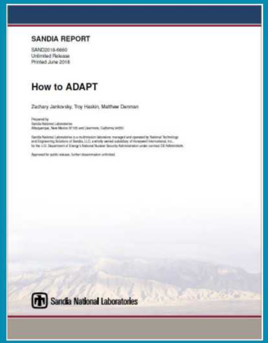
SAND2018-10194C

Recent Analysis and Capability Enhancements to the ADAPT Dynamic Event Tree Driver



SANDIA REPORT

How to ADAPT

Zachary Jankovsky, Troy Haskin, Matthew Denman

Annals of Nuclear Energy

Dynamic event tree analysis with the SAS4A/SASSYS-1 safety analysis code

Zachary Jankovsky[*], Matthew Denman[*], Tunc Aldemir[+]

[*]Sandia National Laboratories [+]The Ohio State University

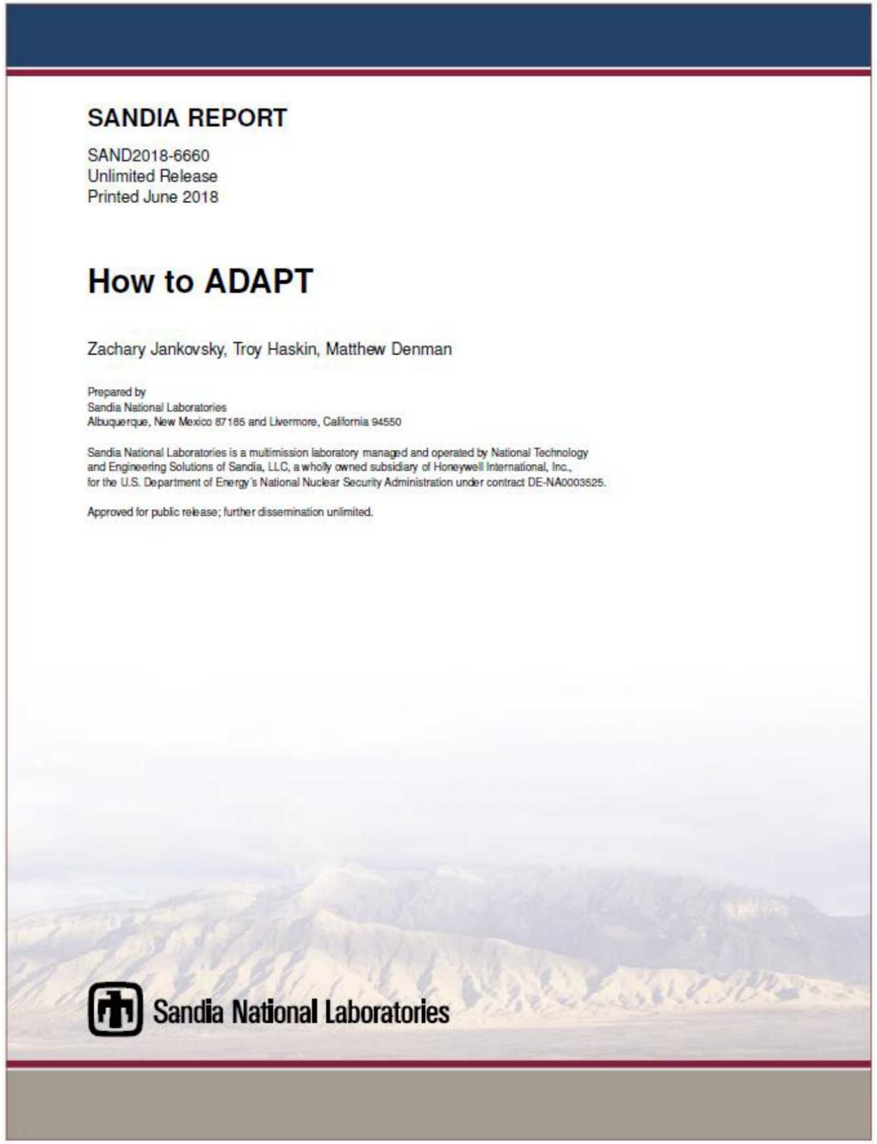
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Acknowledgments

- This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories under a partnership with The Ohio State University Nuclear Engineering Program.
- Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Outline

- Dynamic PRA
- ADAPT Overview
- Recent Analysis Tools
- Performance Improvements
- HPC Operation



SANDIA REPORT

SAND2018-6660
Unlimited Release
Printed June 2018

How to ADAPT

Zachary Jankovsky, Troy Haskin, Matthew Denman

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Approved for public release; further dissemination unlimited.

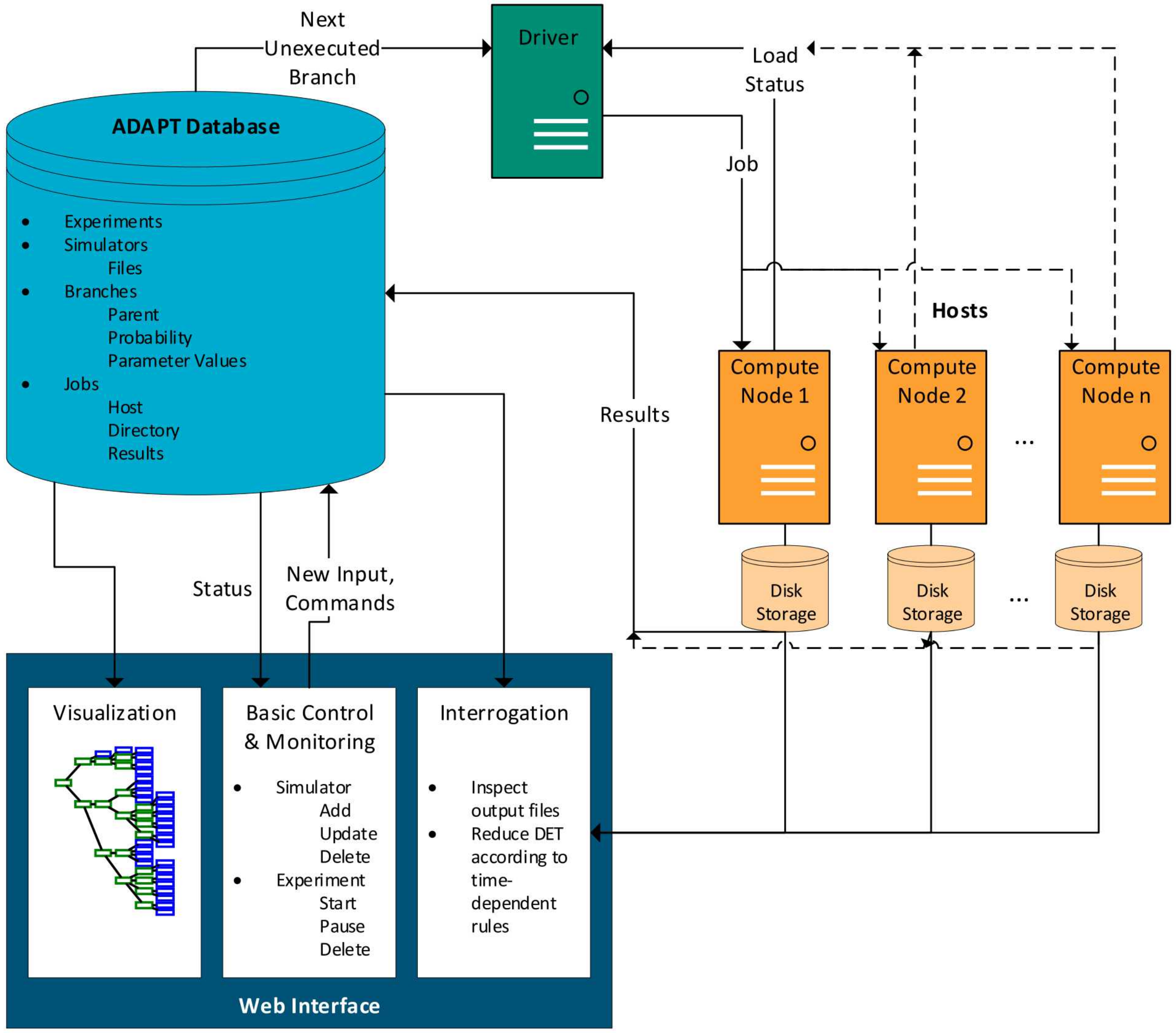
Sandia National Laboratories

Dynamic Probabilistic Risk Assessment (PRA)

- Traditional PRA requires analysts to assume order of events
  - Does not explicitly account for timing of events
    - Will an event have different effects on incident progression based on its timing?
  - Uncertainties in event ordering may be higher in certain problem space
    - E.g., Level 2 PRA for nuclear power plants
- Dynamic PRA is driven by time-resolving models of the relevant phenomena
  - Events occur according to physically-meaningful rules
    - E.g., hydrogen igniter success is queried only when a combustible mixture has accumulated
  - Events may re-occur as appropriate (e.g., valve failure query on cycling)
  - Dynamic event trees (DETs) are easily incorporated into a traditional PRA

ADAPT Approach

- DET driver developed for/by SNL (2006-present)
  - Tracks DET database, launches jobs, and presents results
- Simulator- and domain-agnostic
  - Simulators must meet a short list of requirements
    - Capable of restarting from saved state with new input
  - Simulator interactions performed via signal files rather than shared memory
    - Traceability
    - Portability over diverse computational hosts



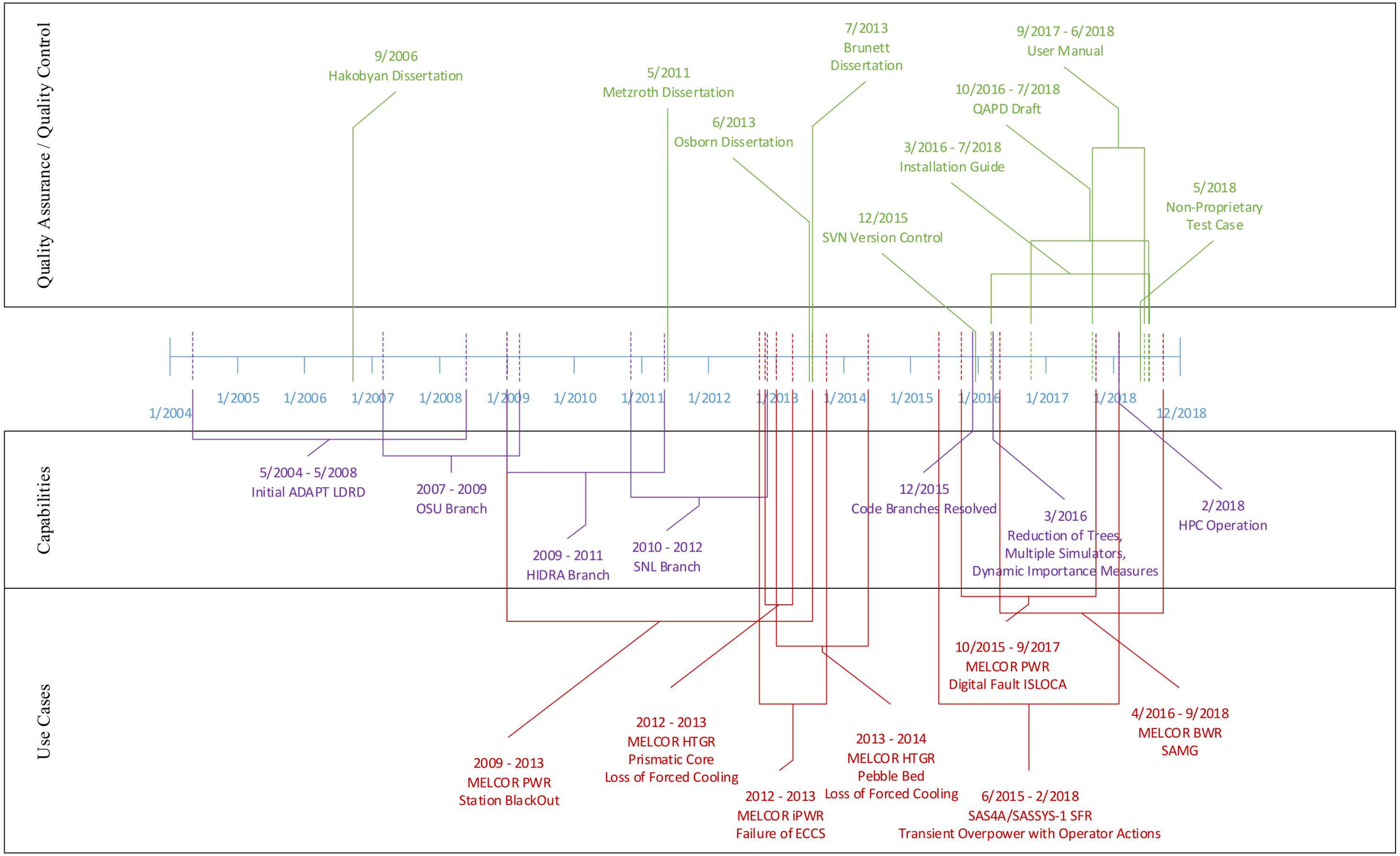
ADAPT Applications

Years	System	Incident	Simulator(s)
2006-2011	PWR	SBO	MELCOR
2009	SFR	Aircraft Crash	RELAP5
2013	PWR	SBO	MELCOR
2013-2014	PWR	SBO	MELCOR
2014	HTGR	LOFC	MELCOR
2015-2017	PWR	SBO	MAAP4
2015-2017	SFR	TOP	SAS4A/SASSYS-1
2015-2018	PWR	ISLOCA	MELCOR, RADTRAD
2015-2018	BWR	SBO	MELCOR
2016-2018	SNF Cask	Derailment	STAGE, RADTRAN

PWR: Pressurized Water Reactor
SFR: Sodium-cooled Fast Reactor
HTGR: High Temperature Gas-cooled Reactor
BWR: Boiling Water Reactor
SNF: Spent Nuclear Fuel

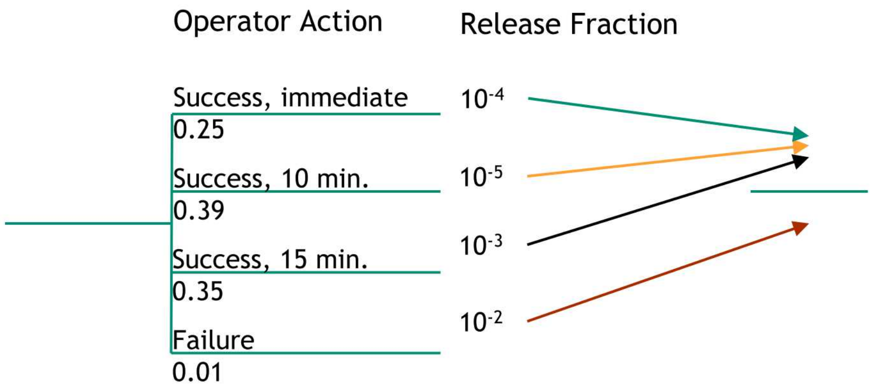
SBO: Station Blackout
LOFC: Loss of Forced Cooling
TOP: Transient Overpower
ISLOCA: Interfacing System Loss of Coolant Accident

ADAPT Timeline



Recent Analysis Tools (1/2)

- Dynamic Importance Measures (DYIs)
  - Compare expected values of chose consequences by branching condition value
    - Event occurrence vs non-occurrence, e.g.: $\frac{Release\ Fraction(operator\ action\ succeeds)}{Release\ Fraction(operator\ action\ fails)}$
    - Event extent vs non-occurrence, e.g.: $\frac{Release\ Fraction(operator\ action\ succeeds\ in\ 15\ minutes)}{Release\ Fraction(operator\ action\ fails)}$
    - Event extent vs all occurrence, e.g.: $\frac{Release\ Fraction(operator\ action\ succeeds\ in\ 15\ minutes)}{Release\ Fraction(operator\ action\ succeeds)}$
  - Mechanistically generate DYIs and rank to find impactful relationships

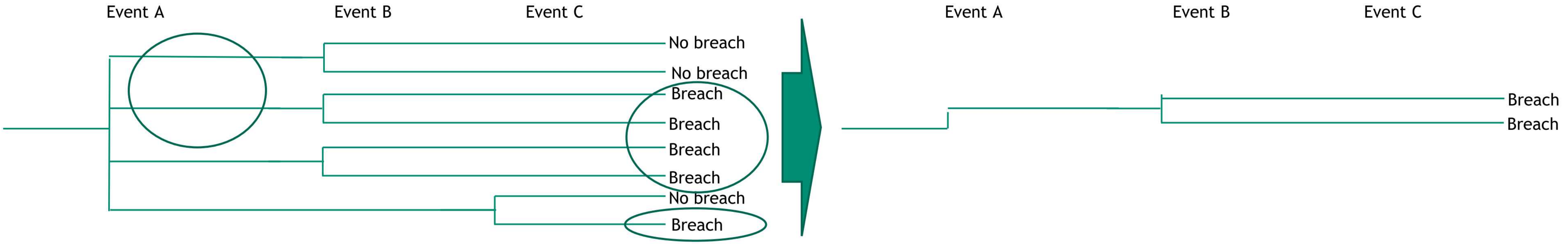


$$=\frac{Release\ Fraction(operator\ action\ succeeds)}{Release\ Fraction(operator\ action\ fails)}=\frac{\frac{0.25*10^{-4}+0.39*10^{-5}+0.35*10^{-3}}{0.25+0.39+0.35}}{\frac{0.01*10^{-2}}{0.01}}=0.04$$

- The expected value of the release fraction when the operator action succeeds is 0.04 times the expected value when the operator action fails.

Recent Analysis Tools (2/2)

- Multiple Simulator Analyses
  - Allows a DET to be driven by any number of simulators
  - Each branching condition transfers to pre-determined simulator
    - Processing steps must be defined for each allowed transition
      - E.g., MELCOR-MELCOR, MELCOR-MACCS, MACCS-MACCS, but not MACCS-MELCOR
- Reduction of DETs according to time-dependent rules
  - E.g., return only sequences where operator action succeeded in 11 minutes or less and vessel breached
  - All ADAPT analysis tools may be used on the reduced DET
    - Compare conditional insights to base DET



Performance Improvements (1/2)

- Inherited codebase
  - Designed around ~2006 hardware/software environments
    - Ample opportunity for high ROI improvements
- File operations are costly
  - Results distributed across multiple machines/filesystems
  - Parallelize gathering of results
    - Scales to 98% of $1/n_{cores}$ time required to gather a single variable for all DET branches
    - **Next step: establish ADAPT post-processing scheme to distribute work to additional nodes**
  - Cache results
    - When results are demanded, check if files have changed in any branch of the DET
      - If no change, use a cached copy of results
        - 4x wall time reduction for finished DET
      - If files have changed, pull fresh data
    - **Next step: check branches individually**
      - Further reduction in un-necessary duplication when some branches have changed

Performance Improvements (2/2)

- Database operations are costly
  - Significant overhead in each query

    8,300 queries with one result each take 1,400 times the wall time of a single query with 8,300 results
  - Reduce number of queries
    - Remove database queries from for loops
    - Pull all relevant data in a single query and loop over results in memory
  - Example: pulling relationships of all branches in a DET
    - Previously performed iteratively
      - Database query for each relationship
    - Now entire branches table is pulled in one query
      - Relationships calculated locally
      - Saves 60% wall time
        - Used in many post-processing routines

HPC Operation - Motivation

- Historical use of ADAPT
  - Desktop computer: 40 cores, 10 TB storage
    - Full control over scheduling
  - Local cluster: 200 cores, 200 TB storage
    - High control over scheduling
- Combinatorial explosion
  - Each additional branching condition may significantly expand DET
  - Branch input may require simulator to run for minutes to weeks
  - Easy to generate a DET that is computationally impractical to finish
    - And can be difficult to predict the eventual size of a DET
- Opportunity (Sandia example)
  - Available corporate clusters: 100,000 cores, 10PB storage
    - Little control over scheduling

HPC Operation – A Note on Terminology

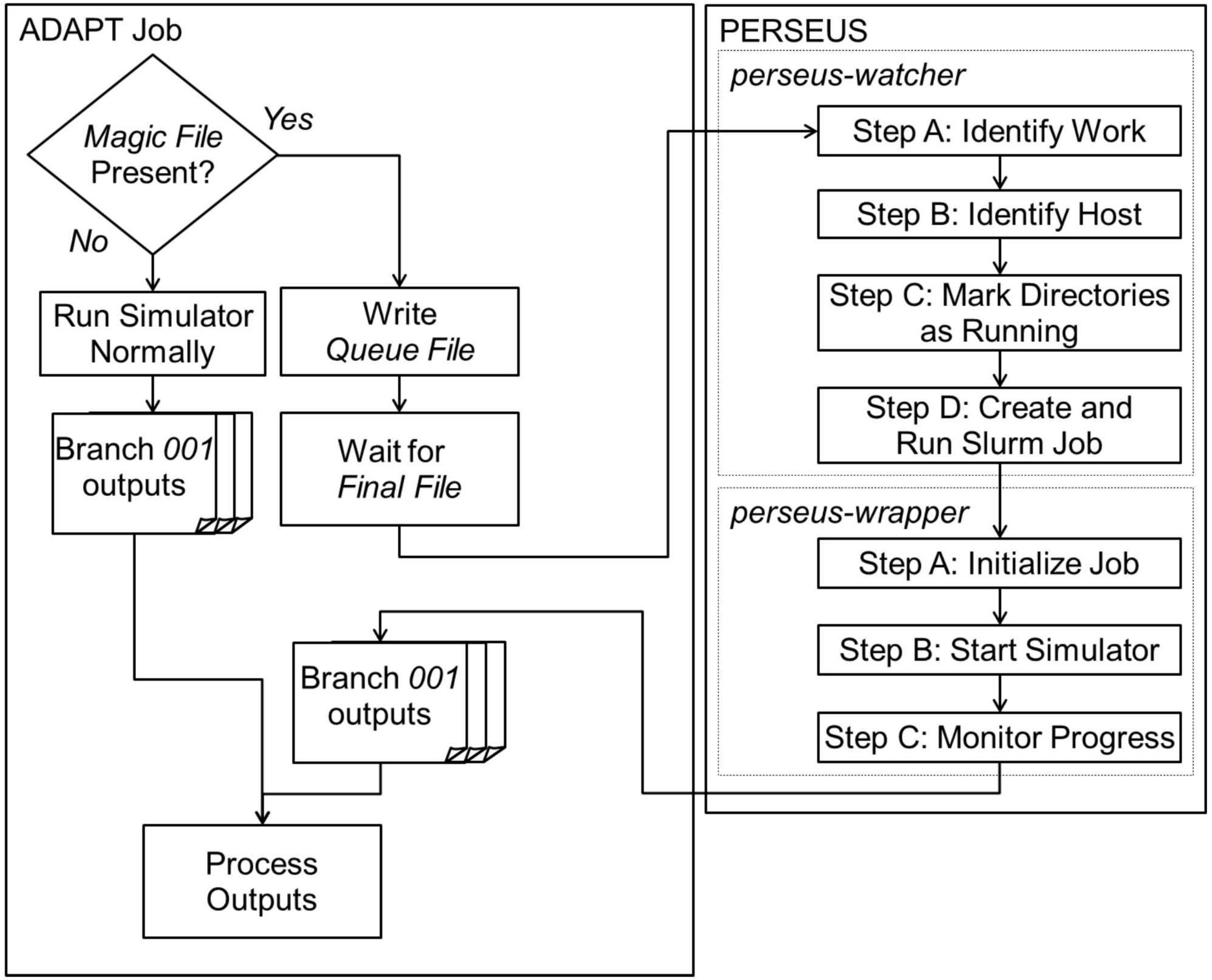
- ADAPT branch:
  - A segment of the analysis with a set of uncertain system parameters that remain constant until a branching condition is reached
- ADAPT job:
  - An attempt to run the input associated with a branch on a particular computational host
- HPC job:
  - A script that is run on a particular computational host until it completes or meets a time limit
    - May include multiple ADAPT jobs

HPC Operation - Constraints

- ADAPT job scheduling
  - Historically has used ssh/scp commands to communicate with computational hosts
    - No special software required on computational hosts
  - Resources allocated a core at a time
  - ADAPT jobs may run until finished with no time limit
- HPC job schedulers have strict requirements
  - Scheduler-specific submission tool
  - Resources typically allocated a node at a time
  - Limited run time
- ADAPT jobs are independent
  - HPC capacity vs capability
- ADAPT jobs are unpredictable in time requirement
- Simulators typically used with ADAPT are single threaded
  - Node-based submission not advantageous

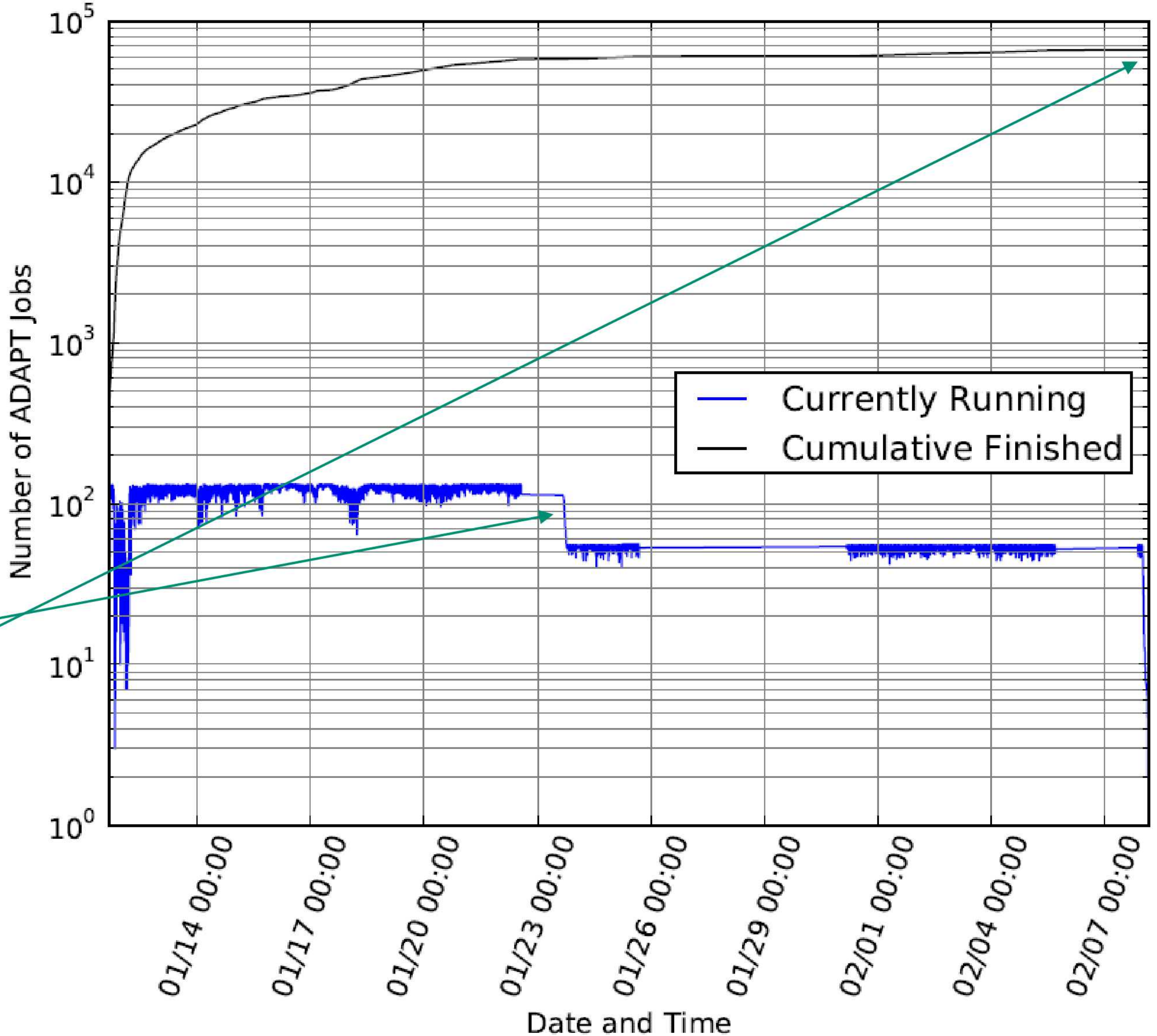
HPC Operation – Proof of Concept Approach

- Intercept running ADAPT jobs
  - Run normally on local cluster until simulator execution
  - Bundle enough ADAPT jobs to fill an HPC node and submit an HPC job
  - At end of HPC job time limit:
    - If an ADAPT job has finished, signal that HPC work is done
    - If an ADAPT job has not finished, return it to the local cluster for another round on the HPC
  - ADAPT job closeout process does not change
- Production implementation will integrate HPC as an ADAPT computational host type



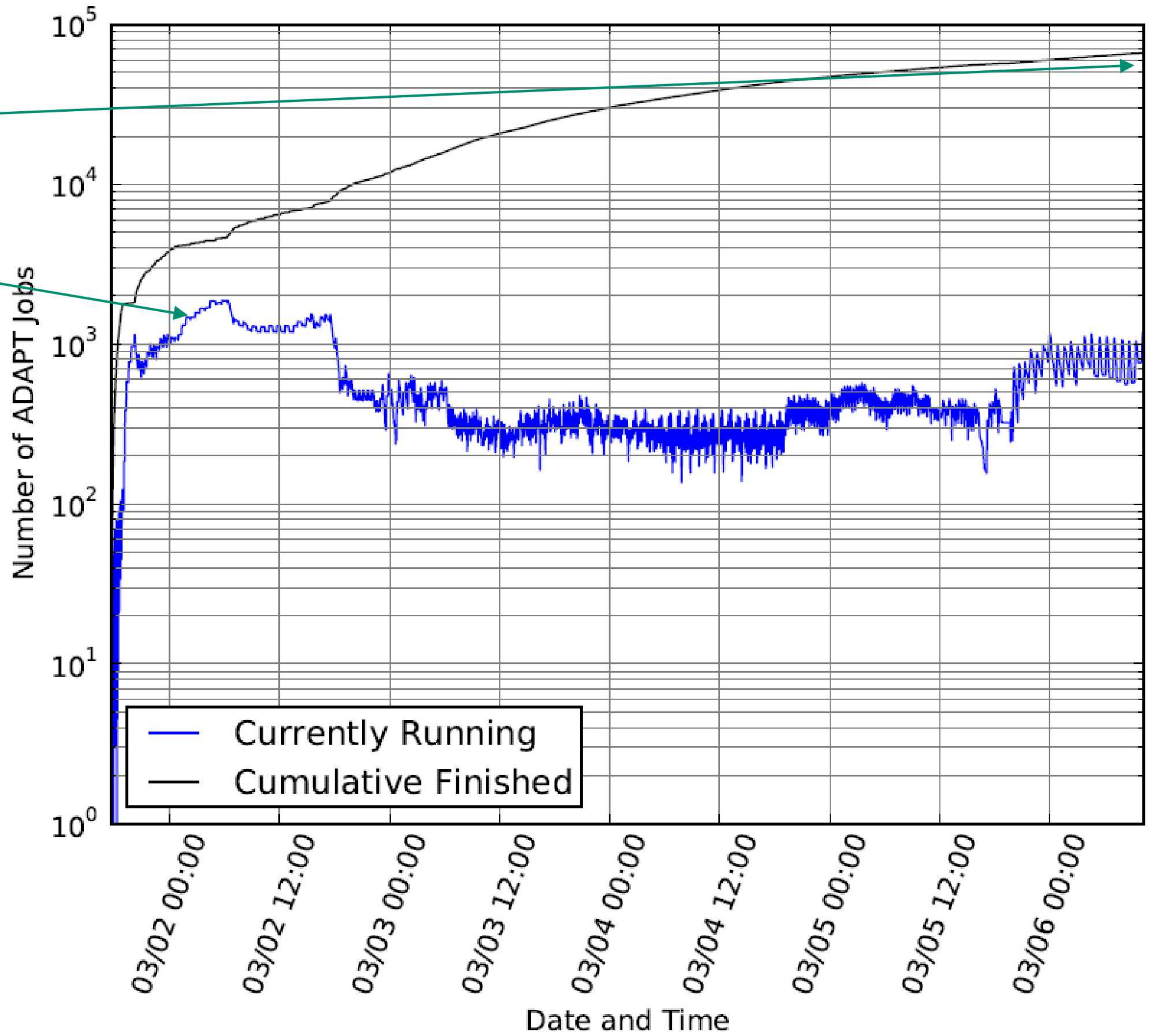
HPC Operation – Test Case on Local Cluster

- Pressurized water reactor interfacing system loss of coolant accident
  - MELCOR severe accident simulator and RADTRAD dose calculation simulator
    - Only MELCOR branches sent to HPCs
  - Uncertain capacity of systems for overpressurization
  - Uncertain success and timing of operator mitigating actions
- Test case run first on local cluster
  - Maximum 132 cores
  - Required to share capacity with another ADAPT case (down to 55 cores)
  - 66,076 branches completed in 27.5 days



HPC Operation – Test Case on HPCs

- Test case run next on HPCs
  - Same progress as small cluster run (66,076 branches completed) in 4.7 days
    - 6x reduction in wall time required for same progress
  - Significant variation in open ADAPT jobs over time
    - Varies with HPC load
- Potential for savings increases with number of queued branches



HPC Operation - Feedback

- Common HPC work packages request multiple nodes and run to completion with little interaction
  - E.g., computational fluid dynamics or finite element analysis problems
- ADAPT on HPCs presents an atypical workload
  - ADAPT frequently polls HPCs for load status to identify HPCs with idle nodes
    - Because single nodes are requested at a time, queueing may be avoided
    - Will be made moot if HPC federation is implemented
  - If all ADAPT jobs in an HPC job finish early, the HPC job finishes early
- HPC administrators took notice
  - Frequent ssh connections to HPC head nodes to check status
  - Significant numbers of HPC jobs not running to requested time
  - Frequent and significant traffic to and from a remote system on the network
  - Coordinated with administrators to identify and test process improvements

Summary

- DPRA can give additional insight to complex event progressions
  - What physical parameters are impactful?
  - How does the timing of human interaction affect the outcome?
- ADAPT is a flexible DET generation and analysis platform
  - Limited only by availability of appropriate simulators
  - Easily adaptable to various computational environments
  - Extensible data analysis tools
    - Scalable from hundreds to 1M+ branches