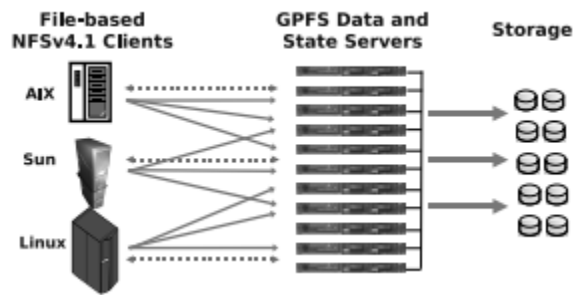


Panache: A Parallel File System Cache for Global File Access

Panache is a read-write, multi-node, highly scalable, high-performance, clustered, file system cache - for parallel data-intensive applications that require wide area file access with additional features like disconnected operations, persistence across failure and consistency management. It simultaneously involves the update of cache from multiple nodes & the data and metadata updates from cache to remote file system using pNFS. Panache also takes care of fluctuations in WAN latencies, is easy to deploy, does not require any specific resources at remote clusters & eliminates single server bottleneck of file access protocol. It also manages conflict handling and resolution in disconnected mode operations managing them in cluster settings.

The fully parallelizable design is due to parallel ingest, parallel access, parallel update, parallel delayed data write-back & parallel delayed metadata write-back. Also, all data and metadata updates are asynchronous which supports WAN latencies and outages. The two basic components of Panache are: GPFS and pNFS. GPFS being the high performance shared disk cluster file system and pNFS protocol enabling the clients for direct and parallel access to storage. The pNFS-GPFS architecture is a three layer structure having file based Clients, GPFS Data and state servers and storage.

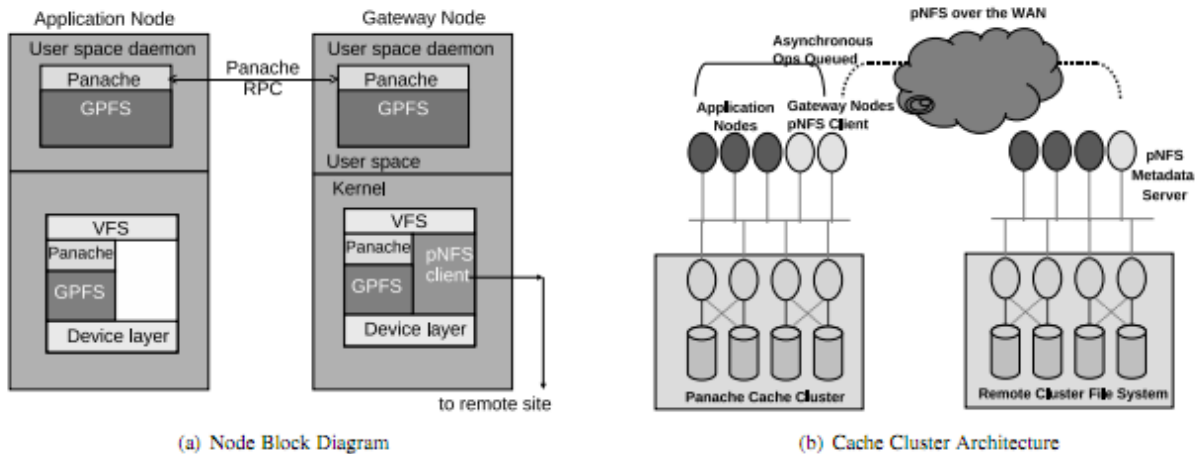


Panache is implemented as a multi-node caching layer, integrated within the GPFS that can persistently and consistently cache data and metadata from a remote cluster. Every node in the Panache cache cluster has direct access to cached data and metadata. Thus, once data is cached, applications running on the Panache cluster achieve the same performance as if they were running directly on the remote cluster. If the data is not in the cache, Panache acts as a caching proxy to fetch the data in parallel both by using a parallel read across multiple cache cluster nodes to drive the ingest, and from multiple remote cluster nodes using pNFS. Panache allows updates to be made to the cache cluster at local cluster performance by asynchronously pushing all updates of data and metadata to the remote cluster.

The cache cluster are of two categories : Gateway nodes and Application nodes. [Local is the client side with local cache. Remote is the clustered environment.] The paper discusses Local consistency, Validity Lag, Synchronization lag and Eventual Consistency as features of Panache. The following observations are discussed wrt to Penanche. **Observation 1** If commands C_i, C_j, C_k are pair-wise time ordered, i.e., $C_i \rightarrow C_j$ and $C_j \rightarrow C_k$ then the three commands form a time ordered sequence $C_i \rightarrow C_j \rightarrow C_k$. **Observation 2** If objects O_x, O_y , and O_y, O_z are pairwise dependent, then O_x, O_z are also dependent objects.

It relies on distributed locking mechanism of the underlying clustered file system and is always locally consistent for updates made at cache cluster. Asynchronous read allows read from cache, but waits for command to check remote site to make sure it is up-to-date. The paper discusses Synchronous and

Asynchronous Operations and dependent metadata operations. Recovery from failure is done through Inode scans. The performance is measured based on metadata performance, WAN performance and I/O performance.



Panache achieves the performance of a clustered file system on a cache hit, large scale applications can leverage a clustered caching solution without paying the performance penalty of accessing remote data using out-of-band techniques. But, the biggest disadvantage of Panache is the way it handles conflicts. In case of asynchronous file updates, the final resolution is a manual procedure which is definitely not a good solution. Also, paper discusses nothing about the cache update policy. Panache faces issues of availability and various performance issues.

The paper acted as a perfect transition paper between the DFS and the regular cluster file system to the wide area performance issues that would be discussed in the forthcoming classes. The discussion in class reiterated the un-clarity in the discussion of the performance issues. The Asynchronous operations forced the Synchronous operations to take a back-seat. It was revealed in the Question Answer session that the paper does not talk about the pre-fetch techniques used in Panache or the size of the local and remote cache.

Reviewer:
Verma, Saakshi