CHAPTER READING
INFO CHAPTER (MISC)

MPI Forum Meeting
December 2020

Enabling Hints

Second paragraph of chapter

Some info hints allow the MPI library to restrict its support for certain operations in order to improve performance or resource utilization. If an application provides such an info hint, it must be compatible with any changes in the behavior of the MPI library that are allowed by the info hint.

Argument Handling

When info is used as an ~~argument to a nonblocking~~ IN or INOUT argument to any MPI routine, it is parsed before that routine returns, so that it may be read, modified or freed immediately after return.

This seems reasonable, but changes the intent

- Do we need to mention “incomplete routines”
- Why not OUT?

Addition of MPI_INFO_GET_STRING

Addition on front page

recognizes a key but not the associated value. MPI_INFO_GET_NKEYS, MPI_INFO_GET_NTHKEY, MPI_INFO_GET_VALUELEN, ~~and~~ MPI_INFO_GET, and MPI_INFO_GET_STRING must retain all (key,value) pairs so that layered functionality can

This function retrieves the value associated with key in a previous call to MPI_INFO_SET. If such a key exists, it sets flag to true and returns the value in value, otherwise it sets flag to false and leaves value unchanged. buflen on input is the size of the provided buffer, for the output of buflen it is the size of the buffer needed to store the value string. If the buflen passed into the function is less than the actual size needed to store the value string (including null terminator in C), the value is truncated. On return, the value of buflen will be set to the required buffer size to hold the value string. If buflen is set to 0, value is not changed. In C, buflen includes the required space for the null terminator. In C, this function returns a null terminated string in all cases where the buflen input value is greater than 0.

If key is larger than MPI_MAX_INFO_KEY, the call is erroneous.

Advice to users. The MPI_INFO_GET_STRING function can be used to obtain the size of the required buffer for a value string by setting the buflen to 0. The returned buflen can then be used to allocate memory before calling MPI_INFO_GET_STRING again to obtain the value string. (*End of advice to users.*)

Addition of MPI_INFO_CREATE_ENV

This routine produces an output object info with the same construction as MPI_INFO_ENV as created during MPI_INIT or MPI_INIT_THREAD when the same arguments are used. This construction is described in Section ??; however, this function can be called when not using the World Model, e.g., when using the Sessions Model. This object is not a direct copy or alias of the MPI_INFO_ENV object and could contain different values based on the input arguments and other sources. Multiple calls to this procedure that are given the same input arguments will produce info objects consistent with the definition of MPI_INFO_ENV. The version for ISO C accepts the argc and argv that are provided by the arguments to `main` or `0` for argc and `NULL` for argv. The user is responsible for freeing the info object via MPI_INFO_FREE. This procedure is local.

This procedure must always be thread-safe, as defined in Section ??. It is one of the few routines that may be called before MPI is initialized or after MPI is finalized.

Advice to users.

In some circumstances (e.g., when passing `0` to argc and `NULL` to argv in C or in Fortran where such arguments do not exist), the info object may not be populated or may be populated incompletely because this procedure is local and the implementation may not be able to determine the correct values. Note that this could result in different values in the resulting info object at different MPI processes.

(*End of advice to users.*)