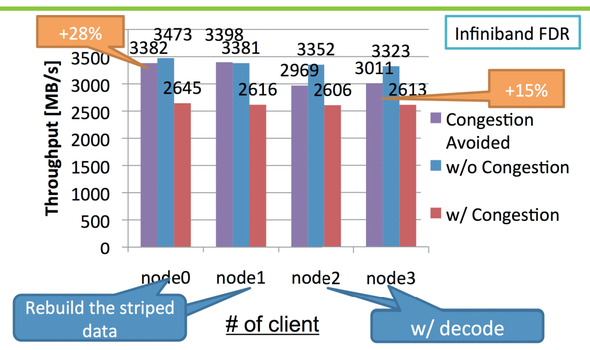




Congestion Avoidance w/ Redundant Data



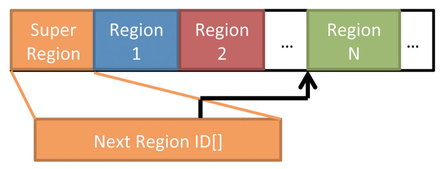
Network of storage node #1 is congested when all clients access the files concurrently. This congestion is avoided by the proposed method, which uses the redundant data to disperse the traffic.

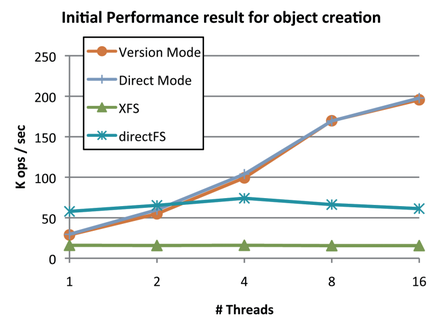


Object Storage for High Speed Storage Device

Design of object storage for Fusion IO ioDrive to achieve maximum IOPS/bandwidth performance.

- ioDrive supports **144PB virtual address space** and **atomic-write.**
- Contiguous 2TB fixed-size **regions** for each object
  - » Region can be specified by the Object ID
  - » One region is for one object.
  - » All meta-data about the object is stored in the region.
  - » Block locks are reduced to 1/1000.
- The object storage supports direct mode and version mode
  - » Direct mode is for fast read/write
  - » Version mode is for fast write and maintains all versions by log structured format



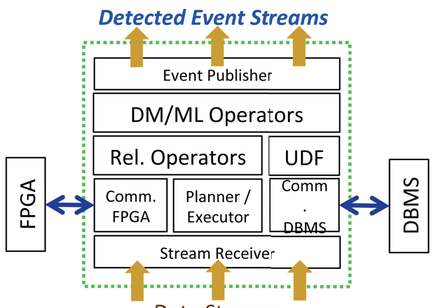


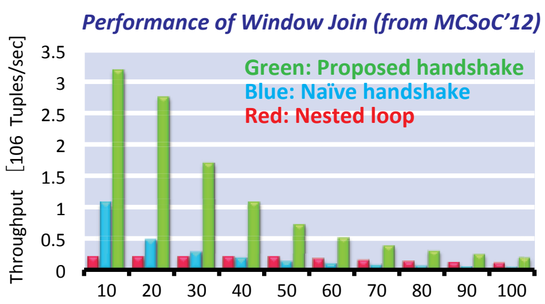
Runtime System

FALCON: Yet Another Data Stream Management System

FALCON is a DSMS that provides both relational and data mining operators.

- Currently supported mining operators: change point detection, distance based outlier, local outlier factor, frequent itemset mining, and Bayesian networks.
- Multiple execution of CPD is accelerated by micro operator sharing (BIRTE' 13).
- Window join is accelerated by adaptive merging network on FPGA (MCSoC' 12, SSDBM' 13).

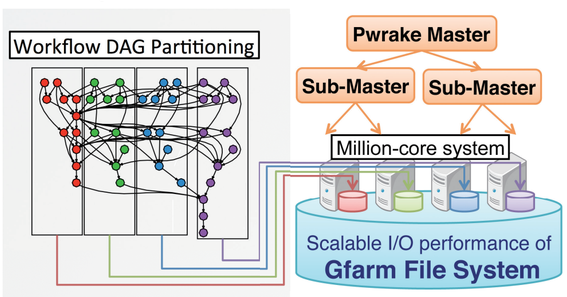




Pwrake: Scalable Workflow System

Pwrake is a workflow engine for data-intensive sciences with the following features.

- Based on ***Rake***, a powerful **Workflow Language**.
  - » *Rake* is a widely-used build tool similar to *Make*. Any complex workflow can be defined in *Rakefile,* powered by Rake's *rule* definition, and *Ruby* language features.
- **Scalable I/O Performance** by utilizing **File Locality**.
  - » Workflow scheduling to minimize data transfer using **Multi-Constraint Graph Partitioning** algorithm.
- **Post Petascale system** is the next target of Pwrake.
  - » **Hierarchical structure** of the next Pwrake manages **100M tasks** executed on **one million cores**.

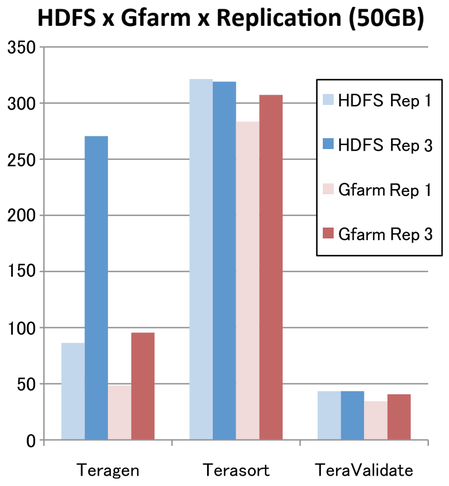


MapReduce and Gfarm File System

Gfarm file system is a network shared file system that supports scalable I/O performance in distributed environment. Executing **MapReduce** applications on top of **Gfarm** provides the following features:

- Data locality
- Fault tolerance by transparent replica access
- Fully *POSIX* compliant file system.
- No need for data import/export to execute MapReduce
- Access from local software by *FUSE API*

Our recent studies show **MapReduce on top of Gfarm** file system can provide up to 50% **higher data throughput** when compared with traditional HDFS.



Data-Aware Task Scheduling

- **Data-aware Task Scheduling** is the scheduling method for the file systems where the file locality affects the I/O efficiency significantly.
- Two Components
  - » **Data-aware Task Dispatch**(DAD) : Dispatch the task to the node with the lowest ***Score***, where ***Score*** is calculated based on the CPU average and the locality of the file accessed by task.
  - » **Replica Generation**(RG):Replicate the frequently accessed file to idle node to increase the system utilization.