Coupling Computer Codes for The Analysis of Severe Accident Using A Pseudo Shared Memory Based on MPI

Young Chul Cho, Chang Hwan Park*, Dong Min Kim
Dept. of Knowledge Eng., FNC Technology Co., Ltd. Heungdeok IT Valley Bldg. 32F, 13, Heungdeok 1ro, Giheung-gu, Yongin-si, Gyoenggi-do, 446-908, Korea
**Corresponding author: ycc@fnctech.com*

1. Introduction

FNC in collaboration with KAERI and KHNP is trying to couple computer codes for the analysis of severe accidents. Recently several computer codes including RELAP & CONTEMPT, SPACE & CAP, etc. are coupled to predict and analyze coupled phenomena in nuclear power plants.

Compared to the coupling for the analysis of severe accident, it is relatively easy to implement the coupling of RELAP & CONTEMPT or SPACE & CAP, as there are two computer codes involved. As there are four codes in-vessel analysis code (CSPACE), ex-vessel analysis code (SACAP), corium behavior analysis code (COMPASS), and fission product behavior analysis code, for the analysis of severe accident, it is complex to implement the coupling of codes with the similar methodologies for RELAP & CONTEMPT or SPACE & CAP. Because of that, an efficient coupling so called Pseudo shared memory architecture was introduced.

In this paper, coupling methodologies will be compared and the methodology used for the analysis of severe accident will be discussed in detail.

2. MPI and Dynamic Linking

2.1 Dynamic Linking

Dynamic linking is a mechanism by which a computer program can, at run time, load a library (or other binary) into memory, retrieve the addresses of functions and variables contained in the library, execute those functions or access those variables, and unload the library from memory. Unlike static linking and loadtime linking, this mechanism allows a computer program to start up in the absence of these libraries, to discover available libraries, and to potentially gain additional functionality.[1][2]

In KNS, most developers could be acquainted with the dynamic linking methodology for the mixed programming with Fortran and C under Windows operating systems. At that time, they used Fortran for the computation, and C for the graphics and graphical user interface. Developers extended the methodology to the coupling of computer codes.

Contrary to the MPI, computer codes coupled with dynamic linking can be told as tight coupled as the developer has to know the details while coupling.

2.2 MPI

MPI, started in 1991, is the result of an effort to make a language-independent communications protocol used to program on parallel computers. MPI is available on major operating systems including UNIX, Linux and MS Windows.[3]

Even though MPI was developed to make parallel programs, it can also be used for computer codes coupling. SPACE/CAP coupling is one of them. Computer codes coupled with MPI can be told as loosely coupled, as the developers don't have to know the internal of each application while coupling. What they have to know are when and what to send/receive data to/from.

3. Computer Codes Coupling Methodologies

The coupling methodologies fall into the following three categories.

- Dynamic linking for RELAP & CONTEMPT coupling
- Point to point communication based on MPI for SPACE & CAP coupling
- Pseudo shared memory based on MPI for the analysis of severe accident

3.1 Dynamic Linking

A simplified flow diagram for the coupling with dynamic library linking methodology is shown in Figure 1.

As can be seen from Figure 1, the slave to be called should be made as a DLL and its input, output and processing part should be called from the master. That means one should know the internals of both computer codes to couple them.

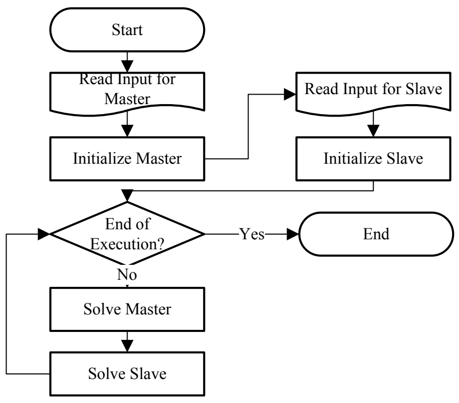

Figure 1: Simplified Flow Diagram with Dynamic Linking

3.2 Point to Point Communication Based on MPI

A simplified flow diagram for the coupling with point to point communication based on MPI is shown in Figure 2.

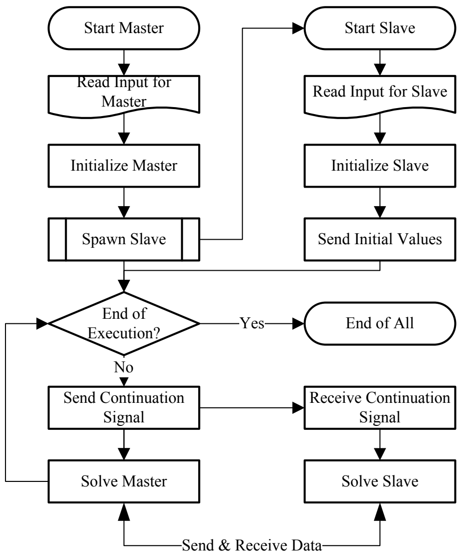

Figure 2: Simplified Flow Diagram with Point to Point Comm.

As two codes are involved in coupling, point to point communication which means direct send and receive communication is adequate for coupling as there is few possibilities of deadlock when sharing data between two processes.

3.3 Pseudo Shared Memory Based on MPI

A simplified flow diagram for the coupling with pseudo shared memory based on MPI is shown in Figure 3.

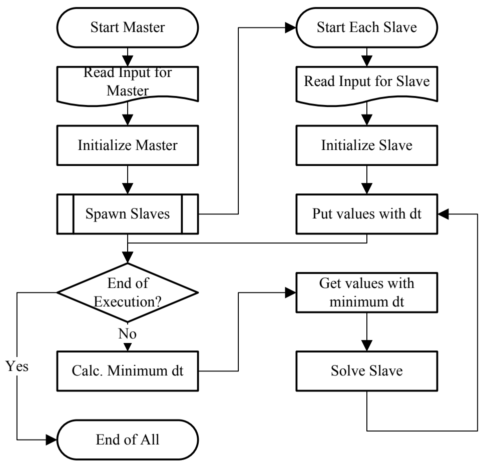

Figure 3: Simplified Flow Diagram with Pseudo Shared Memory

As more than three computer codes are coupled, pseudo shared memory and master process are introduced to remove the complexity. Implementation of coupling with more than three computer codes, the process will easily be in deadlock with direct send and receive communications as there can be mismatching communications.

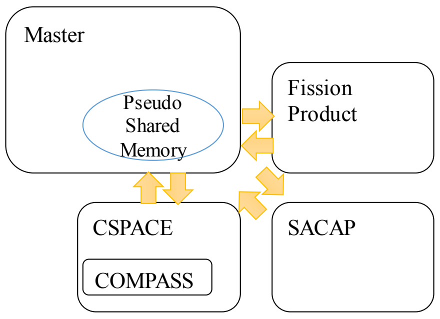

Figure 4: Pseudo Shared Memory Architecture

As can be seen from Figure 4, the memory area of master process can be used as pseudo shared memory to store and process for the rest of the procedures. The dt from each process can be sent to the master process's memory and each slave process (CSPACE, SACAP, and Fission Product) can get the resulting minimum dt calculated by the master process. Each slave process can directly get and put values required for each process as well as if the memory allocated for the master process is the pseudo shared memory for all processes.

What one should be aware is that each process should access its dedicated data on a specific time while coupling with this methodology.

3.4 Pros and Cons of Each Methodology

The pros and cons of each methodology are listed in Table 1. Cons are in slanted letters.

Table 1: Comparison of Coupling Methodologies

Characteristic	Dynamic Linking	Point to Point Comm.	Pseudo Shared Memory
Need to know the details of both codes	*Yes*	No	No
Portability across multiple operating systems	*No*	Yes	Yes
Extensible to parallel computation	*No*	Yes	Yes
Must execute on the same compute	*Yes*	No	No
Maintain separate versions for standalone and coupled execution	*Yes*	No	No
Debugging coupled codes	Simple	*Complex*	*Complex*
Deadlock	Never	*Yes*	No
Easy to transfer data from one program to another	Yes	*No*	Yes
Compile time program error checking	*No*	Yes	Yes

4. Conclusions

The barrier between in-vessel and ex-vessel has been removed for the analysis of severe accidents with the implementation of coupling computer codes with pseudo shared memory architecture based on MPI. The remaining are proper choice and checking of variables and values for the selected severe accident scenarios, e.g., TMI accident.

Even though it is possible to couple more than two computer codes with pseudo shared memory architecture, the methodology should be revised to couple parallel codes especially when they are programmed using MPI.

REFERENCES

[1] Autoconf, Automake, and Libtool: Dynamic Loading (http://sourceware.org/autobook/autobook/autobook_158.html)

[2] Linux4U: ELF Dynamic Loading (http://linux4u.jinr.ru/usoft/WWW/www_debian.org/Documentation/elf/node7.html)

[3] Gropp, William; Lusk, Ewing; Skjellum, Anthony (1996). "A High-Performance, Portable Implementation of the MPI Message Passing Interface". Parallel Computing. CiteSeerX: 10.1.1.102.9485.