A Persistent Storage Model for Extreme Scale

Shuangyang Yang (LSU), Walter B. Ligon III (Clemson),
Maciej Brodowicz (CREST@Indiana), Hartmut Kaiser (LSU)

STE||AR
stellar.cct.lsu.edu

LSU | Center for Computation & Technology

Layout

- Exascale Challenges
- What We Have: HPX and OrangeFS
- PXFS: A Persistent Storage Model
- First Experiment
- Conclusion and Future Work

Exascale Challenges

- Exascale machines: 100s of thousands or millions of nodes each with 50 to 100 cores, including GPUs and FPGAs.
- Challenges in runtime system:
  - Parallelism management, synchronization, scheduling and etc.
  - Latency hiding, overhead reduction and load balancing.
  - Ease to use for programmers and users.
- Challenges in storage system:
  - SLOW secondary storage devices and even worse in next 10 years.
  - Data size and complexity are growing fast.
  - Global access to permanent storage is desired.

What We Have: HPX

- C++ runtime system for parallel and distributed applications.
- Designed for systems of any scale, from hand-held devices to very large scale systems.
- Active global address space (AGAS) .
- Message driven instead of message passing.
- Fine grained parallelism instead of global barriers.
- Automatic load balancing instead of static work distribution.
- Exposes an uniform, standards-oriented API for ease of programming.
- Open-source, available at http://stellar.cct.lsu.edu/downloads/

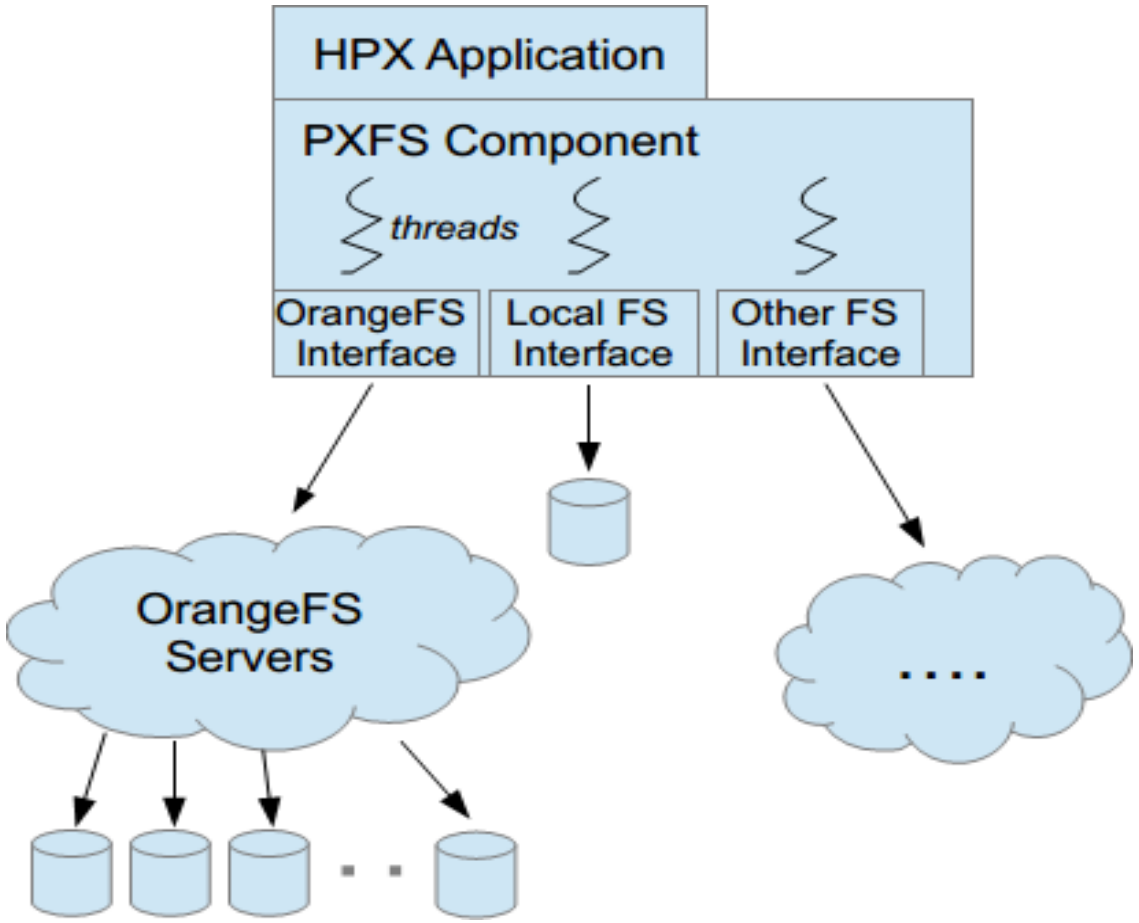
What We Have: Orange File System

- Parallel file system on distributed systems.
- High performance access to disk storage for parallel applications.
- Sophisticated metadata management.
- Hardware independent and multiple client platform support.
- Painless deployment on high end computing systems.
- Commercial support from *Omnibond Systems LLC*.
- Open-source, available at http://www.orangefs.org/

PXFS: A Persistent Storage Model

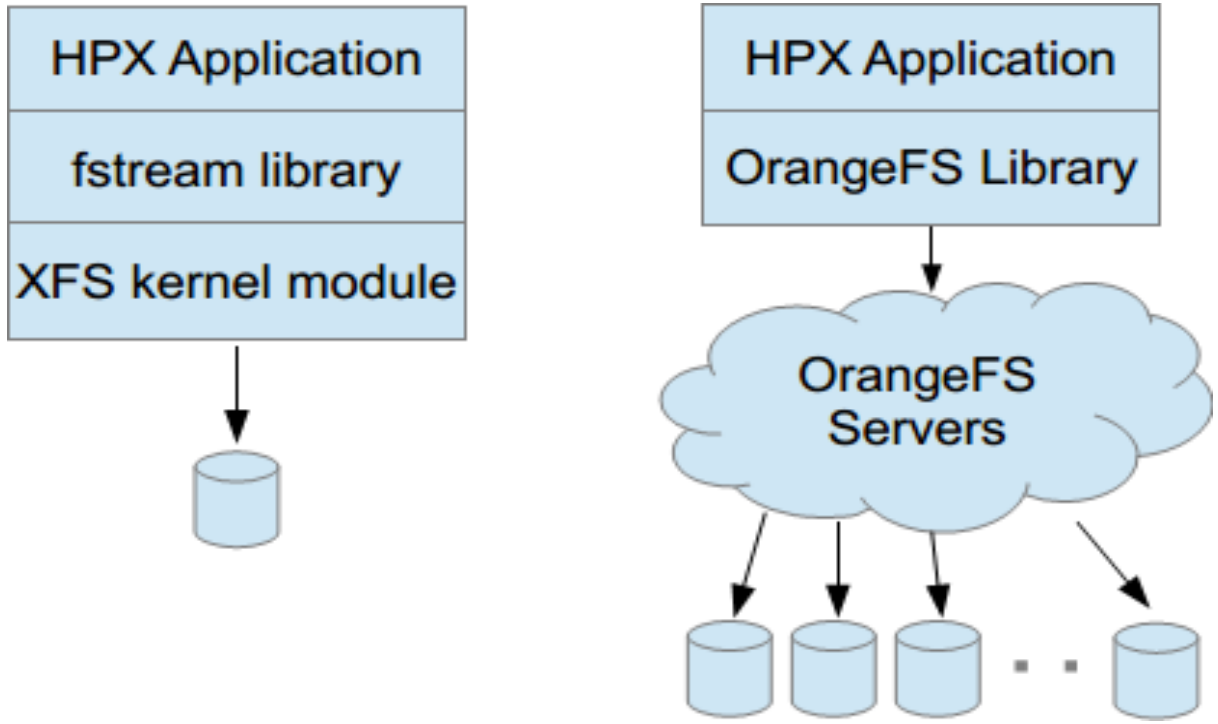
- A proposed I/O model designed to address the challenges of persistent storage in the Exascale era.
- Unify the secondary storage space with the HPX AGAS namespace.
- Bridge the gap between runtime objects and storage object on permanent storage media.
- Define semantics for storage operations in HPX runtime system.
- Manage synchronization and consistency of storage objects.

PXFS: Design Diagram

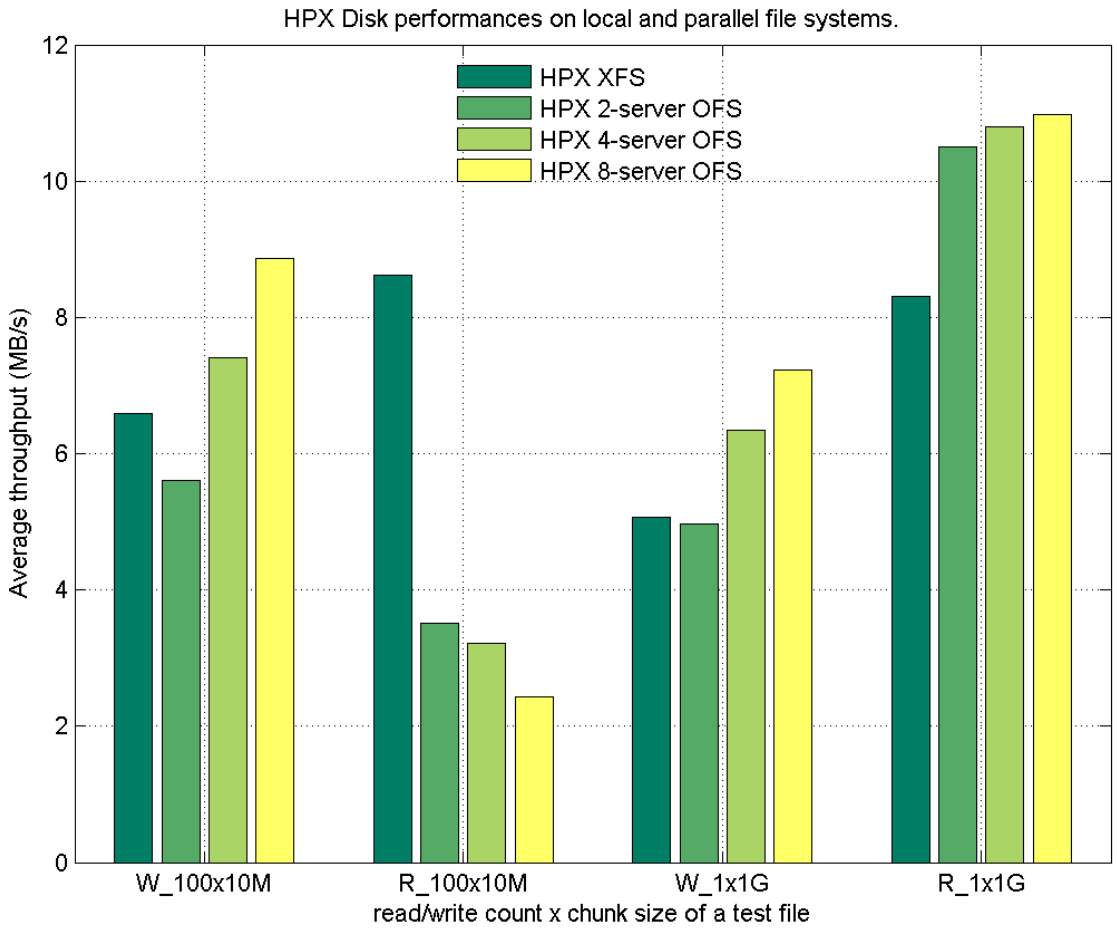


First Attempt: HPX disk performance benchmark

- Directly call OrangeFS APIs from HPX applications to test disk throughput.



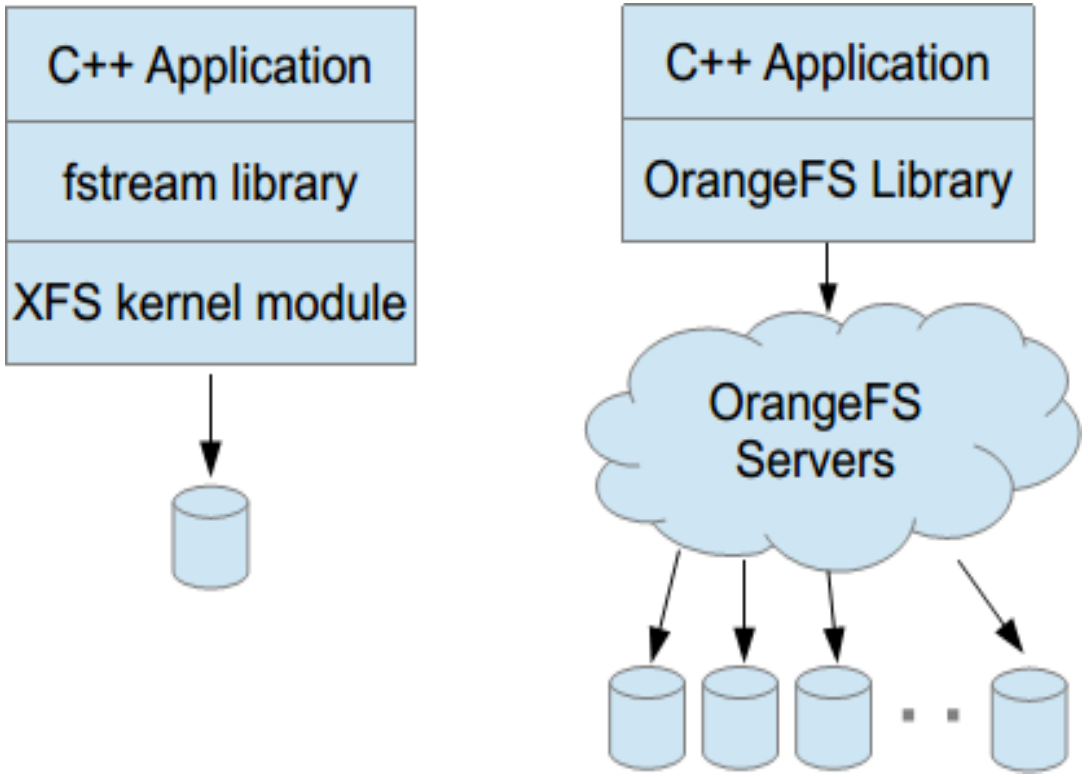
Benchmark Results 1



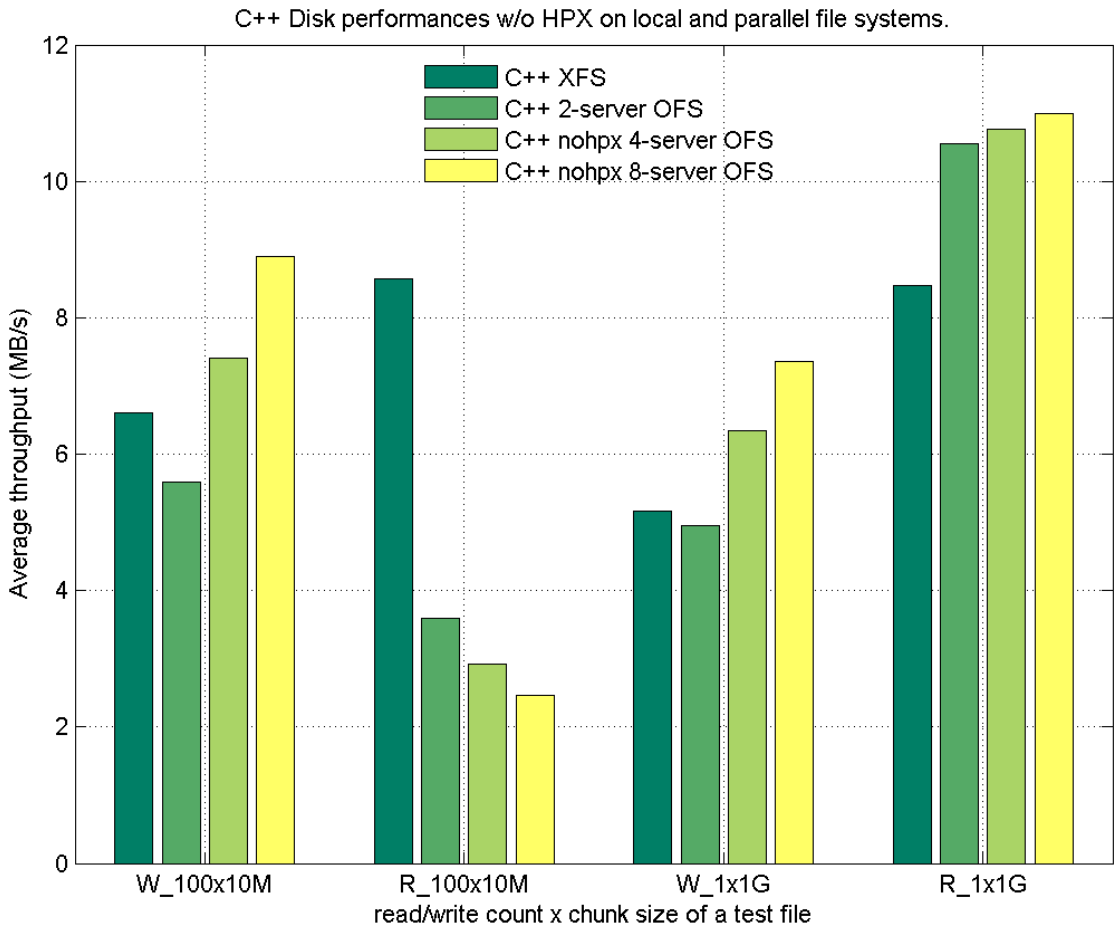
- HPX and OrangeFS glued well.
- OrangeFS has advantage on reading and writing large files w.r.t. local file systems.
- Better throughput with more OrangeFS servers.

Comparison: C++ disk performance benchmark

- Directly call OrangeFS APIs from C++ applications to test disk throughput.



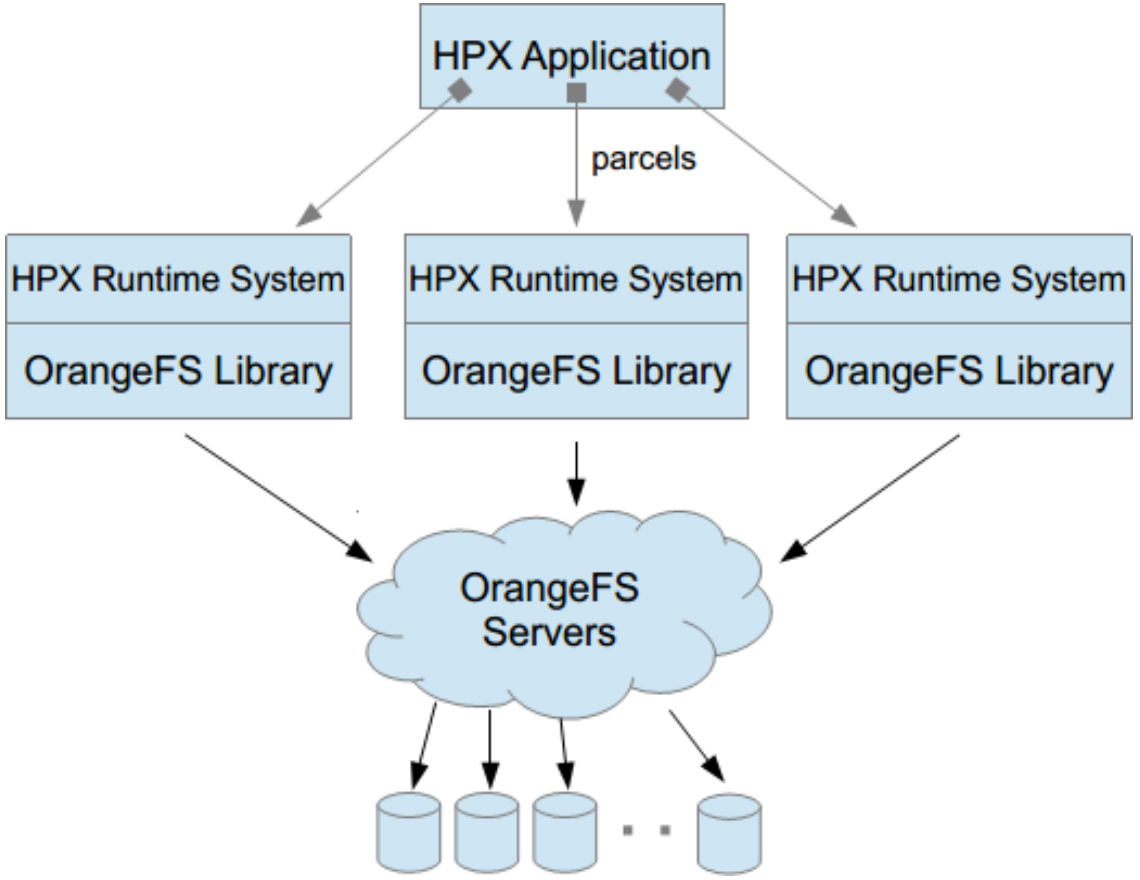
Benchmark Results 2



- Comparing to C++ benchmark, HPX does not add noticeable overhead.

HPX and OrangeFS on distributed machines

- HPX can run on distributed machines.
- OrangeFS distribute files on multiple servers and provide a uniform view.



Future Work

- Design complete PXFS component into HPX runtime system.
- Develop benchmark applications to evaluate the consistency and efficiency of PXFS component.
- Explore metadata management in runtime objects and storage objects.

Acknowledgement