ITER IM Technology Workshop
Jun 8-10, 2011, Cadarache, France

SOAF: A Grid-based framework for Integrating Large-scale Long-run Applications

Japan Atomic Energy Agency (JAEA)
Center for Computational Science and e-Systems (CCSE)
Takayuki Tatekawa, Yoshio Suzuki, Hiroshi Takemiya
Naka Fusion Institute
Nobuhiko Hayashi, Isao Kamata

Index

1. Background
2. Current state and our approach
3. Development of Simple Orchestration Application Framework (SOAF)
4. Application
5. Summary

I. Background

- By the development of simulation techniques and computers, various simulations have been carried out.
- By the integration of several simulation codes, we can carry out more detailed simulations.
  For example, in atomic energy research:

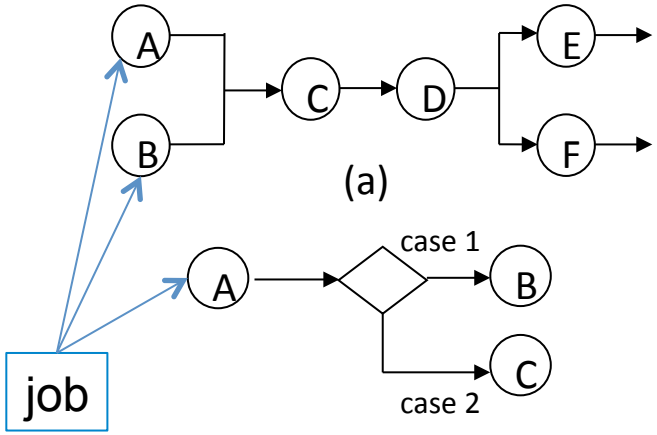
1. Burning Plasma Integrated Code

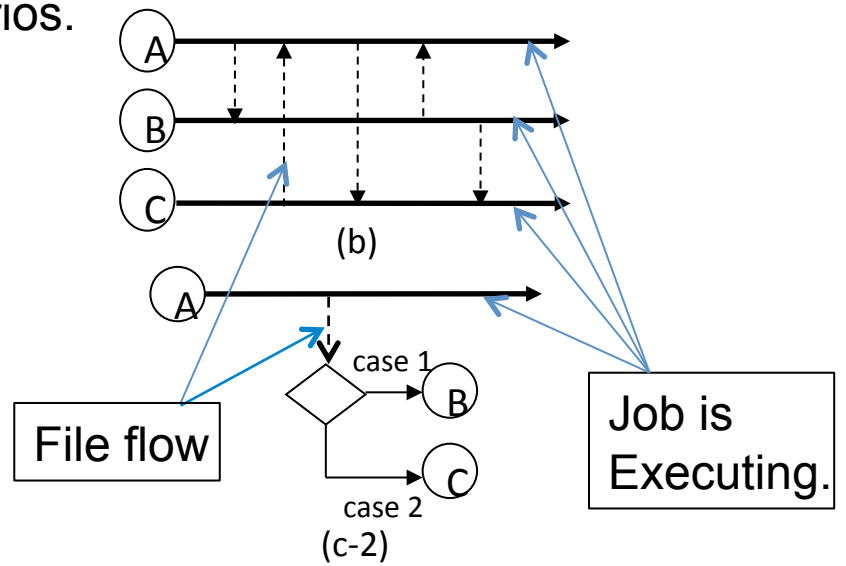
2. 3D virtual plant vibration simulator
   consists of several parts of nuclear reactor (pressure vessel, cooler, pipes).
3. Simulation for Predicting Quake-Proof Capability of Nuclear Power Plants
   consists of codes about earthquake, thermal hydraulics, nuclear system, and so on.

We have developed new framework which can carry out various integrated simulations without substantial change in each simulation code.

II. Current state and our approach

- We roughly categorized the scenarios of cooperative execution of integrated simulation.
  - Here we notice 4 patterns of scenarios.





(a) Sequential type (b) Pipeline type
(c-1) Static conditional branch type (c-2) Dynamic conditional branch type

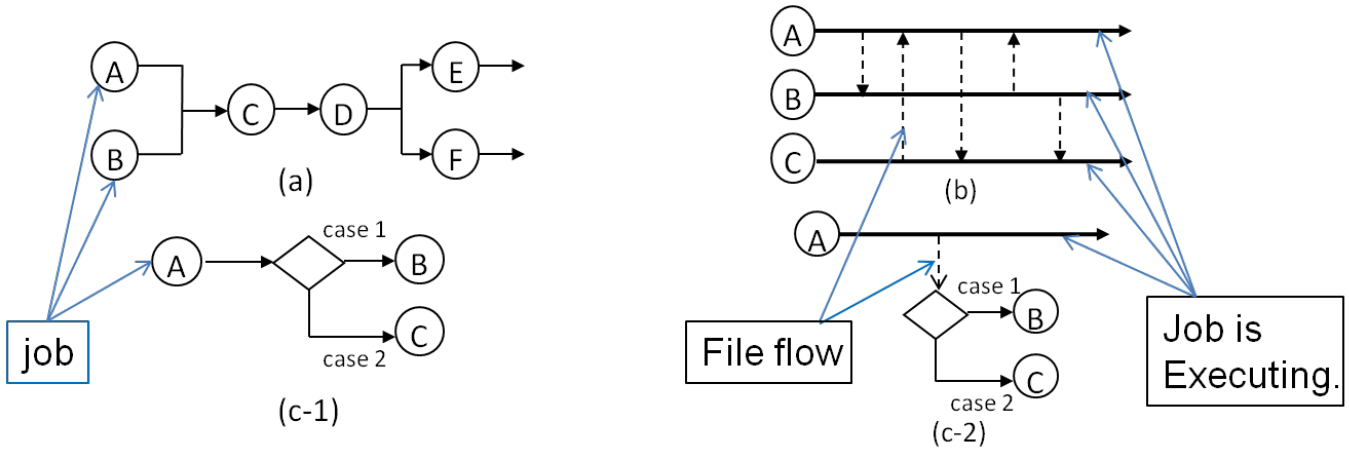
- Current techniques which cooperates simulation codes on grid infrastructure
  - Workflow tool :Kepler, UNICORE workflow engine, TME,.....
  - GridRPC
  - Grid-enabled MPI :STAMPI, MPICH-G, PACS MPI,....

II. Current state and our approach (Problem)

Problem 1 It is hard to make the present techniques adapt themselves to all scenarios.

Workflow tools : Adaptable scenarios are limited.

GridRPC, Grid-enabled MPI : Substantial modification for each code is needed.



Problem 2: It is hard to continue the cooperative execution for long-time.

The integrated simulation is carried out during several days or weeks.

During execution, unexpected outage would be occurred.

-> It is hard to inspect the stop point and restart the cooperative execution.

We have developed the framework which solves these problems.

II. Current state and our approach (Policies)

- Policies for problem solving

For Problem 1

1. Easy to operate various scenarios of cooperative simulation.
2. Minimize the modification of each simulation code for cooperative execution

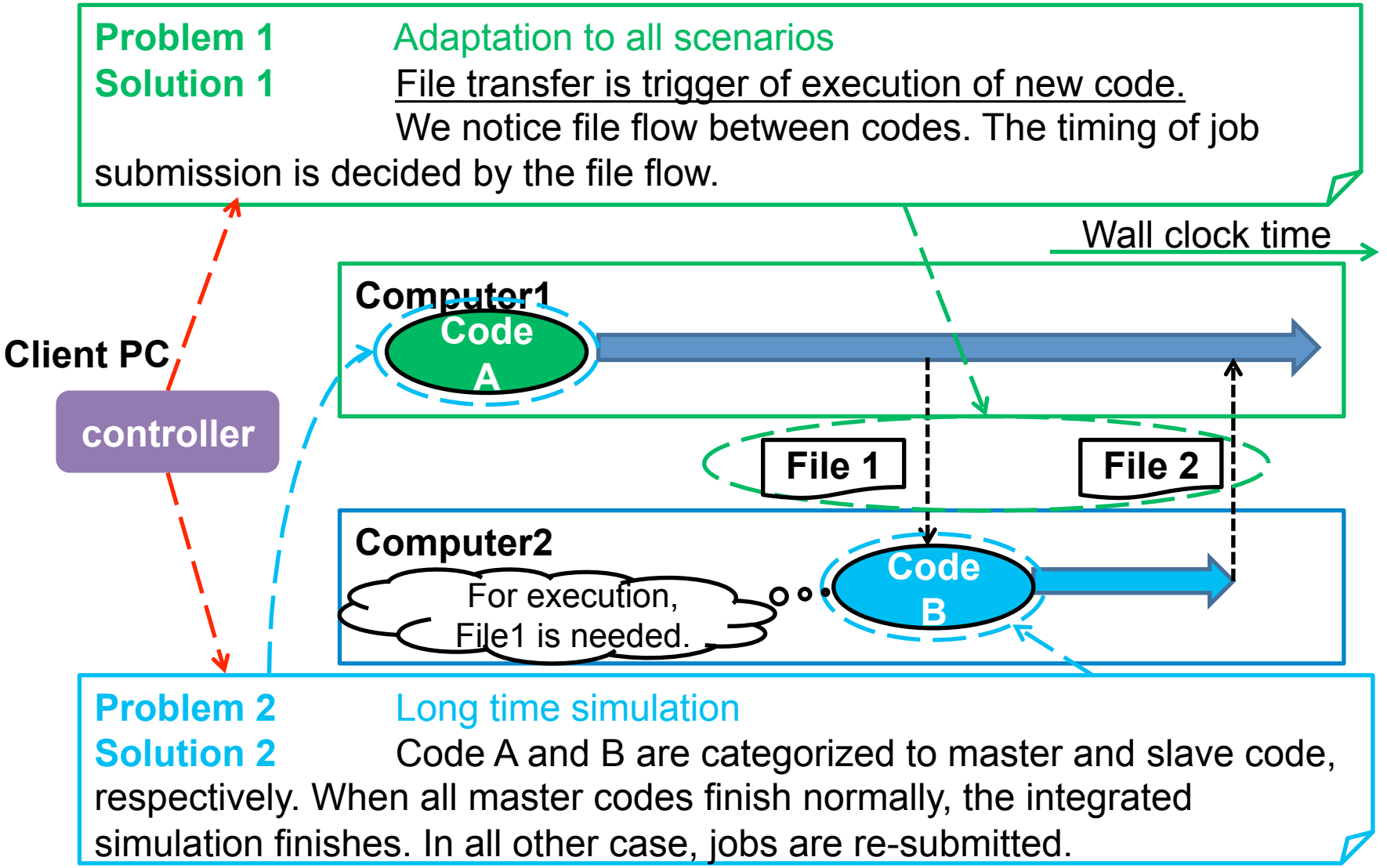
For Problem 2

3. Fault-tolerant mechanism for long-time execution

We propose 2 solutions based on above policies.

II. Current state and our approach (Approach)

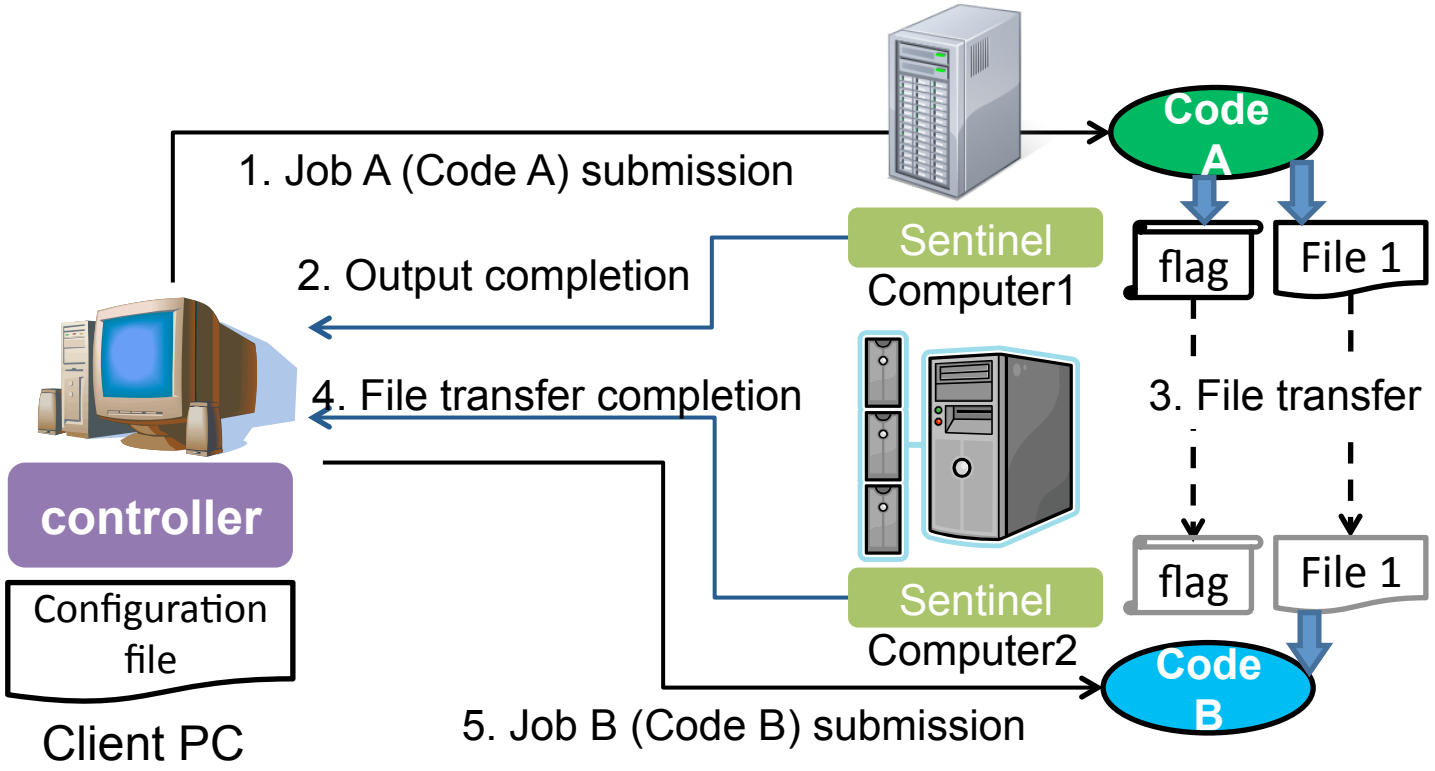
- For example, we consider cooperation of code A and B.



III. Development of Simple Orchestration Application Framework (SOAF)

Solution 1: File flow monitoring function

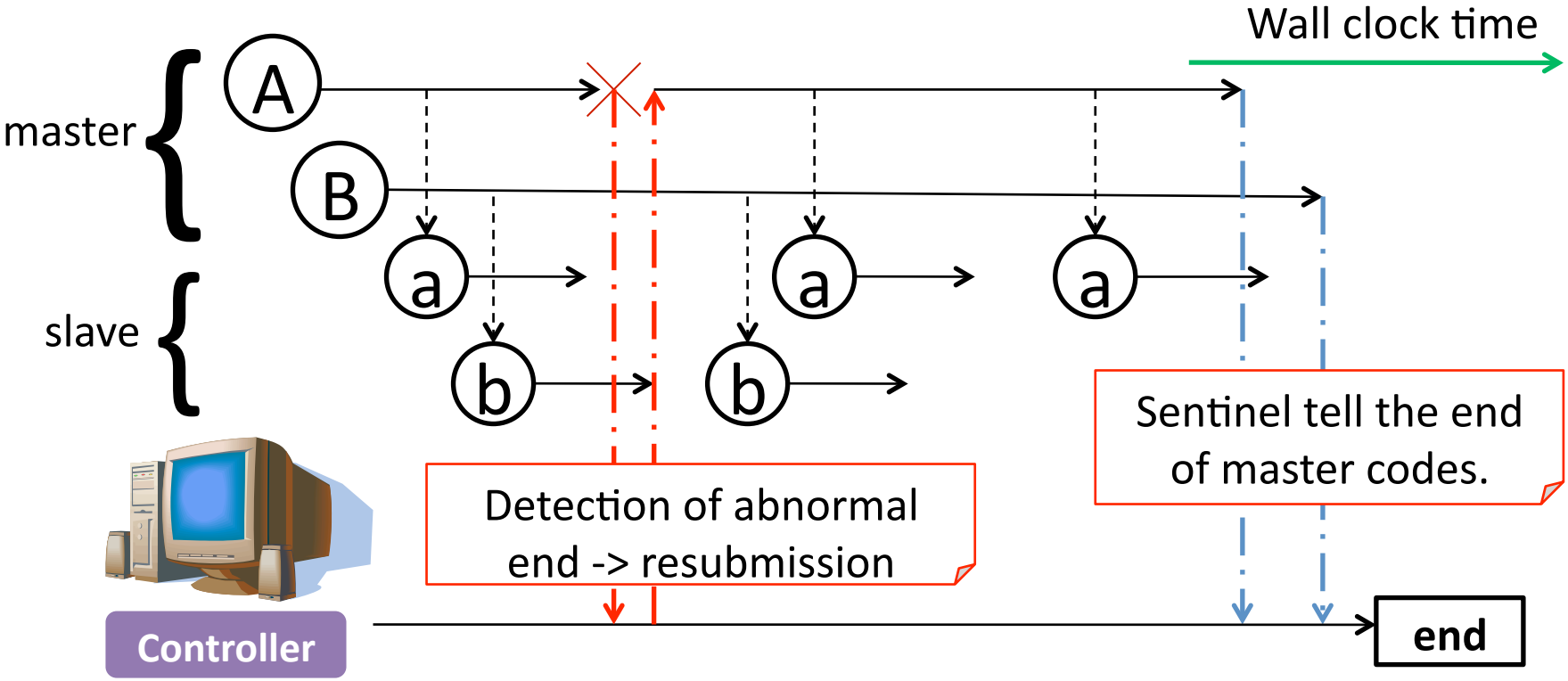
- As a mark of file generation and transmission completion, flag file is generated. Sentinel monitors the flag file.
- When controller on Client PC receives the message from the sentinel, the controller manages file transfer and submits new job.
- The execution codes are required only addition of generation function of flag files.



III. Development of Simple Orchestration Application Framework (SOAF)

Solution 2: Fault-tolerant function

- Slave codes are waiting for execution by the request from master codes. When all master codes finish normally, all job execution finishes.

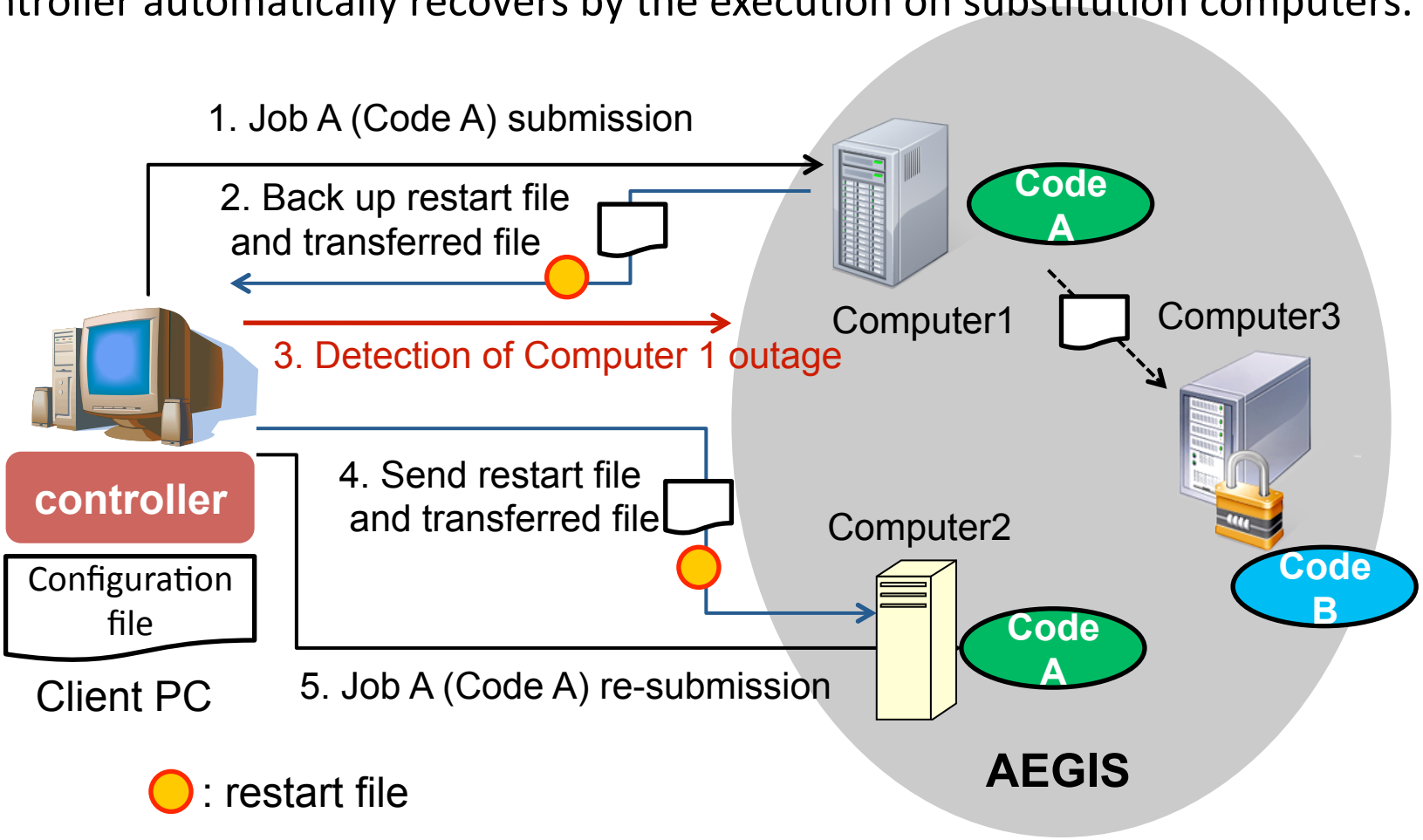


By the detection of abnormal end for any code, the controller re-submits jobs again. We can execute long-time simulation automatically.

III. Development of Simple Orchestration Application Framework (SOAF)

Solution 2: Fault-tolerance function

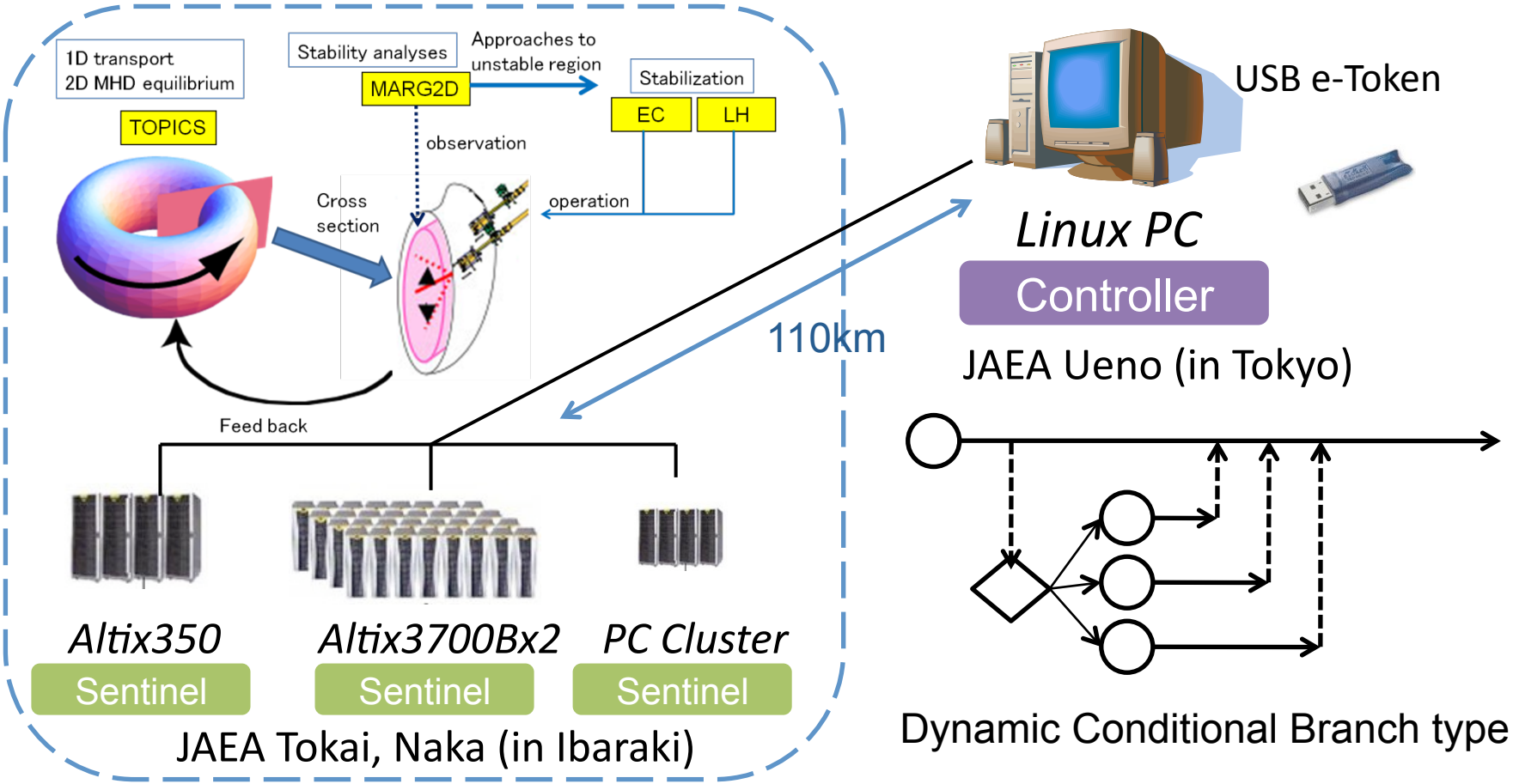
When the cooperative execution is stopped by unexpected outage,
the controller automatically recovers by the execution on substitution computers.



IV. Application - “Burning Plasma Integrated Code”

“Burning Plasma Integrated Code” consists of 4 simulation codes, as an example. These codes are implemented to 3 computers on Atomic Energy Grid Infrastructure (AEGIS).

We verify the file flow monitoring function (Solution 1) with this cooperative execution.

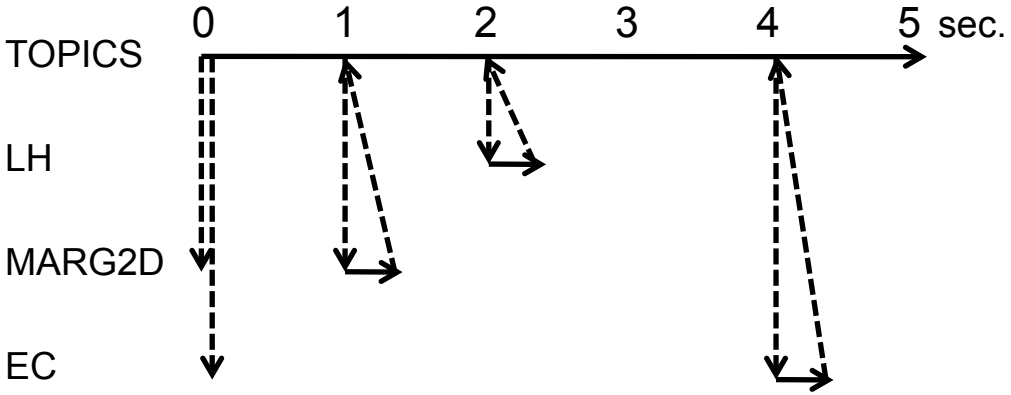


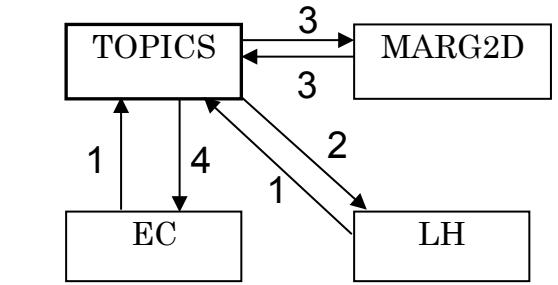
IV. Application - “Burning Plasma Integrated Code”

Scenario 1

Situation of the simulation

Time(sec.)	Events
0	TOPICS starts
1	MARG2D executes
2	LH executes
4	EC executes
5	TOPICS finishes





TOPICS requests execution of other 3 codes.

The files are transferred between TOPICS and other 3 codes.

The number besides arrow means number of files.

The whole codes: about **300,000** lines, modification: about **200 lines**

-> **The modification of codes for cooperative execution is minimized.**

The simulation requires about **40 minutes.** The overhead of SOAF is only **2 minutes.**

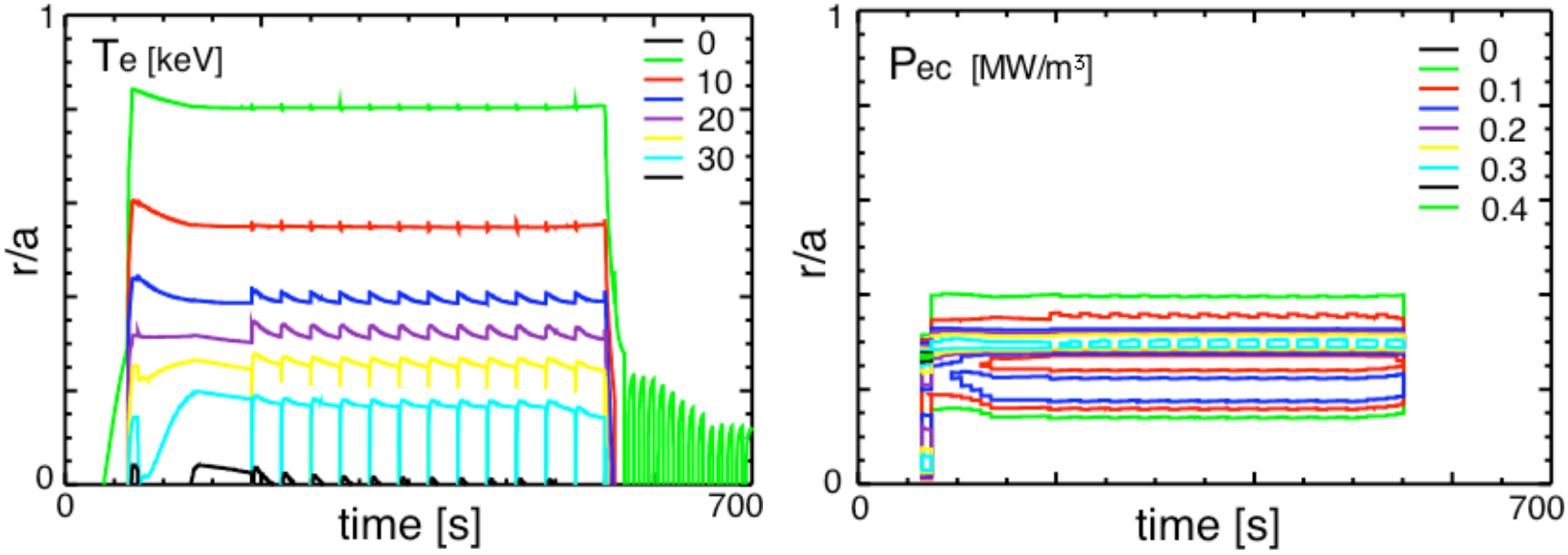
We verified solution 1.

IV. Application - “Burning Plasma Integrated Code”

We applied SOAF to long-time scenario.

Scenario 2

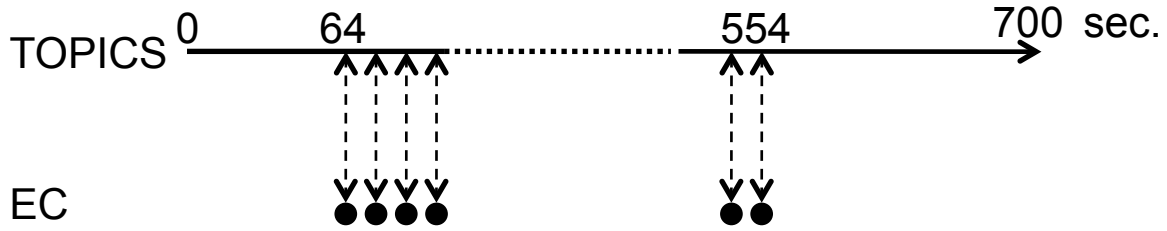
ITER hybrid operation scenario with NB, IC and EC
Almost one day simulation with PC cluster
Only integration between TOPICS and EC code is operated by SOAF.



IV. Application - “Burning Plasma Integrated Code”

Scenario 2

Simulation Time(sec.)	Clock Time (min.)	Events
0	0	TOPICS starts
64	79	EC executes (step 1)
74	112	EC executes (step 2)
…	…	
Every 10 sec.	…	EC executes
…	…	
554	1580	EC executes (step 50)
700	1721	TOPICS finishes



When TOPICS causes abnormal end (interruption), SOAF controller detects the end. Then the controller sends files and resubmits the job during several minutes.

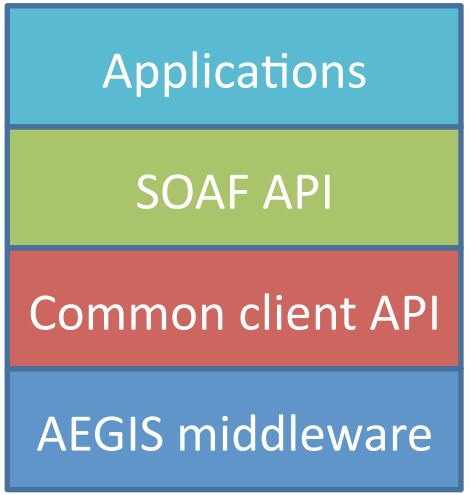
We verified solution 2.

Maintainability, extensibility, and sustainability

- The difficulty of addition of functionalities depends on layer (Grid middleware, API, or applications).
  For example,

SOAF API

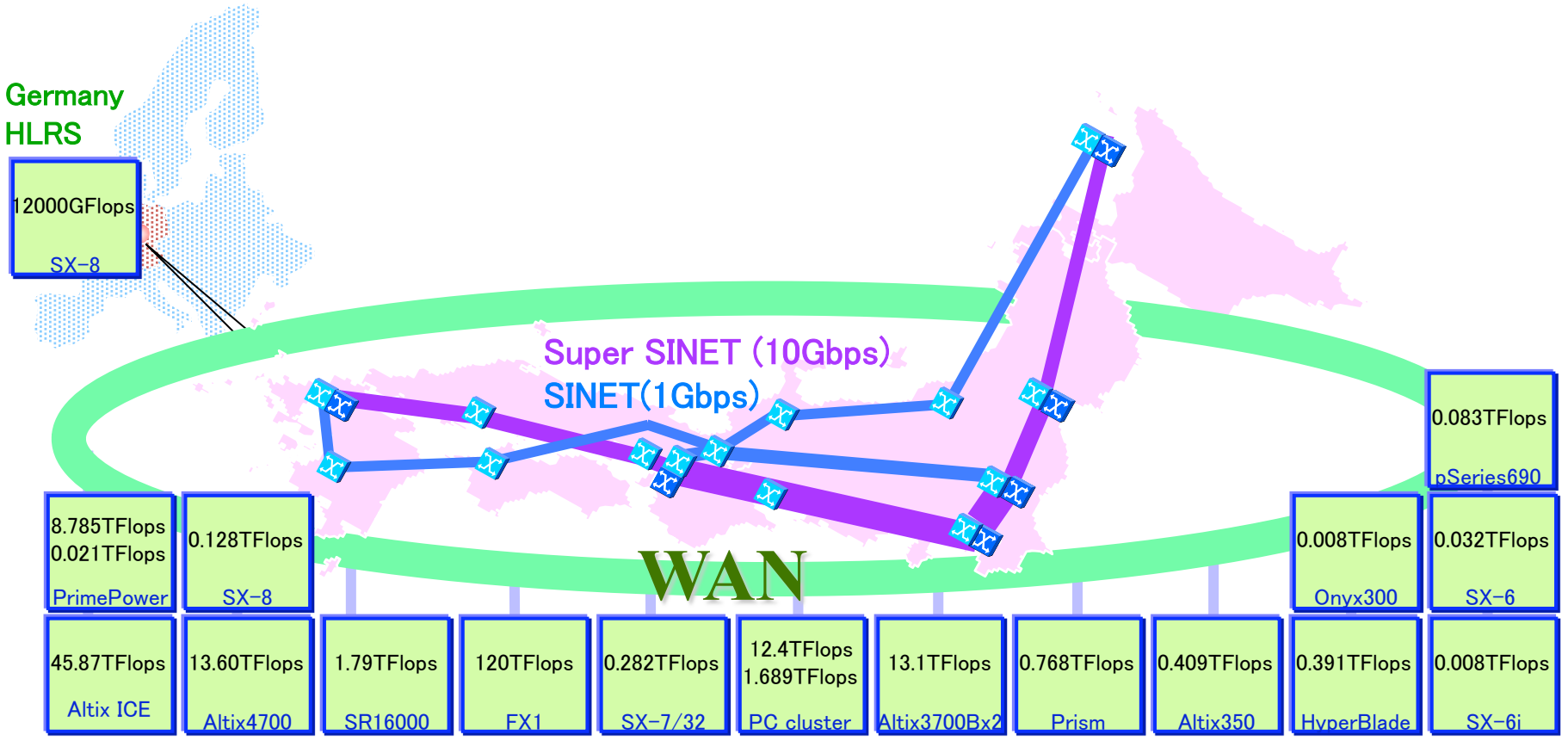
	Controller	Sentinel
Language	C	Perl
Length (lines)	9456	152
Operation System	Linux OS	Linux OS, UNIX



SOAF API uses common client (Authentication, Job management, file transfer) API.

User community

- AEGIS is based on IT-Based Laboratory project (National project, FY2001-).
  64 universities and institutes (JAEA , RIKEN, JAXA, NIMS, NIED, JST etc).
- Budget: FY2001 2.7 billion yen, FY2002 1.1 billion yen, ….
- For maintenance and operation of AEGIS middleware:
  CCSE/JAEA: 3 engineers, other institute: 1 engineer



V. Summary

- We have developed Simple Orchestration Application Framework (SOAF) which adapts various scenarios of cooperative execution without substantial modification for each simulation code.
- From the application to “Burning Plasma Integrated Code”, we verified that the problems has been solved by SOAF.
  1. Various types of cooperative execution is controllable by just definition of a file flow between the simulation codes.
  2. Substantial modification for each simulation code is not needed. (**original codes: about 300,000 lines**, **modification: about 200 lines**)
  3. Job re-submission can be carried out automatically by the fault-tolerance function.
- More following things were indicated from the application result.
  1. The length of the configuration file for the controller is quite shorter than that of original codes. (Scenario 1: about **120 lines**)
  2. The overhead of SOAF is neglectable in whole execution codes. (Scenario 1: **2 min.** **vs** **38 min.)**

Future work

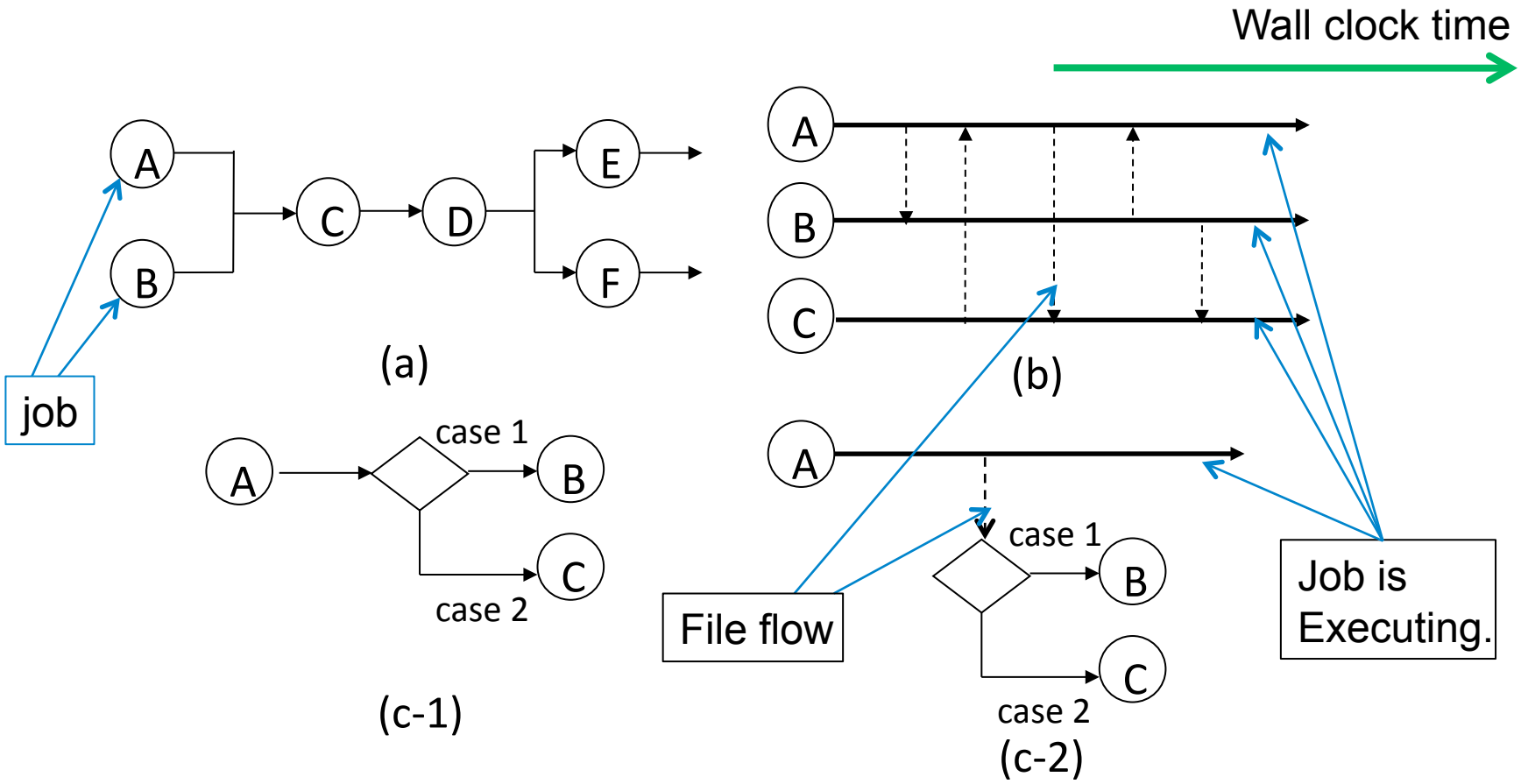
- We implemented fault-tolerant mechanism for job execution.
- Using this mechanism, we have succeeded to execute more than one day simulation.

But,

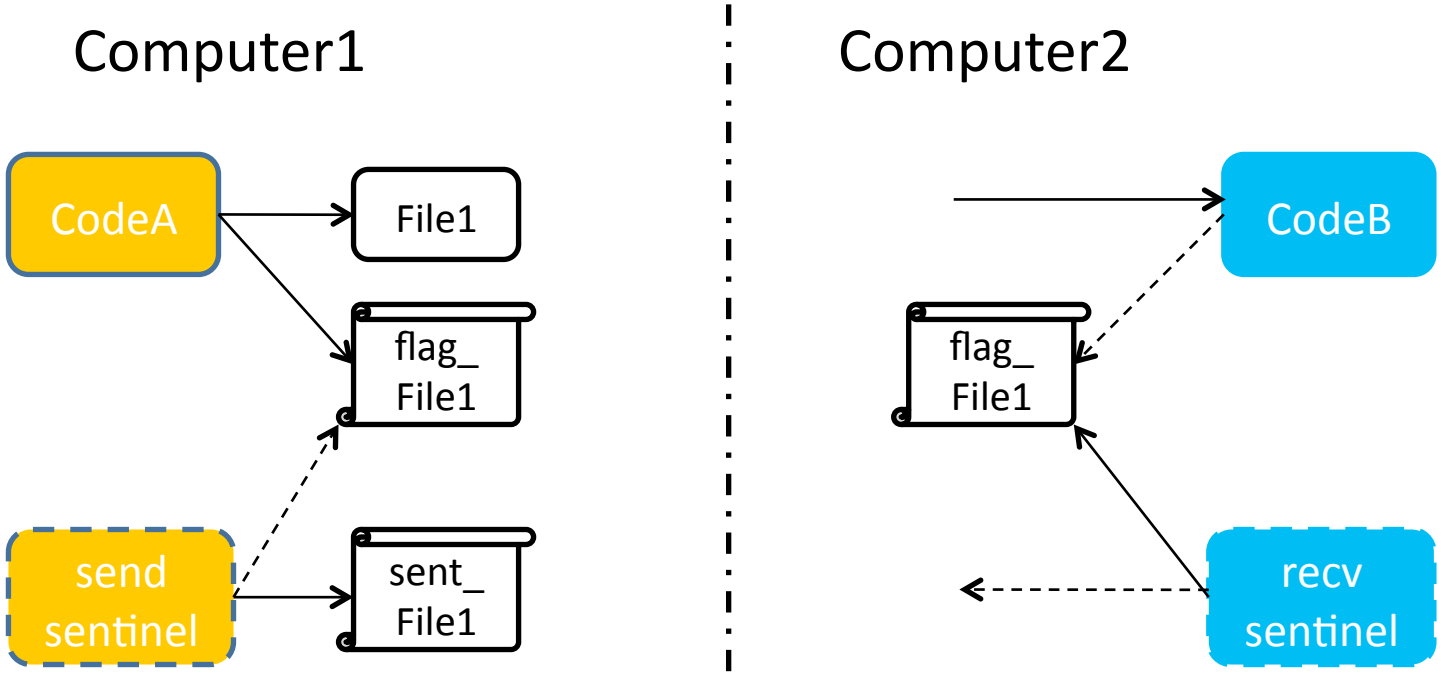
- For long-time simulation, we should notice file transfer between codes. (outage of network during transfer, damage of files)
- We will implement fault-tolerant mechanism for file transfer.

- Our fault-tolerant mechanism cannot avoid error of client PC (controller). -> Reconnection function of client PC to AEGIS is required.

SUPPLEMENTATION


Wall clock time
A
B
C
D
E
F
job
(a)
case 1
A
B
C
case 2
(c-1)
A
B
C
(b)
A
case 1
B
C
case 2
(c-2)
File flow
Job is
Executing.

Procedure of file transfer



Sentinel detects output file and operates file transfer.

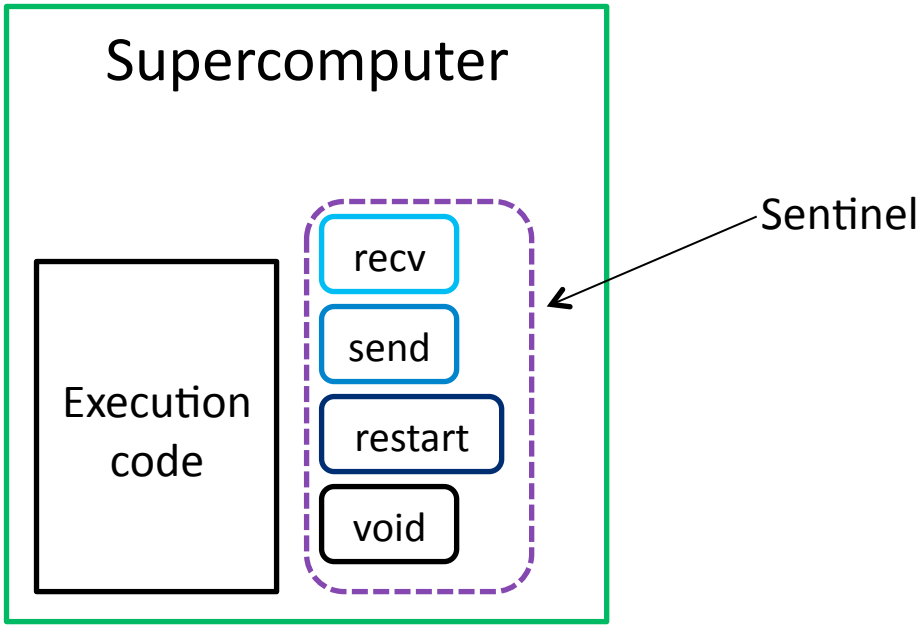
Flag file is a mark of completion of file output and transfer.

For generation and detection of flag files, the modification of the simulation codes is required.

The function of flag files in the simulation codes is supplied by API.

Configuration for supercomputers

- The sentinel is installed beside the execution codes (or the execution scripts).
- The sentinel detects file transfer, preparation of job restart, and job finish.



Configuration file for controller

Configuration file includes information of simulation codes and file flows between codes.

For simple case

```
PROGRAMNUM 2
PROGRAM
   program      codeA
   name         codeA_0
   (info. about codeA)
END

PROGRAM
   program      codeB
   name         codeB_0
   (info. about codeB)
END
```

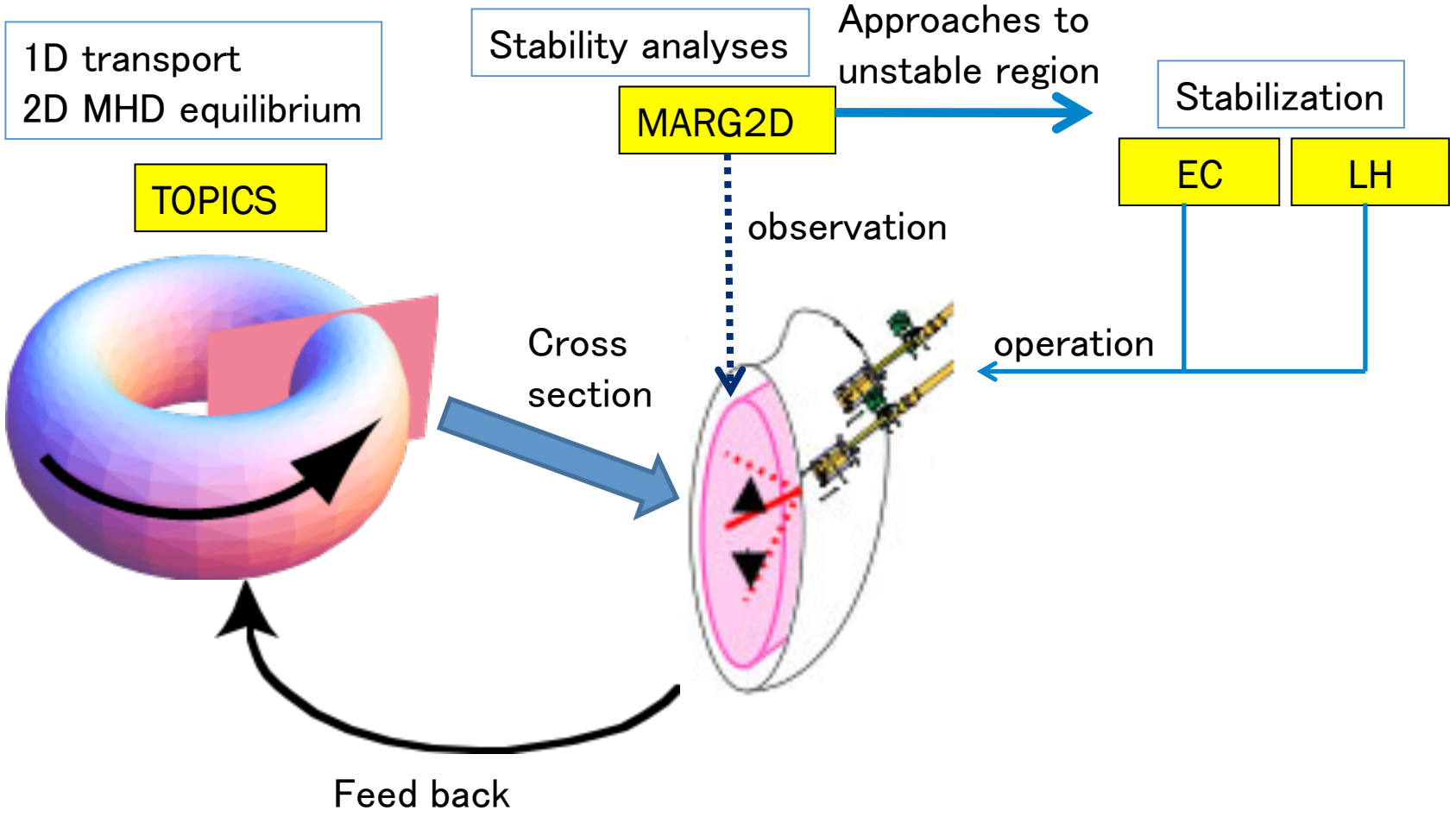
```
FLOWNUM 2
FLOW
   alias        codeA_0
   type         1
   send         File1   codeB_0
   recv         File2   codeB_0
END

FLOW
   alias        codeB_0
   type         0
   recv         File1   codeA_0
   send         File2   codeA_0
END
```

Master code

Slave code

An example of Burning Plasma Simulation

Burning Plasma Simulation codes are developed by
Naka fusion institute, JAEA.

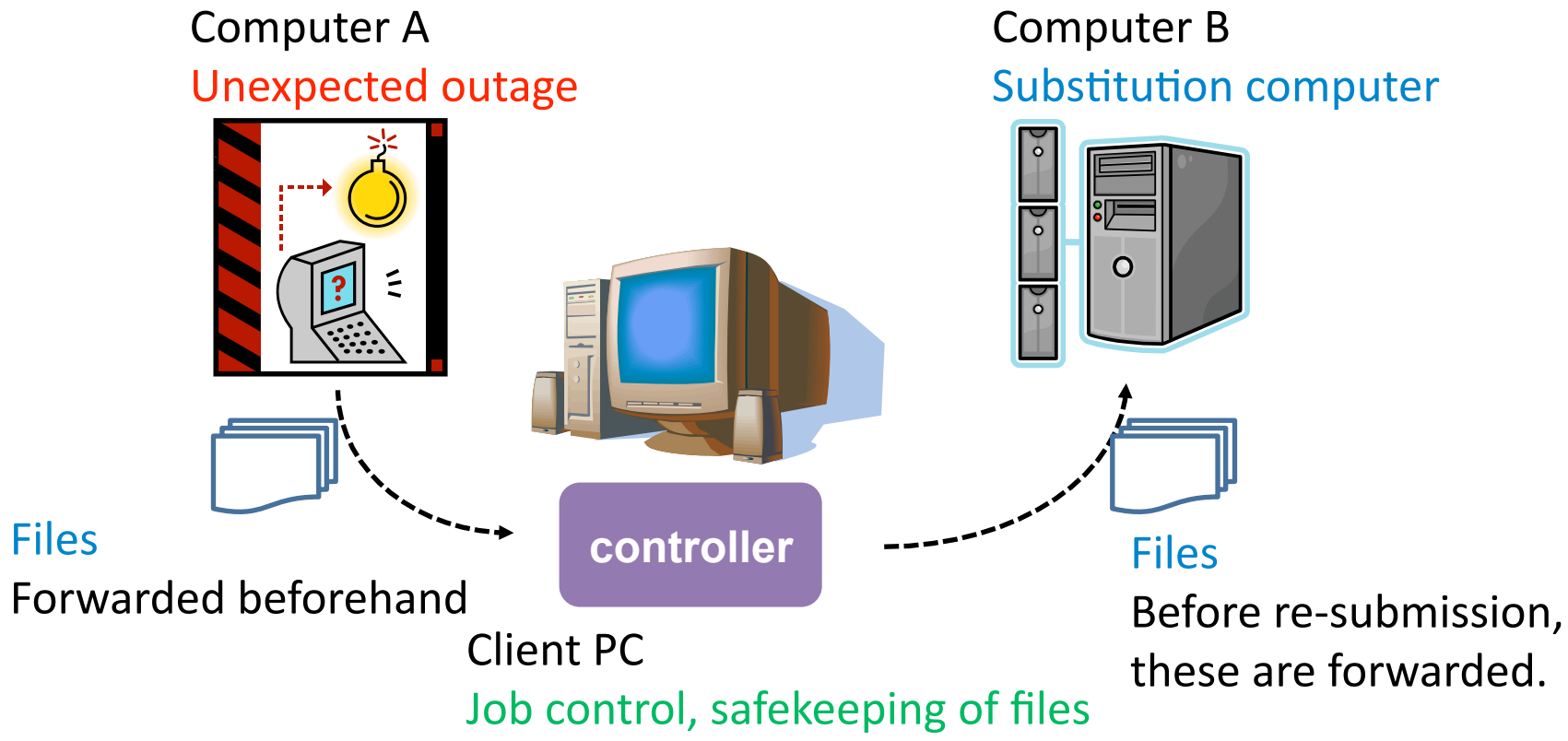


TOPICS requests to execute other codes.

III. Development of Simple Orchestration Application Framework (SOAF)

Solution 2: Fault-tolerance function

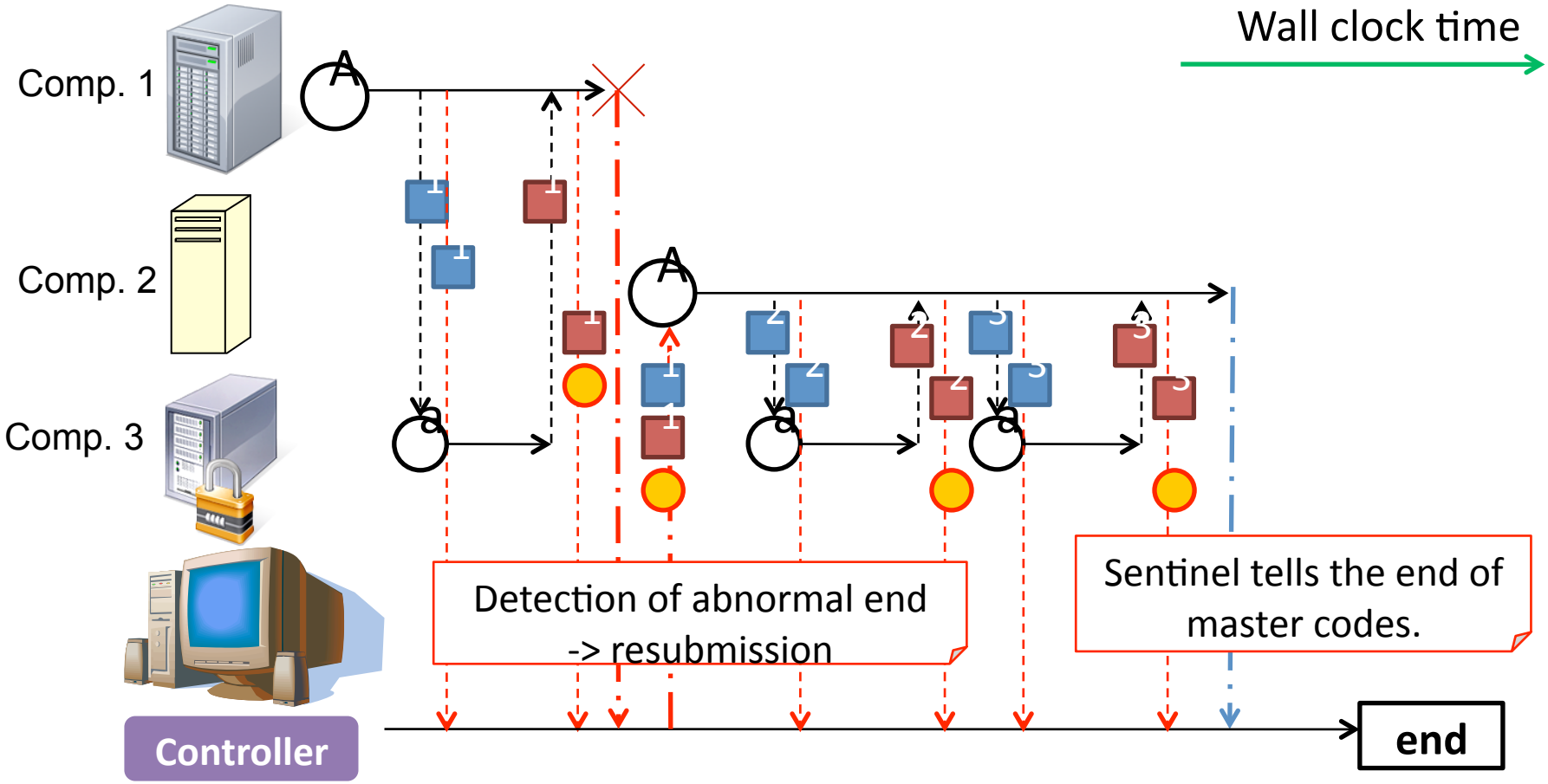
When the cooperative execution is stopped by unexpected outage,
the controller automatically recovers by the execution on substitution computers.



We can handle unexpected outage during long-time simulations.

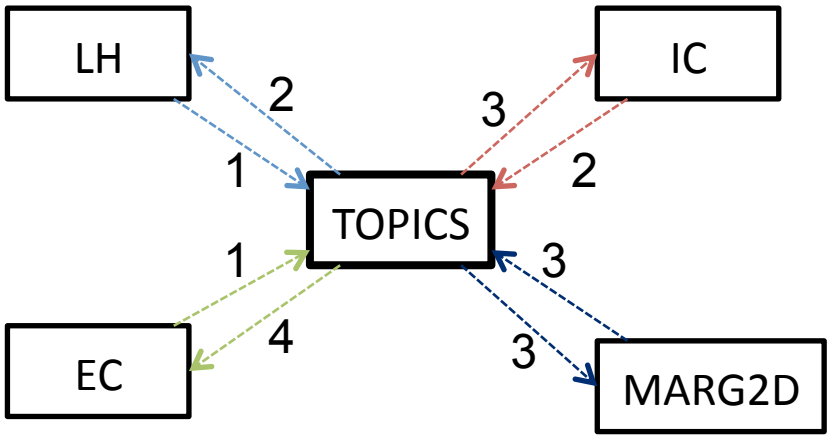
III. Development of Simple Orchestration Application Framework (SOAF)

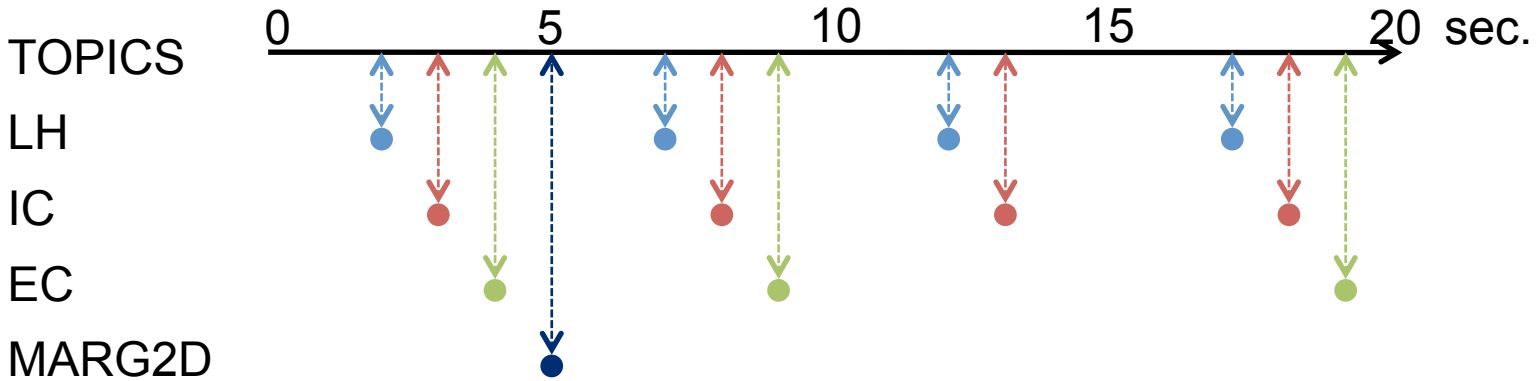
Solution 2: Fault-tolerant function



IV. Application - “Burning Plasma Integrated Code”

Scenario 3

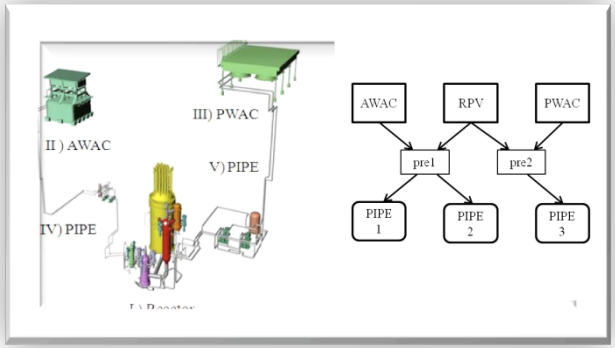


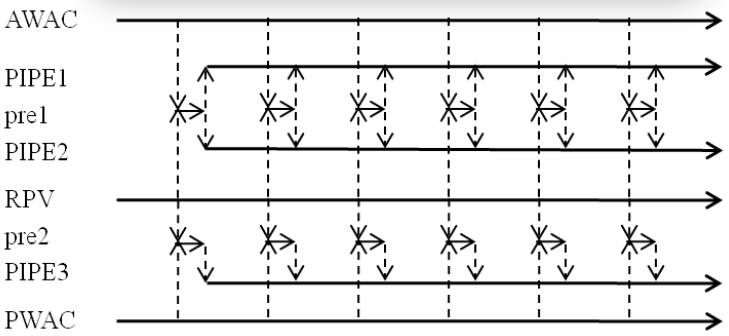


IV. Application –several scenarios-

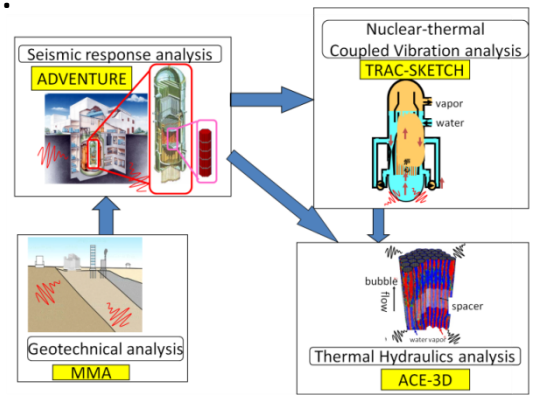
Solution 1: Verification of file flow monitoring function 2

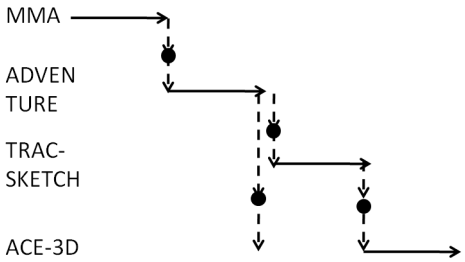
We have verified the function with other scenarios.





3D virtual plant vibration simulator





Simulation for Predicting Quake-Proof Capability of Nuclear Power Plants

For these scenarios, we can execute cooperative operation by the definition of file flows between the codes.

-> **SOAF can be applied to various scenarios.**

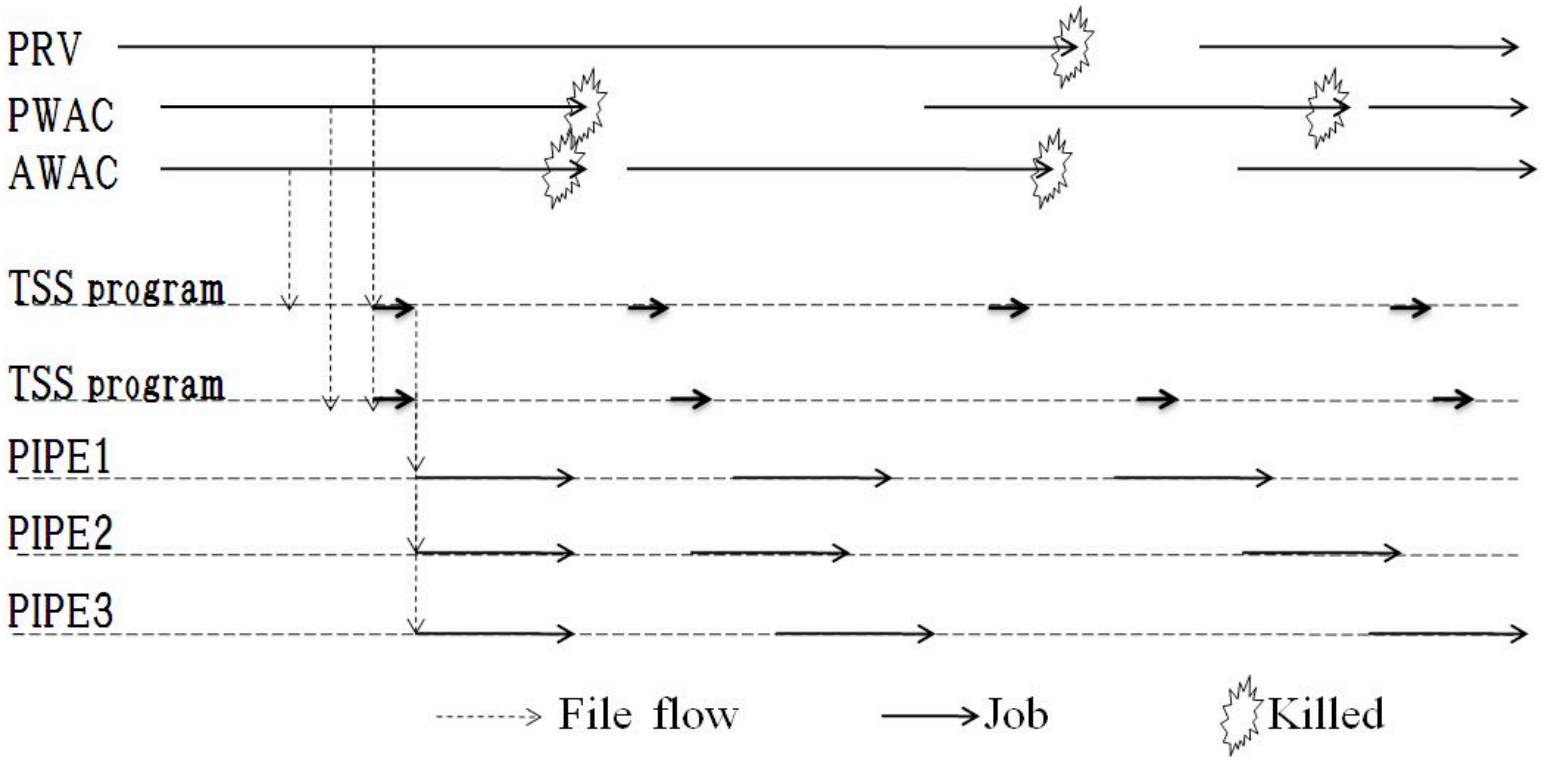
IV. Application –several scenarios-

3D virtual plant vibration simulator

•During the whole simulation time of about 160 hours (100 time steps), all of data files were transferred to their target directories in pipelined scenario.

•Immediate resubmissions of jobs killed due to time limit exceed were done for total simulation period.: RPV(1times), PWAC(3times), AWAC(9times)

Schematic view of jobs flow chart between six job classes and two TSS programs

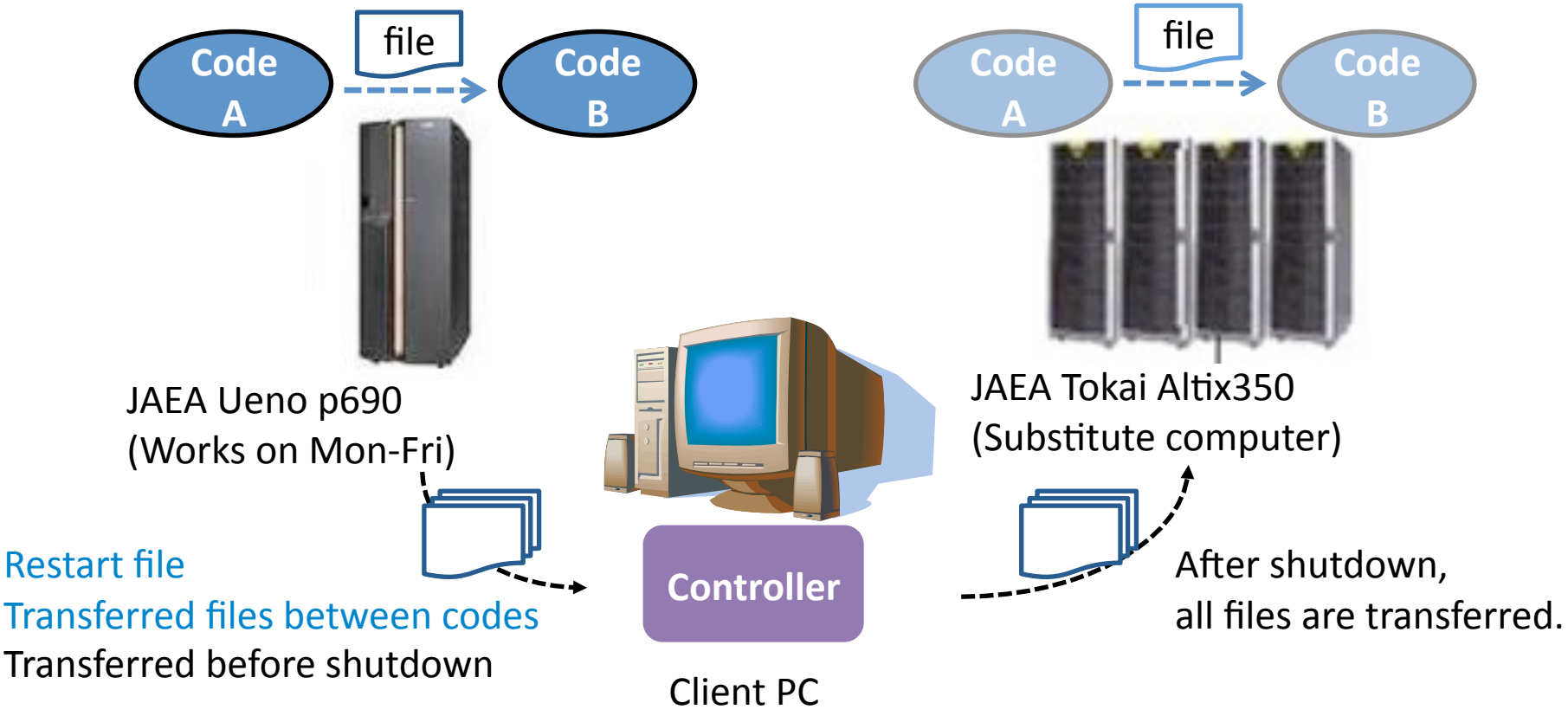


IV. Application –Fault-tolerant function-

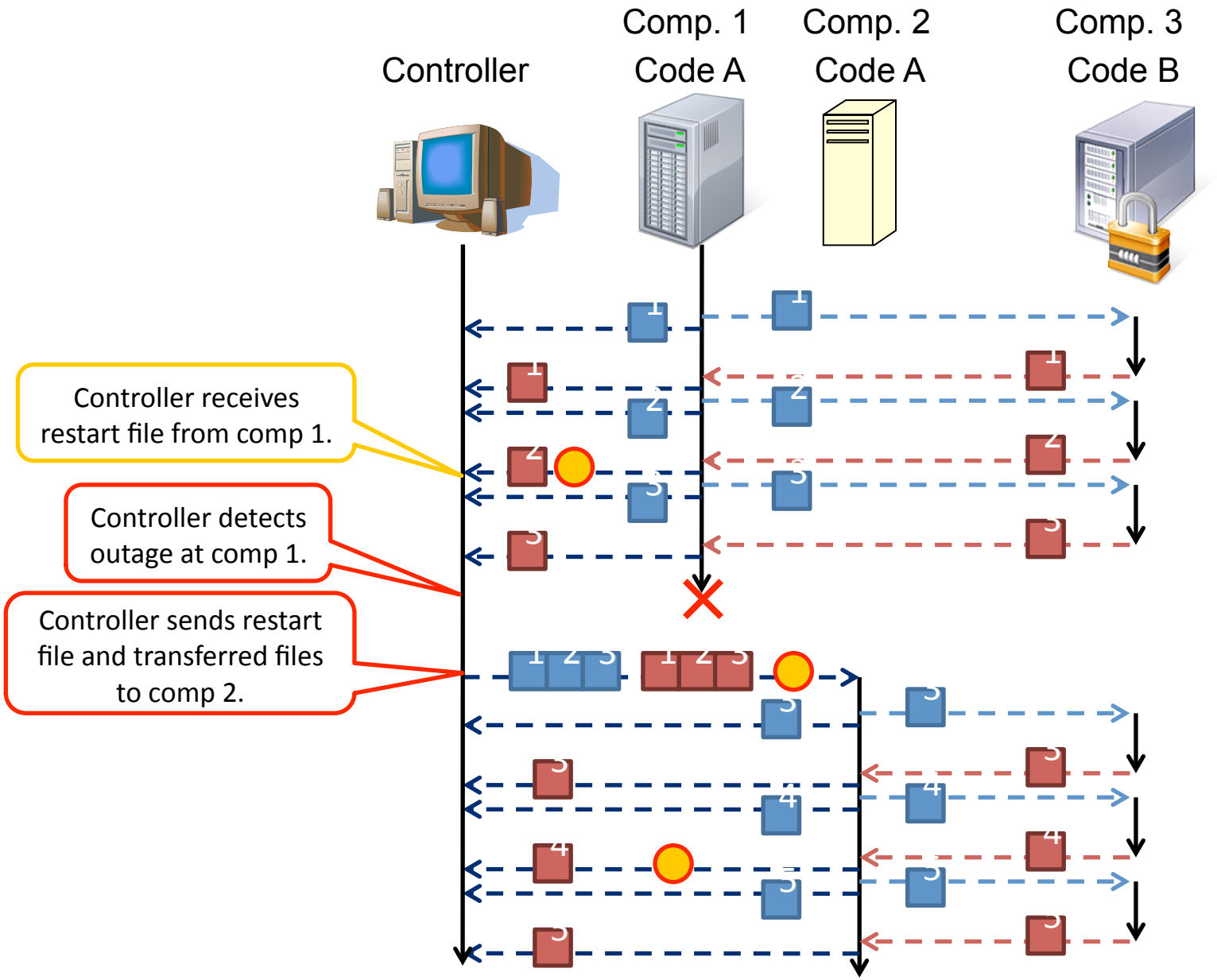
Solution 2: Verification of Fault-tolerant function

We have verified the resubmission of the running job at computer scheduled outage.

-> **We verified that fault-tolerant function works on hardware outage.**



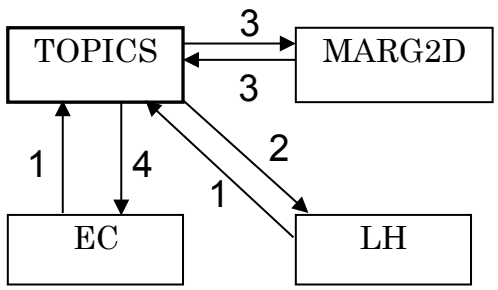
IV. Application –Fault-tolerant function-

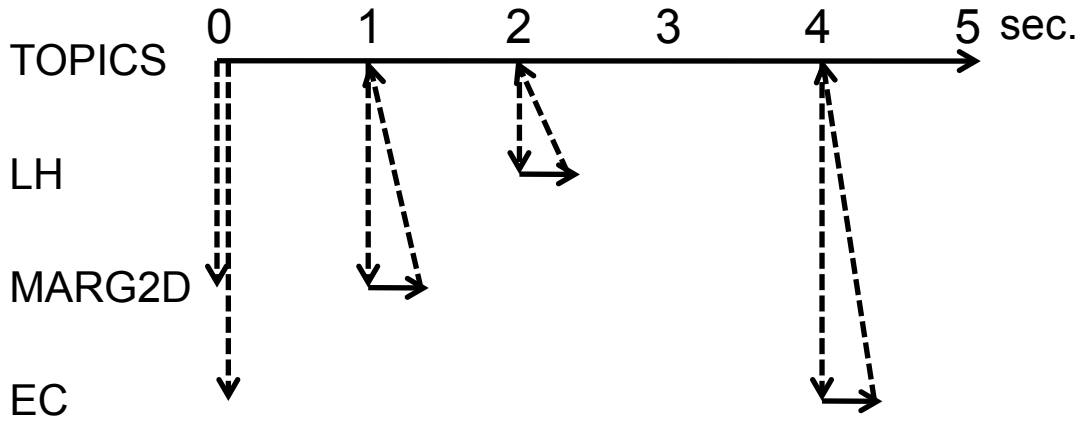


The construction of SOAF

	Controller	Sentinel
Language	C	Perl
Length (lines)	9456	152
Operation System	Linux OS	Linux OS, UNIX

Configuration file (Scenario 1): 125 lines


TOPICS
MARG2D
EC
LH
3
3
1
4
2
1


0
1
2
3
4
5
sec.
TOPICS
LH
MARG2D
EC