

Open MPI Tool Interface

Overview

- **For 1.7.x/1.8.x:**
 - Full MPI_T functionality
 - Introduce MCA variable system API: `mca_base_var`
 - Convert all existing references to `mca_base_param` to use the new system
 - Provide thin shim layer for `mca_base_param` calls
 - New MCA framework system: `mca_base_framework`
 - Convert all existing frameworks to the new framework system
- **For 1.9.x/2.0.x:**
 - Remove `mca_base_param` API
 - Update `mca_base_component_t` with two new members: `mca_component_priority`, `mca_project_name`. These will be stashed in reserved in 1.7.x
 - Add warnings about misspelled variable names (ex. `OMPI_MCA_btl=self,sm,openib`)
 - Add variable validation function to `mca_base_component_t`

MCA Variable System Overview

- **Replaces MCA Parameter System**
- **Three Registration Functions:**
 - `mca_base_var_register`: takes 4 names (project, framework, component, variable)
 - `mca_base_component_var_register`: takes an `mca_base_component_t` and a variable name
 - `mca_base_framework_var_register`: takes a `mca_base_framework_t` and a variable name
- **Caller Must Provide Backing Store For Variable**
 - Backing store must be one of: `char **`, `int *`, `bool *`
 - `mca_base_var_set_value` will update the location provided at registration time
 - If the variable backing store is known there is no need to use the API to lookup the value
 - Variable system will free string values on deregistration/finalize

MCA Variable System Features

- **Support For MCA Variable Groups**
 - Equivalent to MPI_T categories
- **Support For Non-Overrideable Variables**
 - Set in OMPI_PREFIX/etc/openmpi-mca-params-override.conf
 - Once set in this file they can not be overridden by any other file value or environment value
 - Print help message if users attempts to set an overridden variable
 - Fixes #75
- **Support For Boolean Variables**
 - Support true/false values for both integer and boolean variables
Ex: OMPI_MCA_foo=true <> OMPI_MCA_foo=1
- **Support For Enumerated Integer Variables**



UNCLASSIFIED

Slide 4

MCA Variable System Features (Cont.)

■ MCA Variable Verbosity Level

- 9 Levels:
 - MCA_BASE_VAR_VERBOSITY_USER_[BASIC|DETAIL|ALL],
MCA_BASE_VAR_VERBOSITY_TUNER_[BASIC|DETAIL|ALL],
MCA_BASE_VAR_VERBOSITY_MPIDEV_[BASIC|DETAIL|ALL]
- Equivalent to MPI_T verbosity
- Need to audit all existing MCA parameters and assign a verbosity. Default will be MCA_BASE_VAR_VERBOSITY_MPIDEV_ALL

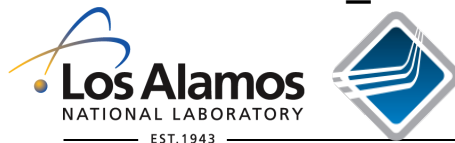
■ MCA Variable Scope

- Hint on the scope of a parameter (constant, read-only, local, group, all)
- Equivalent to MPI_T scope

■ New Variable Source: Command-line

■ MCA Variable Source Environment Variables:

- OMPI_MCA_SOURCE_x=[file:filename|command]



UNCLASSIFIED

Slide 5

MCA Framework System

- **Standardizes Framework registration/open/close**
- **Framework registration/open/close Must Use Framework Functions**
 - `mca_base_framework_register`, `mca_base_framework_open`, `mca_base_framework_close`
 - Framework open/close are reference counted
- **All Frameworks Must Export An `mca_base_framework_t` Structure**
 - Declare with `MCA_BASE_FRAMEWORK_DECLARE`
 - Framework name must be of the form `<project>_<framework>_base_framework`
Ex: `opal_shmem_base_framework`
- **All Framework Member Functions Are Optional**
 - Default register function will register all components
 - Default open function will open all components
 - Default close function will close all components
 - Many existing frameworks can use default functions.



UNCLASSIFIED

Slide 6

MCA Framework System (Cont.)

- **List Of Frameworks Will Be Generated At configure Time**
 - `mca_base_framework_t *project_frameworks[]`; can be found in `project/include/frameworks.h`
- **Framework Registration Flags**
 - Register all components: `MCA_BASE_REGISTER_ALL`
 - Ignores component selection variables (Ex: `-mca btl` would be ignored)
 - Register only static components: `MCA_BASE_REGISTER_STATIC_ONLY`
- **Provides Each Framework With Verbosity and Output**
 - Opened by `mca_base_framework_register/open` (value may change between these functions)
 - Closed on `mca_base_framework_close`
 - Eliminates boiler-plate code
- **Can Be Subclassed**

MCA Framework System (Cont.)

- **New Framework Helper Functions**
 - `mca_base_framework_components_register/open/close`
- **Old Framework Helper Functions Going Away**
 - `mca_base_framework_components_open` replaces `mca_base_components_open`
 - `mca_base_framework_components_close` replaces `mca_base_components_close`
- **We Will Update All Existing Frameworks To The New Infrastructure**
 - Changes will be RFCd as part of the `mca_base_framework` change

MPI Tool Interface

- **Supported From 1.7.x (likely 1.7.1)**
- **Control Variables**
 - Full support through the `mca_base_var` system
- **Categories**
 - Full support through `mca_base_var_group_*`
- **Performance Variable**
 - Initial support using existing `peruse` interface
 - `Peruse` interface will be deprecated in 1.7.x and removed in 1.9.x

Questions? Comments?

- Is There A Need For double Support In mca_base_var?



UNCLASSIFIED

Slide 10

Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NNSA

