Info Query

Point-to-Point WG
December, 2016

Agenda

Objective: Clarify behavior of COMM/WIN/FILE_GET/SET_INFO operations

1. Example scenarios - what should happen?
2. Desired semantics - what is the pattern?
3. Proposed changes - a suggested expression

Example Scenario 1

Code:

```
MPI_COMM_DUP_WITH_INFO(MPI_INFO_NULL);
MPI_COMM_GET_INFO(info);
```

Result:

The returned INFO object will be MPI_INFO_NULL

No key/values were supplied by the user

Example Scenario 2

Code:

```
MPI_COMM_DUP_WITH_INFO(
          INFO{(kBad1,vGood1), (kGood2,vBad2)} );

MPI_COMM_GET_INFO(info);
```

Result:

The returned INFO object will be MPI_INFO_NULL

All key/values supplied by the user were ignored

Example Scenario 3

Code:

```
MPI_COMM_DUP_WITH_INFO(MPI_INFO_NULL);

MPI_COMM_SET_INFO(
        INFO{kGoodAtCreationOnly, vGood1} );

MPI_COMM_GET_INFO(info);
```

Result:

The returned INFO object will be MPI_INFO_NULL

All key/values supplied by the user were ignored

Example Scenario 4

```
MPI_COMM_DUP_WITH_INFO(
        INFO{kGoodAtCreationOnly, vGood1} );
```

```
MPI_COMM_SET_INFO(
        INFO{kGoodAtCreationOnly, vGood2} );
```

```
MPI_COMM_GET_INFO(info);
```

Result:

The returned INFO is INFO{kGoodAtCreationOnly, vGood1}

All key/values supplied by the user in SET were ignored

Example Scenario 5

```
MPI_COMM_DUP_WITH_INFO(
        INFO{kGood, vGood1} );

MPI_COMM_SET_INFO(
        INFO{kGood, vGood2} );

MPI_COMM_GET_INFO(info);
```

Result:

The returned INFO is INFO{kGood, vGood2}

All key/values supplied by the user in SET were not ignored

Example Scenario 6

```
MPI_COMM_DUP_WITH_INFO(
        INFO{kGood, vGood} );

MPI_COMM_SET_INFO(MPI_INFO_NULL);

MPI_COMM_GET_INFO(info);
```

Result:

The returned INFO is:

INFO{kGood, vGood}?

or MPI_INFO_NULL?

Desired Semantics of Comm/Win/File Info Query

1. The key/value pairs returned by MPI_GET_INFO must always have been supplied by the user and not ignored by the MPI library

2. The semantics of supplying an INFO during object creation and during MPI_SET_INFO should be identically defined

3. The MPI library is permitted, but not required, to 'ignore' any INFO key/value pair supplied by the user

4. If an INFO key/value pair is ignored by MPI, then it will not appear in subsequent MPI_GET_INFO calls

5. If an INFO key/value pair was not ignored by MPI, then it must appear in subsequent MPI_GET_INFO calls

Proposal Draft:

https://github.com/mpi-forum/mpi-standard/pull/27

Open Question:

What should happen in a call to SET_INFO if an info key is set on the object and omitted in the info provided to SET_INFO?

Option A) The info key is deleted

Option B) The info key is not modified

Pro/Con:

A) You can delete info keys, at the expense of backward compatibility
B) You can’t ever delete an info key, but you can set it to its default value

B wins the straw poll

Open Question:

Should GET_INFO be allowed to return additional info keys that were not supplied by the user?

A) Yes (9) -- has to be this one because of the “filename” info key
B) No (8)
C) Didn’t vote (Lots)

Assuming that GET_INFO can return additional info keys beyond those supplied by the user -- should we require that *all* info keys defined by the standard are returned in GET_INFO? (so that all MPI libraries have the same behavior)

A) Yes (Most)
B) No (4)