

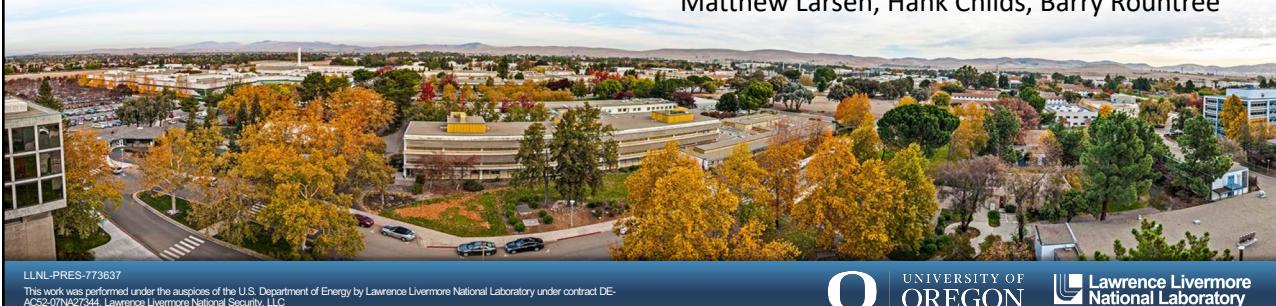
# Power and Performance Tradeoffs for Visualization Algorithms

IPDPS'19, Rio de Janeiro, Brazil

21 May 2019



Stephanie Labasan, LLNL, Univ. of Oregon  
Matthew Larsen, Hank Childs, Barry Rountree



LLNL-PRES-773637

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC.



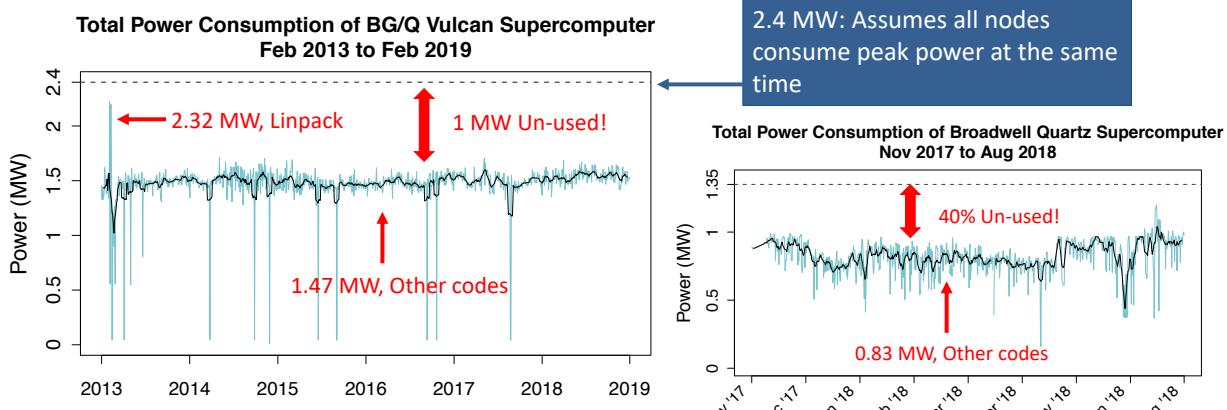
## Our study is at the intersection of overprovisioning and visualization

- **Problem:** Power is a critical constraint for next generation supercomputers
- **Solution:** Hardware overprovisioning (...and also hardware improvements)
  - “More nodes,” but enforce power limits, degrading performance
- **This study:**
  - Premise: Visualization algorithms under power limits will differ from HPC applications
    - Visualization is data intensive
  - What we did: Consider a representative set of visualization algorithms and investigate tradeoffs between slowdown and power usage
  - Outcome: Understand execution characteristics under power limits
  - Why this is important: Exploit characteristics for better performance

## Outline

- Motivation
  - Power/Overprovisioning
  - Why Visualization?
  - Overprovisioning and Visualization
- Experimental Overview
- Results
- Conclusion and Future Work

## Current system power utilization is sub-optimal



- Limits the size of the system, reducing system throughput
- Power capacity left on the table, more science could be done

## Total system power usage and costs are rising

- Power becoming an increasingly scarce resource
  - Expensive cost
  - Limitations enforced by supercomputer facilities
- Target is 20-40 MW system power usage for 1 Exaflop
  - Need to maximize limited power resources
- **Key Premise:** To meet power consumption goals at exascale, need to innovate power-efficient techniques not only in hardware, but also in software

## Improve utilization of power resources with *overprovisioning*

### Traditional

- All nodes can run at peak power simultaneously

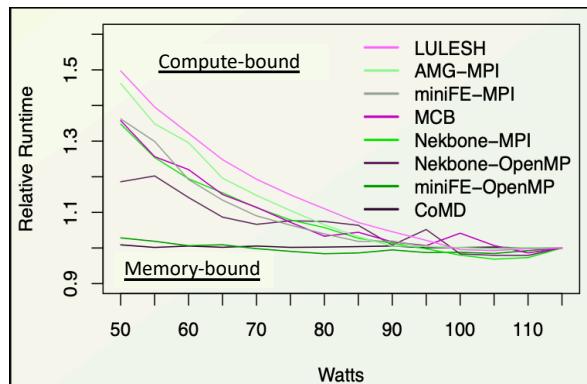
### Overprovisioning

- Increase compute capacity by adding more nodes
  - Not all nodes can run at max power simultaneously
- Limit power usage per node
  - Example: 2X nodes, each running at 1/2 power → benefits common use case

For overprovisioning to be effective, all aspects of the HPC ecosystem (including visualization!) must be reconsidered with power in mind.

## Limit power usage with hard power limits

- **Outcome:** Takes longer to run, but uses less power
- **Why:** Non-linear relationship between power and CPU frequency
  - Lower CPU frequency results in less power usage
  - But, subcomponents still consume power at the same rate
- Application behavior under power limits varies



## In a power-constrained environment, performance of visualization routines will be affected

- Central manager coordinating power allocations across system
  - Given a power constraint, adjust application accordingly
- We are moving towards a workflow where the visualization and analysis are concurrently running with the simulation (*i.e.*, *in situ*)
- **This study:** Visualization component under a power limit

## Outline

- Motivation
  - Power/Overprovisioning
  - Why Visualization?
  - Overprovisioning and Visualization
- Experimental Overview
- Results
- Conclusion and Future Work

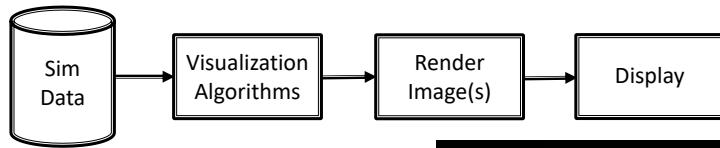
## Why visualization?

- #1: Visualization is important for scientific insight
- #2: Visualization can use significant resources on the HPC system (10-20% of the total runtime)
- #3: With overprovisioning, visualization workloads respond differently than simulation workloads
  - discussed in overprovisioning/visualization section)

## #1: Visualization is important for understanding scientific simulation data

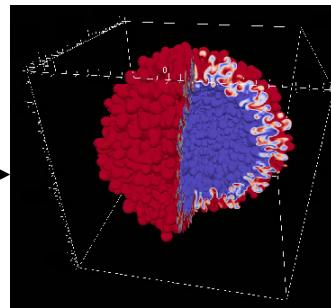
- Central actor in the scientific discovery process:

- Three use cases:
  - Communicate
  - Validate
  - Explore



```

2.14484 2.13043 2.11357 2.09448 2.07345 2.12172 2.13936 2.15521 2.16894
2.18022 2.1888 2.19483 2.19966 2.20665 2.21965 2.23869 2.25985 2.2795
2.29598 2.30887 2.31819 2.32493 2.32651 2.32577 2.322 2.31549 2.30665
2.29597 2.28405 2.27154 2.25906 2.24724 2.23658 2.22748 2.22021 2.21485
2.2114 2.20969 2.20948 2.21048 2.21231 2.21461 2.21696 2.21895 2.22014
2.22008 2.2183 2.21437 2.2079 2.1986 2.18632 2.17108 2.15304 2.13249
2.10979 2.15935 2.17844 2.1958 2.21118 2.22435 2.23523 2.24414 2.2523
2.26215 2.27613 2.29422 2.31388 2.33243 2.34835 2.36102 2.37024 2.37593
  
```



## #2: Visualization will share resources with the simulation

- Traditional approach is *post hoc* processing
  - Not feasible at exascale: Compute – and thus ability to generate data – is increasing faster than I/O
- Data analysis and visualization occur *while* simulation is running
- Changes to HPC ecosystem pose challenges for visualization
  - Total execution time may increase since compute resources are shared
  - Optimizing visualization will benefit overall performance

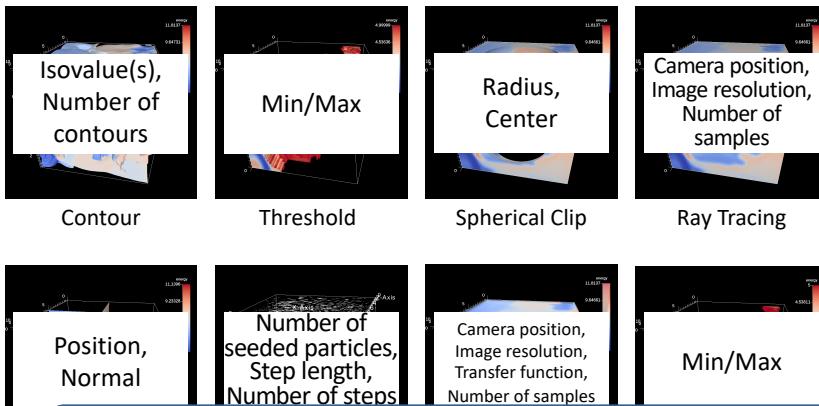
## Outline

- Motivation
  - Power/Overprovisioning
  - Why Visualization?
  - Overprovisioning and Visualization
- Experimental Overview
- Results
- Conclusion and Future Work

## Overprovisioning and visualization merits special attention

- **Why?** Visualization is data intensive
  - → Need visualization-centric strategies to thrive in this constrained environment

## Hundreds of visualization algorithms, not practical to optimize each individually



- Execution behaviors differ across algorithms
- Unique configurations for each algorithm
- Pipeline algorithms

Our study is foundational for understanding the execution behaviors for a *representative* set of all algorithms under power limits.

200<sup>th</sup> cycle

## Overprovisioning and visualization merits special attention

- Why?** Visualization is data intensive
  - Need visualization-centric strategies to thrive in this constrained environment

### Research Questions

- How do visualization workloads perform under power limits?
- If you must run at a lower power limit, how much slowdown will you see?

## Outline

- Motivation
  - Power/Overprovisioning
  - Why Visualization?
  - Overprovisioning and Visualization
- Experimental Overview
- Results
- Conclusion and Future Work

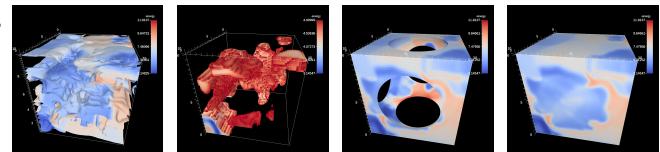
## Study Factors

- Processor-level power limit
  - Visualization algorithm
  - Data set size
- Single node of LLNL RZTopaz system (Intel Broadwell)
    - 36 cores per node running at 2.1 GHz
  - Leverage `msr-safe` kernel driver to set power limit from user space
  - Enforce power limits with Intel's RAPL
    - Acceptable limits between 120W down to 40W

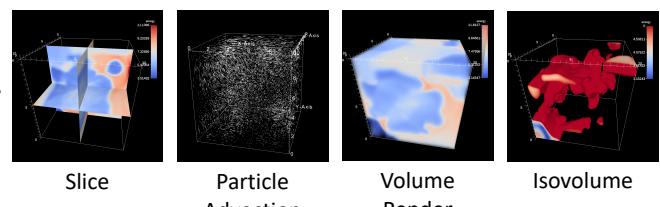
<https://github.com/llnl/msr-safe>

## Study Factors

- Processor-level power limit



- Visualization algorithm



- Data set size

Lawrence Livermore National Laboratory  
LLNL-PRES-773637

UNIVERSITY OF  
OREGON

CASC

NNSA  
National Nuclear Security Administration  
19

## Study Factors

- Processor-level power limit

- CloverLeaf3D: Hydrodynamics proxy application

- Visualization algorithm

- Data set sizes per node
  - $32^3, 64^3, 128^3, 256^3$

- Data set size

- Number of cells per node range from 32K to 16M

Lawrence Livermore National Laboratory  
LLNL-PRES-773637

UNIVERSITY OF  
OREGON

CASC

NNSA  
National Nuclear Security Administration  
20

## Methodology

- Study conducted in 3 phases
- Phase 1: Vary power limit for a single visualization algorithm and data set size
- Following phases varies 1 factor at a time to investigate effects
  - Phase 2: Vary visualization algorithm
  - Phase 3: Vary data set size

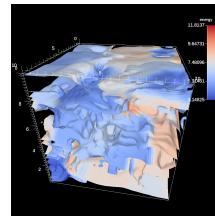
## Outline

- Motivation
  - Power/Overprovisioning
  - Why Visualization?
  - Overprovisioning and Visualization
- Experimental Overview
- Results
- Conclusion and Future Work

## Phase 1 Results: Vary Processor-Level Power Limit

- Reducing power limit by 3X:
  - Runtime increases by 1.2X
  - Frequency decreases by 1.2X
- Contour algorithm does not see a slow down proportional to power limit

P	$P_{ratio}$	Contour			
		T	$T_{ratio}$	F	$F_{ratio}$
120W	1.0X	33.477s	1.00X	2.55GHz	1.00X
110W	1.1X	33.543s	1.00X	2.41GHz	1.06X
100W	1.2X	33.579s	1.00X	2.55GHz	1.00X
90W	1.3X	33.519s	1.00X	2.55GHz	1.00X
80W	1.5X	33.617s	1.00X	2.54GHz	1.01X
70W	1.7X	30.371s	0.91X	2.54GHz	1.00X
60W	2.0X	30.394s	0.91X	2.50GHz	1.02X
50W	2.4X	31.066s	0.93X	2.52GHz	1.01X
40W	3.0X	39.198s	1.17X	2.07GHz	1.23X



- Favorable power savings

Lawrence Livermore National Laboratory  
LLNL-PRES-773637

UNIVERSITY OF  
OREGON

CASC

NNSA  
National Nuclear Security Administration  
23

## Changes for Phase 2

- Changes: Execute all visualization algorithms under varying power limits
- Goal: Investigate the data intensity of additional visualization algorithms

Lawrence Livermore National Laboratory  
LLNL-PRES-773637

UNIVERSITY OF  
OREGON

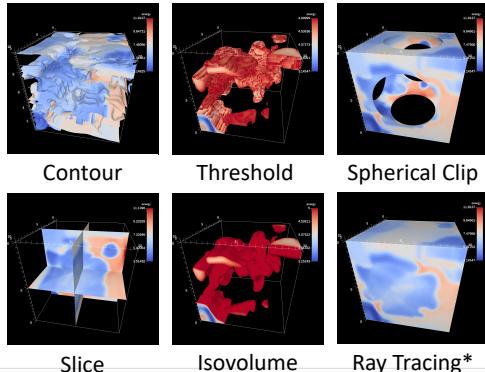
CASC

NNSA  
National Nuclear Security Administration  
24

## Phase 2 Results: Vary Visualization Algorithm

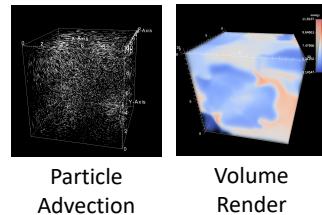
### Power Opportunity ✓

- Performance remains constant until lowest power cap



### Power Sensitive ✗

- Performance degrades relative to power cap
- Compute-bound workloads



\*Ray tracing includes 3 operations: (1) find external faces (50%), (2) build spatial acceleration structure, and (3) trace rays

Lawrence Livermore National Laboratory  
LLNL-PRES-773637

UNIVERSITY OF  
OREGON

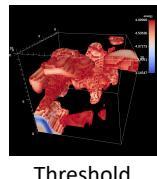
CASC

NNSA  
National Nuclear Security Administration  
25

## (cont'd) Phase 2 Results: Comparing power opportunity and power sensitive algorithms

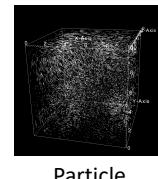
### Power Opportunity: Threshold ✓

- Iterate over the cells in data set
- Remove cells that do not have values in the specified range
- Simple computation dominated by loads and stores of each cell



### Power Sensitive: Particle Advection ✗

- Place a massless particle at a seed location
- Displace the particle according to the vector field
- Result is an “integral curve” corresponding to the trajectory the particle travels – Euler, RK4, ...



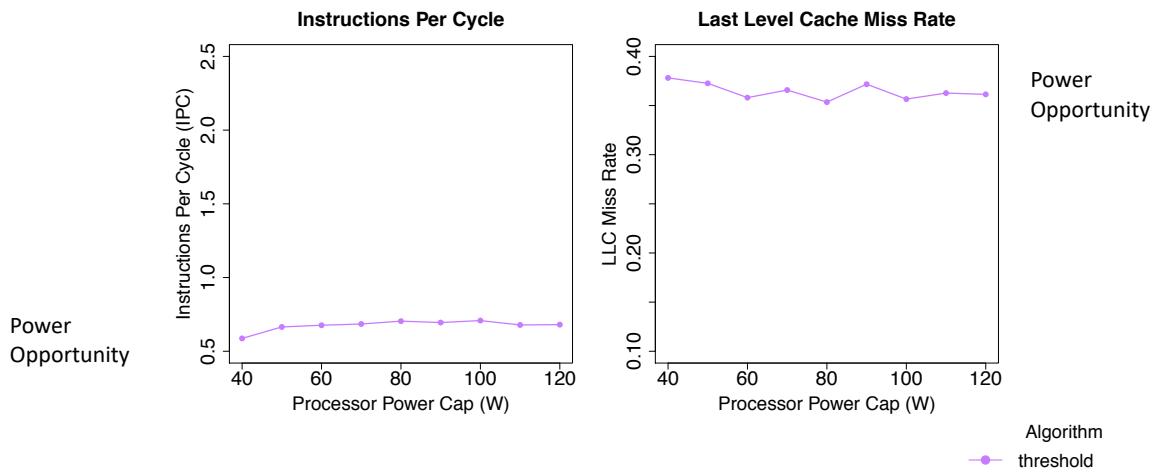
Lawrence Livermore National Laboratory  
LLNL-PRES-773637

UNIVERSITY OF  
OREGON

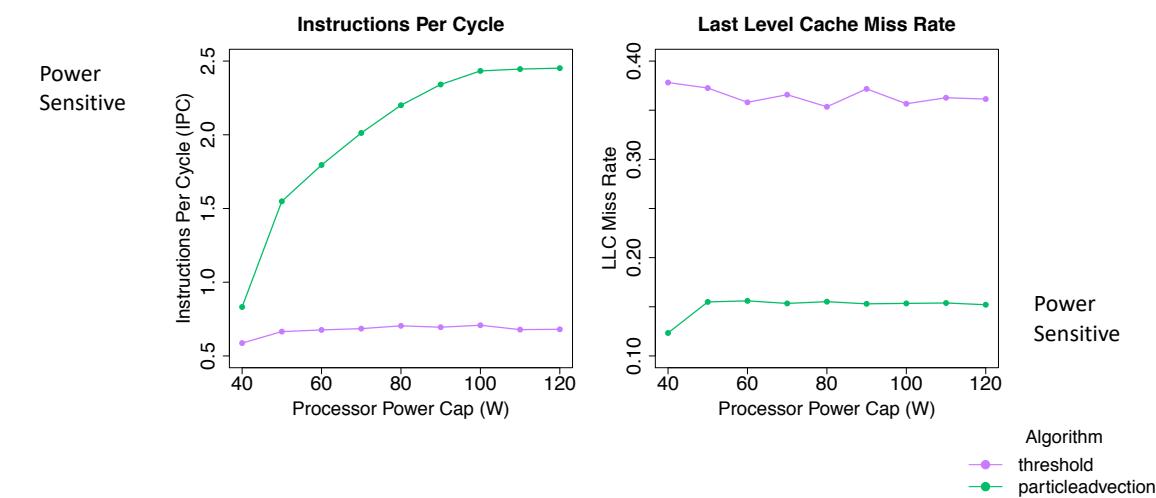
CASC

NNSA  
National Nuclear Security Administration  
26

**(cont'd) Phase 2 Results: Power opportunity algorithms are data intensive, while power sensitive algorithms are compute intensive**



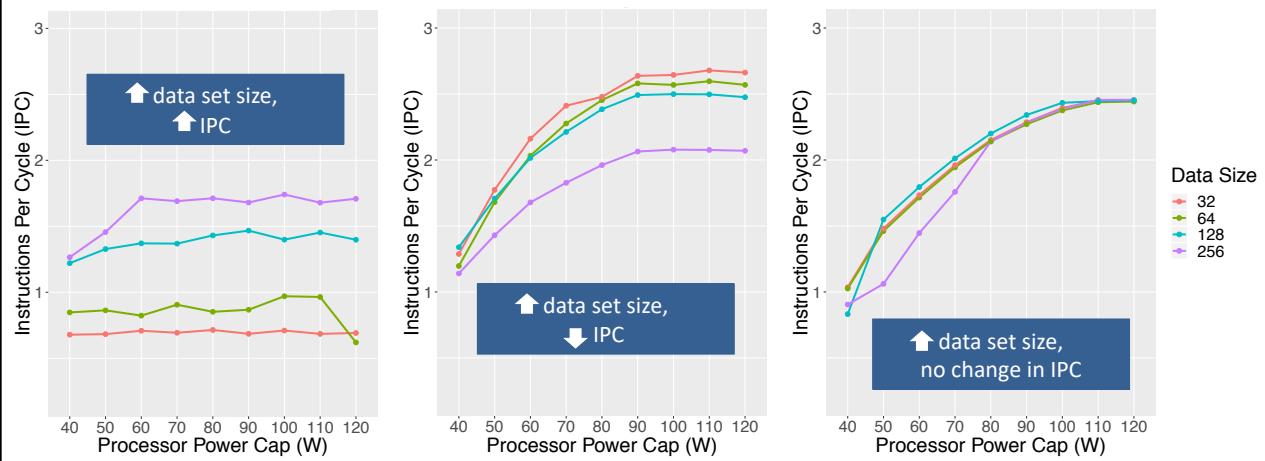
**(cont'd) Phase 2 Results: Power opportunity algorithms are data intensive, while power sensitive algorithms are compute intensive**



## Changes for Phase 3

- Changes: Execute all visualization algorithms with different data set sizes
- Goal: Investigate the impacts of data set size on power and performance tradeoffs

## Phase 3 Results: Vary Data Set Size



## Outline

- Motivation
  - Power/Overprovisioning
  - Why Visualization?
  - Overprovisioning and Visualization
- Experimental Overview
- Results
- Conclusion and Future Work

## Takeaways inform how visualization algorithms can adapt to imposed power limits

- VTK-m contour implementation is sufficiently data intensive to avoid a significant slowdown when the power limit is reduced
- Most algorithms consume low amounts of power, providing opportunities for energy/power savings (*e.g.*, data intensive)
- Tradeoffs become more favorable with a smaller data set size per node
- **Future work:** Integrate these learnings into a power-aware runtime system, such that more informed decisions can be made about how to allocate power in an in situ workflow

## Thank you!

- VTK-m contour implementation is sufficiently data intensive to avoid a significant slowdown when the power limit is reduced
- Most algorithms consume low amounts of power, providing opportunities for energy/power savings (e.g., data intensive)
- Tradeoffs become more favorable with a smaller data set size per node
  
- **Future work:** Integrate these learnings into a power-aware runtime system, such that more informed decisions can be made about how to allocate power in an in situ workflow

“Power and Performance Tradeoffs for Visualization Algorithms”  
Stephanie (Labasan) Brink (brink2@llnl.gov)



33



**Disclaimer**  
 This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.