
SUPV005 - Test of the management of error in MPI

Summary:

This test does not have any physical meaning, it is only one data-processing test.

It aims to check the good performance of the management of the errors in calculations in parallel.

1 Principle of the test

This test checks that one manages to emit errors and to stop the code in parallel properly, including when a processor is not with go of a communication. There are in fact 11 different cases which correspond to modelings of `has` with `K`. All finish in error.

The test must be carried out on 3 processors.

By default, when an error is emitted, it acts of an exception (usual behavior at the time of the studies). If ABORT is indicated, that means that the keyword is used `ERREUR_F=' ABORT'` in `BEGINNING`.

Waiting on a processor simulates a processor which does not answer a communication within the time limit (20 % of time remaining).

The expected behavior is following it for each case:

test	Error on (S) the processor (S)	Waiting on (S) the processor (S)	Expected behavior
has	0		<S>_ERROR The processor #0 announces to the others that they must stop with an exception.
B	1		<S>_ERROR The processor #1 announces that it emitted an exception. The processor #0 announces to the others that they must stop with an exception.
C	2	1	<F>_ERROR The processor #2 announces to the processor #0 that it emitted an exception. The processor #0 does not receive a statute of the processor #1 and stops the execution with <code>MPI_Abort</code> .
D		2	<F>_ERROR The processor #0 does not receive a statute of the processor #2 and stops the execution with <code>MPI_Abort</code> .
E		0	<F>_ABNORMAL_ABORT The processor #1 or #2 (the arrived first) notes that the processor #0 did not receive its statute (ok). He thus takes the initiative to stop the execution with <code>MPI_Abort</code> .
F	2	0	<F>_ABNORMAL_ABORT The processor #1 or #2 (the arrived first) notes that the processor #0 did not receive its statute (ok on #1 or error on #2). He thus takes the initiative to stop the execution with <code>MPI_Abort</code> .
G	1.2	0	<F>_ABNORMAL_ABORT The processor #1 or #2 (the arrived first) notes that the processor #0 did not receive its statute (error). He thus takes the initiative to stop the execution with <code>MPI_Abort</code> .
H	0	1.2	<F>_ERROR The processor #0 receives statute neither processor #1, nor processor #2 and stops the execution with <code>MPI_Abort</code> .
I	2	0.1	<F>_ABNORMAL_ABORT The processor #2 notes that the processor #0 did not

test	Error on (S) the processor (S)	Waiting on (S) the processor (S)	Expected behavior
			receive its statute (error). He thus takes the initiative to stop the execution with MPI_Abort.
J	0 (ABORT)		<F>_ABNORMAL_ABORT As test 0 except that one stops brutally with MPI_Abort.
K	1 (ABORT)		<F>_ABNORMAL_ABORT As test 1 except that one stops brutally with MPI_Abort.

This documentation is voluntarily brief.