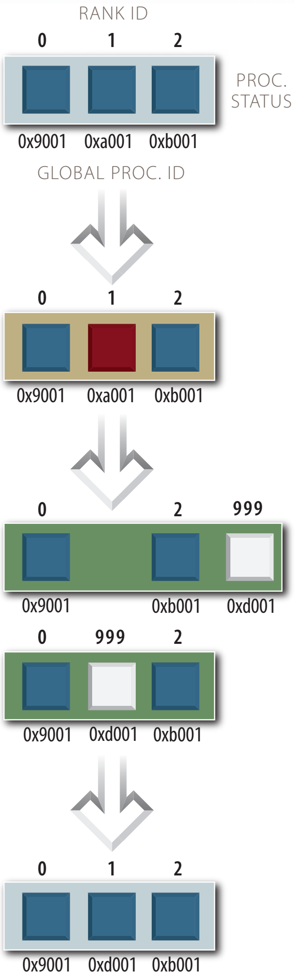
FT-MPI

http://icl.cs.utk.edu/ftmpi/

FAULT-TOLERANT MESSAGE PASSING INTERFACE

FT-MPI is an independent implementation of the MPI 1.2 message passing standard that has been built from the ground up with both user and system level fault tolerance. FT-MPI allows developers to build fault tolerant or survivable applications that do not immediately exit due to the failure of a processor, node, or MPI task. A number of failure modes are offered that allow a range of recovery schemes to be used that closely match different classes of parallel applications. FT-MPI is unique because it avoids restarting surviving nodes, which can be a considerable advantage on very large scale systems where rescheduling and restarting of the entire application is the only current option.

Although FT-MPI has additional features compared to other non-commercial implementations, its performance is competitive. FT-MPI has many adjustable parameters for enhancing performance such as self-tuning collectives and very efficient handling of sparse user derived datatypes.



STATE MONITORING

The FT-MPI Comminfo display utility can attach to any running FT-MPI application and gives users the ability to monitor the execution status of their applications in detail such as during a failure and recover cycle.

The Comminfo display to the left shows the status of the MPI COMM WORLD communicator for a three task MPI application. The blue color of each task and the light blue color of the communicator indicate that no failures or problems have been detected.

FAULT DETECTION

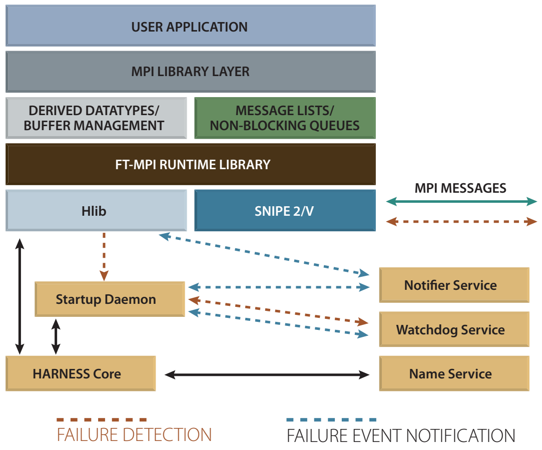
Comminfo display for an application with an exited process prior to any recovery. Task with rank one has exited unexpectedly as indicated by the red color of its status box. At this point a user could click on the status box for further details. The parent communicators status has also changed indicated by the change of color of the outer status box.

FAULT CORRECTION

Comminfo display for an application with an exited process as MPI attempts to replace the failed task with a newly created one. Initially the new task is marked with a white status box as it is started prior to it successfully being added to the incomplete communicator. The location of the new task is decided by the application. The status color of the communicator has changed to indicate that an active recovery is in progress.

FAULT RECOVERY

Comminfo display for an application after a successful recovery. The blue status boxes indicate all processes are executing normally and the outer status box indicates that the communicator is once again valid. Note that the global task identifiers (GIDs) have changed compared to the initial ones above.



HARNESS

HARNESS AT ICL http://icl.cs.utk.edu/harness/

HARNESS (Heterogeneous Adaptable Reconfigurable Networked Systems) is an experimental metacomputing system built around the services of a highly customizable and reconfigurable distributed virtual machine (DVM) that can run on anything from a Supercomputer to a PDA. HARNESS is built on three key areas of research; Parallel Plug-in interface, Distributed Peer-to-Peer Coherent Control algorithms, and Multiple DVM Dynamic Collaboration.

HARNESS is a Department of Energy funded collaborative project involving the University of Tennessee, Emory University, and Oak Ridge National Laboratory.

IN COLLABORATION WITH

EMORY UNIVERSITY

OAK RIDGE NATIONAL LABORATORY

SPONSORED BY

LACSI

SUPPORT FROM

Microsoft

THE UNIVERSITY of TENNESSEE
Computer Science Department

ICL UT
INNOVATIVE COMPUTING LABORATORY

CITR
CENTER for INFORMATION TECHNOLOGY RESEARCH

FT-MPI

FAULT-TOLERANT MESSAGE PASSING INTERFACE

The Heterogeneous Adaptable Reconfigurable Networked System (HARNESS) is an experimental metacomputing system built around the services of a highly customizable and reconfigurable distributed virtual machine (DVM) that can run on anything from a supercomputer to a PDA. A DVM is a tightly coupled computation and resource grid that provides a flexible environment to manage and coordinate parallel application execution.

The HARNESS system implemented by the Innovative Computing Laboratory (ICL) at the University of Tennessee is based on a very lightweight multi-threaded architecture and is aimed at supporting low latency parallel execution of dynamically loadable code, as well as being a research platform for higher level plug-ins. To assist users migrating to the HARNESS system, two plug-ins have been developed that allow direct access to current message passing libraries. Emory University has developed a PVM plug-in that allows a PVM application to execute on top of the Emory H2O framework. ICL has developed a new implementation of MPI known as Fault Tolerant MPI (FT-MPI), which makes the advantages of the HARNESS system's robustness available from within MPI applications. FT-MPI is also capable of executing under the ORNL developed HARNESS system.

FT-MPI is an independent implementation of the full MPI 1.2 message passing standard that has been built from the ground up with both user and system level fault tolerance. In normal implementations of MPI, when a failure occurs the application enters an unspecified state, which the system resolves by halting the entire application. In the case of very large or long running applications this can be very expensive even when checkpointing is employed. FT-MPI however allows developers the ability to build fault tolerant or survivable applications that do not immediately exit due to the failure of a processor, node, or MPI task. FT-MPI offers a number of failure modes that allow a range of recovery schemes to be used that closely match different classes of parallel applications. FT-MPI is unique as it avoids restarting surviving nodes. In other systems, the only current option is to restart the whole application, which can be very expensive for large applications. Although FT-MPI can allow for automatic restart of applications in case of failures by co-operating with checkpoint libraries, it is really meant to allow development of algorithm level fault tolerant applications. More specifically, some applications adapt to failures by changing algorithms or data distributions. Building such dynamic applications with current MPI implementations is not possible. A number of example fault tolerant applications are distributed with the current release that cover all the different recovery modes supported by FT-MPI.

Although FT-MPI has additional features compared to other non-commercial implementations, its performance is comparable to MPICH (1 & 2) as well as LAM-7. FT-MPI has many adjustable parameters for enhancing performance such as self-tuning collectives and very efficient handling of user derived datatypes. As FT-MPI is a super set of MPI, existing MPI 1.2 applications can execute on FT-MPI without modification. Additionally, a number of MPI-2 features and API bindings (such as C++) are also supported. FT-MPI supports most UNIX/POSIX 32 and 64 bit platforms and also runs natively under Microsoft Windows.

The FT-MPI specification won the "Requirements for HPC Systems Software" award at the International Supercomputer Conference 2004 in Heidelberg.

Many of the subsystems developed within FT-MPI have contributed to a new community open source MPI implementation "Open MPI", a collaboration between Los Alamos National Laboratory, Indiana University, the University of Tennessee, and HLRS Stuttgart. The process fault tolerance within FT-MPI will be added to "Open MPI" as a runtime loadable module.

HARNESS is a Department of Energy (DOE) funded collaborative project involving the University of Tennessee, Emory University and Oak Ridge National Laboratory.

HARNESS AT ICL
http://icl.cs.utk.edu/harness/

HARNESS AT ORNL
http://www.epm.ornl.gov/harness/

HARNESS AT EMORY
http://www.mathcs.emory.edu/harness/

E-MAIL **HARNESS** (ACROSS ALL PARTNER SITES)
harness@cs.utk.edu

FT-MPI AT ICL
http://icl.cs.utk.edu/ftmpi/

Open MPI
http://www.open-mpi.org/

E-MAIL **FT-MPI** (ICL/UT TEAM)
ftmpi@cs.utk.edu

THE UNIVERSITY of TENNESSEE
Computer Science Department

ICL UT
INNOVATIVE COMPUTING LABORATORY

CITR
CENTER for INFORMATION TECHNOLOGY RESEARCH