

C-PFS\*, is the high performance parallel file system component of the HPCC software suite providing enhanced disk throughput required by the IO intensive parallel applications. The aggregate bandwidths of the cluster interconnect PARAMNet, and the disk IO channels are effectively utilized to provide the IO throughput. It specifically targets the parallel applications conforming to MPI 2 specifications although it supports UNIX IO function calls as well.

The architecture of C-PFS is based on user level file system concepts. By making use of KSHIPRA for distributed communication and co-ordination, it achieves an end-to-end user level implementation. The file system adaptively operates by learning about the application and system behavior to optimize the IO throughput.

The existing UNIX utilities for managing the file system are supported, as well as supplemented by a set of management tools required due to distributed nature of the file system.

### DESCRIPTION

Parallel File Systems has specifically evolved in response to the IO needs of parallel scientific and engineering applications (see box).

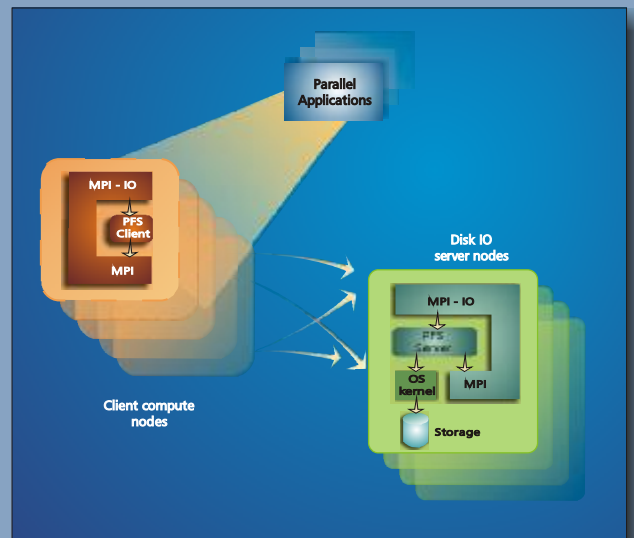
MPI-IO is the standard interface used by such applications for performing IO operations. Hence, an optimised implementation of MPI-IO interface is provided with the C-PFS.

### APPLICATION PROGRAMMING INTERFACES

C-PFS supports the following interfaces:

- ▶ IO extensions to the MPI 2 specification, MPI-IO
- ▶ Most of the POSIX IO related calls

IO extensions to the MPI 2 specification from the MPI Forum, called MPI-IO, allow the programmers to carry out complex file sharing operations and enable the system implementation to map the concurrency in the IO operations on the system architecture. An efficient implementation of MPI-IO library on C-PFS is supported. Binary compatibility with POSIX IO calls, enables the usage of UNIX files systems management utilities and traditional client-server applications.



Software Architecture of C-PFS

### KEY FEATURES

- ▶ Intelligent prefetching policies to avoid latencies
- ▶ Framework to plugin application specific prefetching policies
- ▶ Adaptive techniques to automatically tune with various network and disk speeds
- ▶ Full user level implementation of file system
- ▶ Supports MPI 2 and UNIX interfaces
- ▶ File system Management through standard UNIX tools and utilities
- ▶ Scalable on large clusters

\*Portions of C-PFS code have been derived from the Sun Cluster tools source code made available through the SCSL program that provides the basic structure for the parallel file system.

## PRODUCT OVERVIEW

### FEATURES

C-PFS has performance features designed to support most of the typical workloads of the scientific and engineering applications. By having multiple servers and server-directed-IO mechanism in the clients, it avoids centralized control over the data movement and maximizes the usage of the switched data paths on the cluster interconnect. Fine grain locking make complex partitioning and concurrent IO possible.

Apart from the efficient implementation techniques, the file system has several features that exploit the IO behavior of the applications. It learns about the access patterns of the applications on the fly and utilizes this information to optimize the performance. The framework also allows users to exploit the knowledge of application access pattern by integrating the information with C-PFS. Based on the application IO message size and the performance of the network and the disk, system parameters are adaptively changed to maintain a pipeline for the efficient usage of resources.

#### Prefetching

Owing to large datasets that are common in this class of applications, intelligent prefetching has been chosen as the primary technique to mask the disk

latency. A minimal amount of caching has been provided to improve the performance of finegrained applications. The default predictors are based on text compression algorithms such as LZ and PPM. Predictors based on the text-based compression are very effective in IO access patterns having simple arithmetic formulations. APIs have also been provided for users to specify better algorithms based on application behavior.

#### Adaptive pipelining

Based on the performance of the network and the disk, the packet sizes for communication between the IO servers and the clients are dynamically resized, such that disk bandwidth is effectively utilized. This is particularly very useful in collective IO operations.

#### UNIX management utilities

To simplify the management of parallel file system, a UNIX interface is provided to facilitate the usage of UNIX file system management tools. Tools such as mount, fsck etc are supported so that these traditional tools can be used to repair and maintain the file system. Binary compatibility for the applications compiled with UNIX IO calls is provided.

### Why Parallel File Systems?

A Parallel file system has primarily evolved to cater to the requirements of scientific and engineering application community where large IO throughput is required for a single application. In a simpler model, this requires every single IO operation to be handled by large number of disk spindles to provide high throughput. Also, the threads in the application job typically share a set of few files in a complex, yet well-defined manner. Issues are further complicated for a distributed memory parallel machine where the data sets are scattered across the different application threads on various nodes require sophisticated scatter-gather communication operations. Design issues of the parallel file systems address these requirements by providing a parallel IO programming interface and also a system architecture that can support high throughput. With clusters emerging as an affordable parallel computing platform, commodity based PFS implementation such as C-PFS provide high performance IO on the cluster.

### AVAILABILITY

Supported Hardware	:	Workstation Clusters
Supported Operating System	:	AIX, Solaris
User Interfaces	:	Command Line (for management functions)
Supported Languages	:	C, Fortran
Prerequisite Hardware	:	Any network with TCP-IP support, PARAMNet
Supported APIs	:	MPI 2, POSIX IO calls



## Centre for Development of Advanced Computing

C-DAC Knowledge Park, No. 1, Old Madras Road, Byappanahalli, Bangalore - 560 038, India  
Tel: +91-80-534 1874, 534 1909 Fax: +91-80-524 7724  
e-mail: [bdm@cdacindia.com](mailto:bdm@cdacindia.com) website: <http://www.cdacindia.com>

#### Head Office

Pune University Campus, Ganeshkhind,  
Pune - 411 007, India  
Tel: +91-20-569 4000/01/02/03  
Fax: +91-20-569 4059

#### New Delhi

A 335, Shivalk Enclave,  
Near Malviya Nagar,  
New Delhi - 110 017  
Tel/Fax: +91-11-667 4689/91/97  
e-mail: [bd@cdacindia.com](mailto:bd@cdacindia.com)

#### Hyderabad

2nd Floor, Delta  
Chambers,  
Ameerpet,  
Hyderabad - 500 016  
Tel: +91-40-340 1331/32  
Fax: +91-40-340 1531

• Chennai: +91-44-371 9226/27

• Kolkata: +91-33-321 2357

• Thiruvananthapuram: +91-471-554086