

Titolo

IMPLEMENTAZIONE SU CALCOLATORE PARALLELO DI METODI E CODICI PER LE ANALISI DI SENSITIVITA' ED INCERTEZZA NEL CALCOLO DI INCIDENTI SEVERI

Descrittori

Tipologia del documento: Rapporto Tecnico Finale

Collocazione contrattuale: Accordo di Programma ENEA-MiSE su Sviluppo Competenze Scientifiche nel Campo della Sicurezza Nucleare e Collaborazione ai Programmi Internazionali per il Nucleare di IV Generazione.

Argomenti trattati: Reattori ad Acqua Leggera, Incidenti Severi, Analisi di sensitività e incertezza, RAVEN, MELCOR, CRESCO6

Sommario

Nel presente rapporto tecnico vengono riportati i risultati dell'attività di ricerca sviluppata da Sapienza Università di Roma in collaborazione con ENEA nell'ambito del PAR 2018, Accordo di Programma ENEA-MiSE. In particolare, l'implementazione sulla piattaforma CRESCO 6 del software RAVEN è qui descritta. Durante tale attività è stata verificata la corretta installazione di RAVEN, e un caso test di incidente severo, accoppiando RAVEN e MELCOR, è stato condotto. L'analisi dei risultati mostra che i risultati ottenuti sul cluster Sapienza per il caso di analisi di sensitività e valutazione delle incertezze dell'incidente occorso nel 2011 alla unità 1 della centrale nucleare di Fukushima Daiichi, sono consistenti con quelli ottenuti su CRECO 6.

Note

Il presente lavoro è stato preparato con il contributo del personale ENEA e CIRTEN:


Authors: F. Mascari, A. Cervone (ENEA),
F. Giannetti, M. D'Onorio, L. Gramiccia, G. Caruso (Sapienza)
ref. Doc.: CERSE-UNIRM RL 6100/2018




Copia n.

In carico a:

2			NOME			
			FIRMA			
1			NOME			
			FIRMA			
0	EMISSIONE	01/02/19	NOME	F. Mascari	F. Rocchi	F. Rocchi
			FIRMA	<i>F. Mascari</i>	<i>Federico Rocchi</i>	<i>Federico Rocchi</i>
REV.	DESCRIZIONE	DATA		REDAZIONE	CONVALIDA	APPROVAZIONE

 Ricerca Sistema Elettrico	Sigla di identificazione ADPFISS-LP1-125	Rev. 0	Distrib. L	Pag. 2	di 18
--	--	------------------	----------------------	------------------	-----------------

IMPLEMENTAZIONE SU CALCOLATORE PARALLELO DI METODI E CODICI PER LE ANALISI DI SENSITIVITA' ED INCERTEZZA NEL CALCOLO DI INCIDENTI SEVERI

 Ricerca Sistema Elettrico	Sigla di identificazione	Rev.	Distrib.	Pag.	di
	ADPFISS-LP1-125	0	L	3	18

SOMMARIO

1.	INTRODUZIONE	5
2.	DESCRIZIONE DEL SOFTWARE RAVEN.....	6
2.1	Struttura del software	6
3.	PROCEDURA DI IMPLEMENTAZIONE	7
3.1	Compilazione Python in locale e dei pacchetti aggiuntivi necessari.....	7
3.1.1	Default Python.....	8
3.2	Metodologia alternativa per l'installazione di Python e dei pacchetti necessari.....	8
3.3	Compilazione di RAVEN.....	9
3.4	Modifiche dell'interfaccia RAVEN-MELCOR	9
4.	VERIFICA DELL'INSTALLAZIONE	9
4.1	Test interfaccia	9
4.2	Test caso PAR2017	11
4.1	Paragone con precedenti analisi	14
5.	CONCLUSIONI.....	14
6.	RINGRAZIAMENTI	14
7.	ABBREVIAZIONI.....	15
8.	REFERENZE	15



1. INTRODUZIONE

L'analisi deterministica della sicurezza degli impianti nucleari viene utilizzata per caratterizzare analiticamente i fenomeni che si destano in una centrale nucleare durante una progressione incidentale (DBA ebdba) dovuta a postulati eventi iniziatori. In generale, lo spettro di eventi più gravosi viene selezionato per tali analisi (ad esempio un DBA come la rottura a ghigliottina della gamba fredda). BDBA e incidenti severi sono anche essi considerati per valutare le possibili azioni di mitigazione che possono essere prese per prevenire processi di degradazione del core e mitigare il potenziale rilascio di materiale radioattivo [1-4].

In generale l'analisi deterministica della sicurezza degli impianti nucleari [1-5], fornisce le informazioni necessarie per valutare se i requisiti di sicurezza sono soddisfatti da un impianto in condizioni transitorie e, nel caso di una progressione incidentale dovuta a incidenti di tipo severo con fallimento del contenimento, danno anche la possibilità di avere una stima del potenziale termine sorgente. Gli strumenti computazionali, detti anche codici, sono gli strumenti chiave per eseguire queste analisi considerando gli svariati campi di applicazione (ad esempio codice di sistema termoidraulico, codice di incidente severi, ecc.).

Dopo l'incidente di Fukushima, l'interesse di ciascun paese che utilizza l'energia nucleare come parte del proprio mix energetico è stato maggiormente focalizzato sulle strategie di mitigazione degli incidenti di tipo severo [6-12]. Come conseguenza svariate analisi sono state eseguite al fine di caratterizzare le strategie di mitigazione per la gestione di tali incidenti al fine di caratterizzare la loro adeguatezza.

Considerando la complessità e i differenti e svariati fenomeni che si destano durante un incidente di tipo severo e l'eventuale rilascio di materiale radioattivo, le attività di ricerca sugli incidenti severi sono fondamentali per caratterizzare i principali fenomeni che determinano l'evoluzione del transitorio e per supportare la valutazione di SAM. Sotto questo aspetto un ruolo chiave viene svolto dai codici per incidenti di tipo severo come ASTEC [13], MAAP [14] e MELCOR [15], ecc., [16] che, immagazzinando tutte la conoscenza prodotta dalle campagne sperimentali, permettono la caratterizzazione del comportamento in condizioni transitorie di un impianto nucleare durante un incidente di tipo severo caratterizzandolo in termini qualitativi e quantitativi. Diversi modelli e correlazioni sono stati implementati in questi codici. L'eventuale "code-user" deve caratterizzare i parametri che caratterizzano questi modelli durante lo sviluppo dell'input-deck.

Considerando la necessità di ridurre alcune incertezze ancora presenti e considerando il livello di sviluppo e la maturità raggiunta dei codici per incidenti severi, il loro utilizzo insieme a strumenti computazionali per la stima della incertezza dell'analisi è di corrente interesse per la comunità scientifica [17].

In generale I risultati di una simulazione condotta con un codice di calcolo (di tipo termoidraulico e per incidenti severi) sono soggetti a diverse fonti di incertezza: approssimazioni nei modelli implementati nel codice, mancanza di una precisa conoscenza sui parametri dell'impianto e del combustibile, variazioni delle condizioni iniziali e al contorno [18]. Tutte queste potenziali fonti di errore vanno individuate e stimate in termine di influenza sui parametri finali delle analisi. L'analisi delle incertezze può essere condotta attraverso diversi software quali, per esempio DAKOTA [19] e RAVEN [20]. In questo report il software preso come riferimento è il software RAVEN. Il metodo di incertezza considerato per queste analisi è il metodo probabilistico per la propagazione dell'incertezza dei parametri di input [18, 21, 22].

Nei calcoli di tipo deterministico, RAVEN permette di definire i parametri di incertezza da dare come input ad un altro codice che poi condurrà l'analisi dei vari transitori. Tali parametri di input sono derivati da diverse funzioni di densità di probabilità (esempi sono la distribuzione normale e quella uniforme, determinate a seguito di processi di valutazione ingegneristici). RAVEN agisce sull'input del codice che conduce la simulazione del transitorio andando ad inserirvi i valori dei parametri di interesse dopo averli selezionati dalle curve di distribuzione. Tali parametri vengono collocati al posto di specifiche stringhe alfanumeriche inserite ad hoc dall'utente all'interno dell'input. Una volta sostituiti i parametri nomina il file di input in maniera unica usando il valore campionato. A questo punto lancia la simulazione. RAVEN agisce quindi prima dell'inizio della simulazione. Tali operazioni vengono ripetute in loop per il numero di volte deciso dall'utente. Una volta che le simulazioni sono terminate I

file di output sono valutati per quantificare l'incertezza creata sui risultati dalla variazione dei dati di input. Il grande vantaggio legato all'uso di questo codice è la possibilità di lanciare tante simulazioni in parallelo su differenti core, dato che il singolo calcolo di solito ne richiede soltanto uno. Questi metodi, classificati come metodi probabilistici per propagare le incertezze dei parametri di input, necessitano di molta potenza computazionale.

Lo sviluppo di questi nuovi metodi di analisi è stato possibile grazie al continuo aumento della potenza computazionale, che ha permesso la simulazione di progressioni incidentali (DBA and BDBA) con nodalizzazioni sempre più dettagliate in tempi di calcolo sempre minori. La riduzione dei tempi di calcolo, l'utilizzo di piattaforme con svariati processori funzionanti in parallelo e l'utilizzo di software specializzati per effettuare analisi di incertezza, hanno permesso di lanciare in modo automatico svariati simulazioni permettendo, in tempi ragionevoli, di sviluppare analisi di incertezze anche con input-deck complessi. Ciò, abbinato a una analisi di tipo statistica, permette di stimare l'effetto legato all'incertezza dei parametri di input sulla figura di merito scelta per le analisi. La piattaforma utilizzata per queste analisi è CRESCO 6 [23].

2. DESCRIZIONE DEL SOFTWARE RAVEN

RAVEN [20] [24] è un "software framework" (sviluppato presso INL) che agisce come controllore logico per il codice termoidraulico di sistema RELAP-7 [25] [29] [30] (e RELAP5-3D [26]). RAVEN è inoltre un programma per la Probabilistic Risk Analysis che può eseguire diverse operazioni. È stato realizzato per riprodurre la logica di controllo necessaria per simulare sia il sistema di controllo di centrale che il comportamento dell'operatore. Può inoltre essere usato per campionamenti Monte Carlo di distribuzioni randomiche di eventi e per condurre analisi basate sull'albero degli eventi. Per facilitare all'utente la scrittura dell'input e la comprensione dell'output sono stati sviluppati un'interfaccia grafica, Graphical User Interface (GUI), e un modulo di estrazione di dati (data mining) per il post-processing basato sulla riduzione della cardinalità e della dimensionalità.

RAVEN ha quindi il compito di essere un "information manager": deve cioè selezionare ciò che è rilevante per la valutazione del rischio d'impianto. Le tecniche di data mining messe a disposizione dell'utente sono utilizzate per ricercare attentamente le sorgenti di rischio in termini di componenti chiave dell'impianto, fenomeni fisici fondamentali e sorgenti di incertezza. Queste informazioni possono poi essere utilizzate per mettere in atto strategie di mitigazione volte a ridurre le sorgenti di rischio.

2.1 Struttura del software

RAVEN è stato sviluppato seguendo un approccio modulare così da permettere l'integrazione di diversi linguaggi di programmazione (come C++, Python, etc. etc.) e l'accoppiamento con vari codici di sistema. Chiaramente ogni codice di sistema necessita del proprio specifico set di files di interfaccia per permettere l'accoppiamento. Questi files sono piuttosto standard e possono essere realizzati con una conoscenza di base dei linguaggi di programmazione già menzionati. Tre principali software compongono il programma RAVEN, ciascuno dei quali può operare sia in modalità accoppiata che da solo:

- Control Logic System;
- Graphical User Interface;
- Probabilistic and Parametric framework.

Il cuore della capacità di analisi di RAVEN è rappresentato dal probabilistic and parametric framework.

3. PROCEDURA DI IMPLEMENTAZIONE

La procedura utilizzata per la compilazione di RAVEN su CRESCO6 [23] ha richiesto i seguenti passaggi, vista l'impossibilità di creare un VIRTUAL ENVIRONMENT o aggiungere pacchetti nella versione di ANACONDA [27] già presente sul cluster:

- Installazione manuale di Python2 [28];
- Installazione delle librerie necessarie;
- Compilazione di RAVEN;
- Verifica della corretta installazione.

3.1 Compilazione Python in locale e dei pacchetti aggiuntivi necessari

Per il corretto funzionamento del codice, oltre all'installazione in locale di Python 2, è richiesta l'installazione e la configurazione di molti pacchetti necessari. Per questo è stato necessario adattare la procedura di installazione di default, per tenere conto della mancanza di possibilità di accedere come utente root nel cluster. Il tutto è stato installato in una directory condivisa tra i futuri RAVEN users.

Qui di seguito sono riportati i comandi necessari per correttamente configurare Python in locale e i pacchetti necessari:

```

BASE_DIR="/afs/enea.it/project/sicnuc/bin/raven/raven_libs"
INSTALL_DIR="$BASE_DIR/install"
VE_DIR="$BASE_DIR/ve"
mkdir -p $BASE_DIR
cd $BASE_DIR
wget https://downloads.sourceforge.net/project/swig/swig/swig-3.0.12/swig-3.0.12.tar.gz
tar -xvzf swig-3.0.12.tar.gz
cd swig-3.0.12/
./configure --prefix="$INSTALL_DIR"
make
make install
export PATH="$INSTALL_DIR/bin:$PATH"

cd $BASE_DIR
wget http://prdownloads.sourceforge.net/tcl/tcl8.6.6-src.tar.gz
tar -xvzf tcl8.6.6-src.tar.gz
cd tcl8.6.6/unix
./configure --prefix="$INSTALL_DIR"
make
make test
make install

cd $BASE_DIR
wget http://prdownloads.sourceforge.net/tcl/tk8.6.6-src.tar.gz
tar -xvzf tk8.6.6-src.tar.gz
cd tk8.6.6/unix/
./configure --prefix="$INSTALL_DIR"
make
make install

export LD_LIBRARY_PATH="$INSTALL_DIR/lib:$LD_LIBRARY_PATH"

cd $BASE_DIR
wget https://www.python.org/ftp/python/2.7.13/Python-2.7.13.tgz
  
```

```
tar -xvzf Python-2.7.14.tgz
cd Python-2.7.14/
./configure --prefix="$INSTALL_DIR"
make -j16
make install
```

```
cd
pip install --upgrade --target="$INSTALL_DIR" virtualenv
python "$INSTALL_DIR/virtualenv.py" "$VE_DIR"
source "$VE_DIR/bin/activate"
pip install --user numpy==1.11.0 h5py==2.6.0 scipy==0.17.1 scikit-learn==0.17.1
pip install -- user matplotlib
```

3.1.1 Default Python

Per usare di default la versione di Python appena installata, è necessario crearsi un alias nel file `.bashrc` che si trova nella directory di home.

Editare il file `.bashrc` con vim aggiungendo la stringa:

```
alias python='/afs/enea.it/project/sicnuc/bin/raven/python/Python-2.7.14/python'
alias python2='/afs/enea.it/project/sicnuc/bin/raven/python/Python-2.7.14/python2'
alias python2.7='/afs/enea.it/project/sicnuc/bin/raven/python/Python-2.7.14/python2.7'
```

Una volta proceduto con l'installazione di tutti i pacchetti, prima di ogni avvio di RAVEN, è necessario eseguire i seguenti comandi:

```
BASE_DIR="/afs/enea.it/project/sicnuc/bin/raven/raven_libs"
INSTALL_DIR="$BASE_DIR/install"
VE_DIR="$BASE_DIR/ve"
export PATH="$INSTALL_DIR/bin:$PATH"
export LD_LIBRARY_PATH="$INSTALL_DIR/lib:$LD_LIBRARY_PATH"
source "$VE_DIR/bin/activate"
```

3.2 Metodologia alternativa per l'installazione di Python e dei pacchetti necessari

L'iter descritto al par. 3.1 può essere sostituito dall'installazione di una versione di Miniconda2 (versione ridotta di Anaconda [27] compatibile con la versione di SO installata su CRESCO (CENTOS 6.4)).


Questa procedura semplificata è quella consigliata nel manuale del codice, ma in CRESCO presenta delle difficoltà a causa della mancanza del file `/etc/os-release`, la quale non permette l'installazione delle versioni recenti di Anaconda o Miniconda.

Per questo si è dovuto procedere all'installazione di versioni precedenti. A seguito di numerose prove, la versione maggiormente compatibile e testata è stata la versione 3.8.3-Linux-x86_64.

Una volta installata la versione 3.8.3, alcuni pacchetti (in particolare `binstar`) vanno aggiornati manualmente. Successivamente, va digitato il comando:

```
python scripts/TestHarness/testers/RavenUtils.py --conda-create
```

Che a sua volta genera il comando `conda` da eseguire:

 Ricerca Sistema Elettrico	Sigla di identificazione	Rev.	Distrib.	Pag.	di
	ADPFISS-LP1-125	0	L	9	18

```
conda update --name raven_libraries -y h5py=2.7.1 numpy=1.12.1 scipy=1.1.0 scikit-learn=0.19.1
pandas=0.20.3 xarray=0.10.3 netcdf4=1.4.0 matplotlib=2.1.1 statsmodels=0.8.0 python=2.7
hdf5=1.8.18 swig pylint coverage lxml
```

Si riesce in tal modo ad ottenere l'installazione di Python con tutti i pacchetti necessari per la successiva installazione ed esecuzione di RAVEN, richiamabile con il comando:

```
source activate raven_libraries
```

```
<giannetf@cresco6x002 ~> source activate raven_libraries
(raven_libraries) <giannetf@cresco6x002 ~> cd sicnuc/
```

3.3 Compilazione di RAVEN

La procedura di compilazione di RAVEN ha richiesto una breve modifica al sorgente del codice, dato che quest'ultimo originariamente richiedeva la presenza delle ultime versioni di Miniconda, cosa ad oggi non possibile su CRESCO. Per questo l'INL ha provveduto alla modifica della procedura di compilazione del sorgente, con l'aggiunta dell'opzione `--noconda`. Ciò ha permesso l'installazione con successo su CRESCO.

La specifica branch con le modifiche ad hoc è disponibile al seguente indirizzo: <https://github.com/idaholab/raven/pull/893>

Una volta effettuato il merge con la versione devel è necessario aggiungere i pacchetti contenuti in Moose ed infine compilare il tutto con la seguente istruzione:

```
./build_raven --skip-conda
```

3.4 Modifiche dell'interfaccia RAVEN-MELCOR

L'interfaccia RAVEN-MELCOR [24] è stata sviluppata da Sapienza Università di Roma [32] ed è in fase di modifica, con l'aggiunta di nuove funzioni. Per questo, dopo la compilazione, la versione di default di tale interfaccia contenuta nel codice è stata sostituita con l'ultima versione disponibile, non ancora inserita nel sorgente.

4. VERIFICA DELL'INSTALLAZIONE

Per procedere alla fase di verifica dell'installazione non è stato possibile far girare in automatico tutti i test previsti dal codice. Il comando da eseguire, infatti, ha la necessità dell'aggiunta dell'opzione `--skip-conda`, per evitare i controlli di default fatti dal codice che nel nostro caso generano errori. Tale opzione non è ancora disponibile nella versione attuale del codice. Perciò, dopo la prova manuale dei singoli test riguardanti le parti di codice di interesse per i run accoppiati con MELCOR[31][32][33], si è verificata la possibilità di fare rigirare lo stesso input sviluppato anche nell'ambito del PAR 2017 [22].

```
./raven_framework --skip-conda input_file.xml
```

4.1 Test interfaccia

Si è innanzitutto proceduto con la verifica del test di riferimento riguardante l'interfaccia MELCOR-RAVEN, che, come visibile in Fig. 4.1 e Fig. 4.2 ha dato esito positivo.


```
( 0.40 sec) MonteCarlo : DEBUG -> Continuing with r
duced-size Monte-Carlo sampling.
( 0.40 sec) MonteCarlo : DEBUG -> =====
( 0.40 sec) MonteCarlo : DEBUG -> END SUMMARY
( 0.40 sec) MonteCarlo : DEBUG -> =====
( 0.40 sec) STEP MULTIRUN : Message -> *** Run finis
hed ***
( 0.40 sec) STEP MULTIRUN : Message -> *** Closing the
step ***
( 0.40 sec) STEP MULTIRUN : Message -> *** Step clo
sed ***
( 0.40 sec) SIMULATION : Message -> -- End step testMel
cor of type: MultiRun --
( 0.40 sec) SIMULATION : Message -> -- Beginning step d
umpResults of type: IOStep --
( 0.40 sec) STEP IOCOMBINED : Message -> *** Beginning init
ialization ***
( 0.40 sec) STEP IOCOMBINED : DEBUG -> jobHandler initiali
zed
( 0.40 sec) STEP IOCOMBINED : DEBUG -> for the role Output
the item of class OutputStreamPrint and name dumpMelcorHistorySet has been initia
lized
( 0.40 sec) STEP IOCOMBINED : DEBUG -> for the role Output
the item of class OutputStreamPrint and name dumpMelcorPointSet has been initiali
zed
( 0.40 sec) STEP IOCOMBINED : Message -> *** Initializati
on done ***
( 0.40 sec) STEP IOCOMBINED : Message -> *** Beginning
run ***
( 0.40 sec) STEP IOCOMBINED : Message -> *** Run finis
hed ***
( 0.40 sec) STEP IOCOMBINED : Message -> *** Closing the
step ***
( 0.40 sec) STEP IOCOMBINED : Message -> *** Step clo
sed ***
( 0.40 sec) SIMULATION : Message -> -- End step dumpRes
ults of type: IOStep --

-----
There were 3 warnings during the simulation run:
(1 time) No pivot parameter provided; defaulting to "time".
(1 time) There were 3 failed runs! Run with verbosity = debug for more details.
(1 time) current working dir /afs/enea.it/project/sicnuc/bin/raven/raven/tests/f
ramework/CodeInterfaceTests/MelcorInterface/testMelcor already exists, this migh
t imply deletion of present files
-----
( 0.40 sec) SIMULATION : Message -> Run complete!
```

Fig. 4.2 Schermata di output (bash CRESCO) ottenuta nel test

4.2 Test caso PAR2017

Come secondo step di verifica, si è proceduto a far rigirare lo stesso caso analizzato nel PAR 2017 su cluster Sapienza [17], [22].

4.1 Paragone con precedenti analisi

Ripetuto lo stesso calcolo del PAR 2017 si è proceduto al paragone dei risultati ottenuti. La figura di merito selezionata per questa analisi è la pressione nel drywell, già discussa e presentata in [22]. Sono state analizzate soltanto le prime 10 ore del transitorio.

Come si vede dal grafico seguente, l'andamento dei parametri calcolati segue perfettamente quello ottenuto con la stessa simulazione (e lo stesso numero di casi) girato sul cluster Sapienza. Questo ci da garanzie sul corretto funzionamento sia di RAVEN che dell'interfaccia RAVEN-MELCOR implementata su CRESCO.

L'errore percentuale massimo riscontrato tra le due simulazioni è dell'ordine dello 0.08%, prendendo a riferimento la pressione nel drywell e la relativa banda di incertezza.

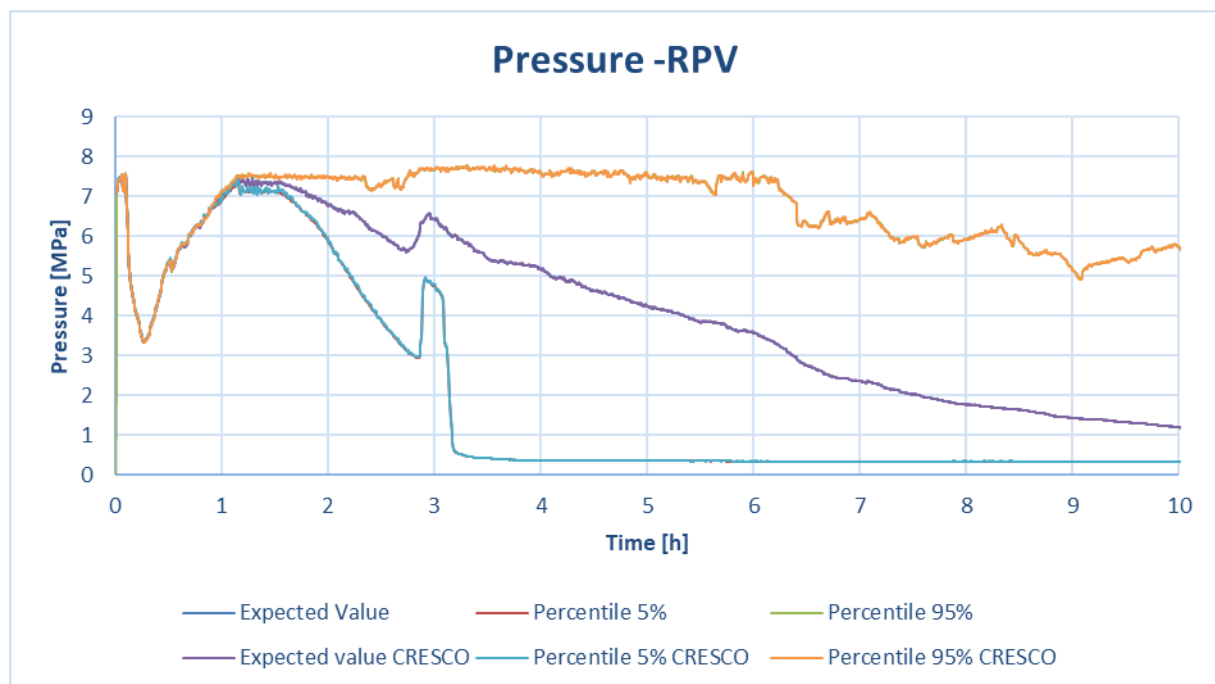



Fig. 4.5 Pressione nel drywell dell'unità 1 di Fukushima e relativa incertezza calcolata con RAVEN

5. CONCLUSIONI

La presente attività è consistita nell'installazione, configurazione e test del codice RAVEN sulla piattaforma CRESCO6. L'attività ha permesso di verificare la corretta installazione, ottenendo praticamente gli stessi risultati, ottenuti sul cluster Sapienza, per il caso di analisi di sensitività e valutazione delle incertezze dell'incidente occorso nel 2011 alla unità 1 della centrale nucleare di Fukushima Daiichi, trattato nel report della scorsa annualità [22], ottenuta con simulazioni MELCOR guidate da RAVEN, rendendo disponibile per la prosecuzione delle attività congiunte l'utilizzo di CRESCO per questo tipo di simulazioni, particolarmente gravose da un punto di vista computazionale.

6. RINGRAZIAMENTI

The computing resources and the related technical support used for this work have been provided by CRESCO/ENEAGRID High Performance Computing infrastructure and its staff [34]. CRESCO/ENEAGRID High Performance Computing infrastructure is funded by ENEA, the Italian National Agency for New Technologies, Energy and Sustainable Economic Development and by Italian and European research programmes, see <http://www.cresco.enea.it/english> for information"

 Ricerca Sistema Elettrico	Sigla di identificazione	Rev.	Distrib.	Pag.	di
	ADPFISS-LP1-125	0	L	15	18


Si ringrazia il Dr. Andrea Alfonsi dell'Idaho National Laboratory, che ha contribuito al successo dell'attività grazie alle modifiche implementate nel sorgente di RAVEN per permetterne il corretto funzionamento nel cluster.

7. ABBREVIAZIONI


ASTEC	Accident Source Term Evaluation Code
BDBA	Beyond Design Basis Accident
BEPU	Best Estimate Plus Uncertainty
BWR	Boiling Water Reactor
DBA	Design Basis Accident
INL	Idaho National Laboratory
LBLOCA	Large Break Loss Of Coolant Accident
LFW	Loss of Feedwater
LOCA	Loss Of Coolant Accident
LWR	Light Water Reactor
MAAP	Modular Accident Analysis Program
MELCOR	Methods for Estimation of Leakages and Consequences Of Releases
NPP	Nuclear Power Plant
PWR	Pressurized Water Reactor
RPV	Reactor Pressure Vessel
RAVEN	Risk Analysis Virtual Environment
RN	RadioNuclide
SAM	Severe Accident Management
SBLOCA	Small Break LOCA
SBO	Station Blackout
SG	Steam Generator
SO	Sistema Operativo
SNL	Sandia National Laboratories
SOT	Start Of the Transient
USNRC	US Nuclear Regulatory Commission

8. REFERENZE


- [1] F. Mascari, F. Giannetti, M. D'Onorio, G. Caruso, A. Naviglio, Analisi di transitori non mitigati: perdita di refrigerante da piccola rottura in PWR, perdita d'acqua di alimento del GV in PWR e SBO in BWR, ADPFISS-LP1-103, 2017
- [2] Deterministic Safety Analysis for Nuclear Power Plants, IAEA Safety Standards Series No. SSG-2, IAEA, Vienna, 2009
- [3] Approaches and Tools for Severe Accident Analysis for Nuclear Power Plants, Safety Reports Series No. 56, IAEA, Vienna, 2008

 Ricerca Sistema Elettrico	Sigla di identificazione	Rev.	Distrib.	Pag.	di
	ADPFISS-LP1-125	0	L	16	18

- [4] F. Mascari, Code Applications For Accident Progression Analyses And Full Scale Predictive Modelling Challenges In Nuclear Reactor Safety, Modelling in Science and Engineering Seminar, 18 October 2017, DoubleTree by Hilton, Manchester, UK (SEMINAR)
- [5] <https://www.iaea.org/newscenter/news/experts-meet-at-iaea-to-evaluate-computer-codes-for-severe-accidents>.
- [6] Severe Accident Management Programmes for Nuclear Power Plants, IAEA Safety Standards, Safety Guide No. NS-G-2.15
- [7] Bentaïb, H. Bonneville, G. Cénérino, B. Clément, F. Corenwinder, M. Cranga, G. Ducros, F. Fichot, D. Jacquemain, C. Journeau, V. Koundy, D. Leteinturier, D. Magallon, R. Meignen, F. Monroig, G. Nahas, F. Pichereau, E. Raimond, J-M. Seiler, B. Tourniaire, J-P. Van-Dorsselaere; D. Jacquemain, Coordinator; Nuclear Power Reactor Core Melt Accidents, Current State of Knowledge, IRSN, EDP Sciences, France (2015).
- [8] M. Pescarini, F. Mascari, D. Mostacci, F. De Rosa, C. Lombardo, F. Giannetti, Analysis of unmitigated large break loss of coolant accidents using MELCOR code, 35th UIT Heat Transfer Conference (UIT2017), IOP Conf. Series: Journal of Physics: Conf. Series 923 (2017) 012009.
- [9] F. Mascari, J. C. DE LA Rosa Blul, M. Sangiorgi, G. Bandini, ASTEC, MAAP AND MELCOR benchmark code analysis of an unmitigated SBO transient in A PWR-900 like reactor, IAEA Technical Meeting on the Status and Evaluation of Severe Accident Simulation Codes for Water Cooled Reactors”, from 9 to 12 October 2017, IAEA's Headquarters in Vienna, Austria.
- [10] F. Mascari, M. D'onorio, F. Giannetti, G. Caruso, A. Naviglio, Analisi di Transitori Non Mitigati: Perdita di Refrigerante da Piccola Rottura in PWR, Perdita D'acqua di Alimento del GV in PWR e SBO in BWR (Analyses of Unmitigated Transients: Small Break LOCA in PWR, Loss of Feedwater in PWR and SBO in BWR), ADPFISS-LP1- 103, 2017” <http://openarchive.enea.it/handle/10840/9211>
- [11] F. Mascari, Marco Pescarini, F. Giannetti, I. Luciani, G. Caruso, Integral Calculations of Severe Accident Scenarios in PWRs and BWRs, ADPFISS-LP1-075, 2016. <http://openarchive.enea.it/bitstream/handle/10840/8170/ADPFISS-LP1-075.pdf?sequence=1>
- [12] C. Lombardo, F. Mascari, P. Buffa, F. Castiglia, M Giardina, G. Palermo, Nodalizzazione MELCOR per lo Studio Integrale di Sequenze Incidentali su Reattori PWR da 900 MWe e Valutazioni Preliminari D'impatto a Breve e Medio Raggio, ADPFISS-LP1-026, Italy (2014).
- [13] Chatelard P., Belon S., Bosland L., Carénini L., Coindreau O., Cousin F., Marchetto C., Nowack H., Piar L., Chailan L, Main modelling features of ASTEC V2.1 major version, Annals of Nuclear Energy (on Press), January 2016
- [14] Fauske & Associates, LLC “Transmittal Document for MAAP5 Code Revision MAAP 5.02”, FAI/13-0801, (2013).
- [15] MELCOR Computer Code Manuals, Vol.1: Primer and Users' Guide, SAND 2015-6691 R; Vol. 2: Reference Manual, SAND 2015-6692 R; Vol. 3: MELCOR Assessment Problems, SAND 2015-6693 R; Sandia National Laboratories, USA (2015).
- [16] J.P. Van Dorsselaere, M. Barrachin, S. Power, M. Adorni, M. Hrehor, F. Mascari, A. Schaffrath, I. Tiselj, E. Uspuras, Y. Yamamoto, D. Gumenyuk, N. Fedotova, O. Cronvall, P. Liska, The Summary of ETSON strategic orientations on research activities, ETSON Research Group activity, atw Vol. 63 (2018), issue 1 January
- [17] M. D'Onorio, F. Giannetti, F. Mascari, G. Caruso, Uncertainty analyses using the RAVEN software tool coupled with MELCOR severe accident code, ANS Best Estimate Plus Uncertainty International Conference (BEPU 2018) Real Collegio, Lucca, Italy, May 13-19, 2018, BEPU2018-282

 Ricerca Sistema Elettrico	Sigla di identificazione	Rev.	Distrib.	Pag.	di
	ADPFISS-LP1-125	0	L	17	18

- [18] A. Bersano, F. Mascari, Evaluation of a Double-Ended Guillotine Break Transient in a Three-Loop PWR-900 Like with TRACE Code Coupled with DAKOTA Uncertainty Analysis, 12th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics, Operation and Safety (NUTHOS-12) Qingdao, China, October 14-18, 2018
- [19] B.M. Adams, M.S. Ebeida, M.S. Eldred, G. Geraci, J.D. Jakeman, K.A. Maupin, J.A. Monschke, J.A. Stephens, L.P. Swiler, D.M. Vigil, T.M. Wildey, W.J. Bohnhoff, K.R. Dalbey, J.P. Eddy, J.R. Frye, R.W. Hooper, K.T. Hu, P.D. Hough, M. Khalil, E.M. Ridgway, A. Rushdi, Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.7 User's Manual, SAND2014-4633 (2017)
- [20] A. Alfonsi, C. Rabiti, D. Mandelli, J. Cogliatti, C. Wang, P. W. Talbot, D. P. Maljovec, C. Smith, "RAVEN Theory Manual and User Guide," INL/EXT-16-38178, Printed March 2017
- [21] H. Glaeser, "GRS Method for Uncertainty and Sensitivity Evaluation of Code Results and Applications," Sci. Technol. Nuc. Ins. (2008)
- [22] F. Mascari, F. Giannetti, M. D'Onorio, A. Giampaolo, L. Gramiccia, G. Caruso, Analisi e valutazioni di incidenti severi in LWR, ADPFISS-LP1-122, 2018, <http://openarchive.enea.it/bitstream/handle/10840/9823/ADPFISS-LP1-122.pdf>
- [23] <http://www.cresco.enea.it/>
- [24] C. Rabiti, A. Alfonsi, J. Cogliatti, D. Mandelli, R. Kinoshita, S. Sen, C. Wang, J. Chen, "RAVEN User Manual," INL/EXT-15-34123, Printed March 2017
- [25] Ivan Rinaldi, Andrea Alfonsi, Joshua Cogliatti, Cristian Rabiti, Fabio Giannetti, Gianfranco Caruso, A Comprehensive Validation Approach Using The RAVEN Code. 2015 ANS Annual Meeting: "Nuclear Technology: An Essential Part of the Solution"
- [26] Narcisi, V., Lorusso, P., Giannetti, F., Alfonsi, A., Caruso, G. Uncertainty quantification method for RELAP5-3D© using RAVEN and application on NACIE experiments (2019) Annals of Nuclear Energy, 127, pp. 419-432.
- [27] Michele Chambers, VP Products & EVP Anaconda Business, Continuum Analytics. Christine Doig, Data Scientist & Product Manager, Continuum Analytics, Dr. Kristopher Overholt, Ph.D. & Product Manager, Continuum Analytics, Dr. Stanley Seibert, Ph.D. & High Performance Python Team Leader, Continuum Analytics (2016) HIGH PERFORMANCE PYTHON FOR OPEN DATA SCIENCE (WHITEPAPER)
- [28] Python Software Foundation. Python Language Reference, version 2.7. Available at <http://www.python.org>
- [29] C. Rabiti, A. Alfonsi, J. Cogliatti, D. Mandelli, and R. Kinoshita, "REACTOR ANALYSIS AND VIRTUAL CONTROL ENVIRONMENT (RAVEN) FY12 REPORT," Tech. Rep. INL/EXT-12-27351, Idaho National Laboratory (INL) (2012).
- [30] C. Rabiti, A. Alfonsi, J. Cogliatti, D. Mandelli and R. Martineau, "RAVEN as Control Logic and Probabilistic Risk Assessment Driver for RELAP-7," in "Proceeding of American Nuclear Society (ANS), San Diego (CA)," (2012), vol. 107, pp. 333–335
- [31] MELCOR Computer Code Manuals, Vol.1: Primer and Users' Guide, SAND 2015-6691 R; Vol. 2: Reference Manual, SAND 2015-6692 R; Vol. 3: MELCOR Assessment Problems, SAND 2015-6693 R; Sandia National Laboratories, USA (2015).
- [32] "MELCOR Computer Code Manuals", Vol. 2: Reference Manual, Version 2.2.9541 2017, SAND2017-0876 O, January 2017;
- [33] "MELCOR Computer Code Manuals", Vol. 1: Pimer and Users' Guide, Version 2.2.9541 2017, SAND2017-0455 O, January 2017.

 Ricerca Sistema Elettrico	Sigla di identificazione ADPFISS-LP1-125	Rev. 0	Distrib. L	Pag. 18	di 18
--	--	------------------	----------------------	-------------------	-----------------

- [34] G. Ponti et al., "The role of medium size facilities in the HPC ecosystem: the case of the new CRESCO4 cluster integrated in the ENEAGRID infrastructure", Proceedings of the 2014 International Conference on High Performance Computing and Simulation, HPCS 2014, art. no. 6903807, 1030-1033