# Multiphysics Simulation at Exascale

Bert Still
AX Division, Weapons and Complex Integration

SOS16, Santa Barbara CA, March 13-15 2012

LLNL-PRES-526974

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

# Exascale computing presents unique challenges to multi-physics integrated codes

## Improved Physics

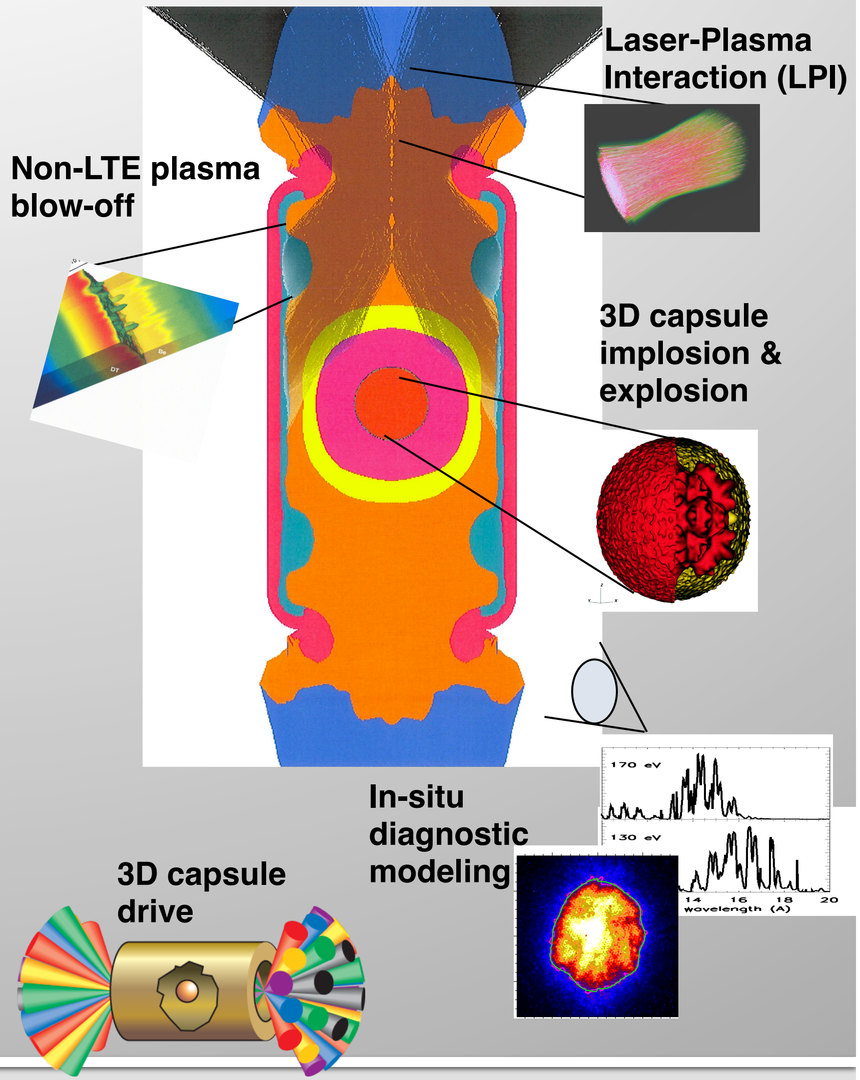
- Laser beam effects
- Plasma blow-off and effect on drive, symmetry
- Capsule implosion details
- Explosion symmetry
- Atomic physics
- Line radiation transport

**+**

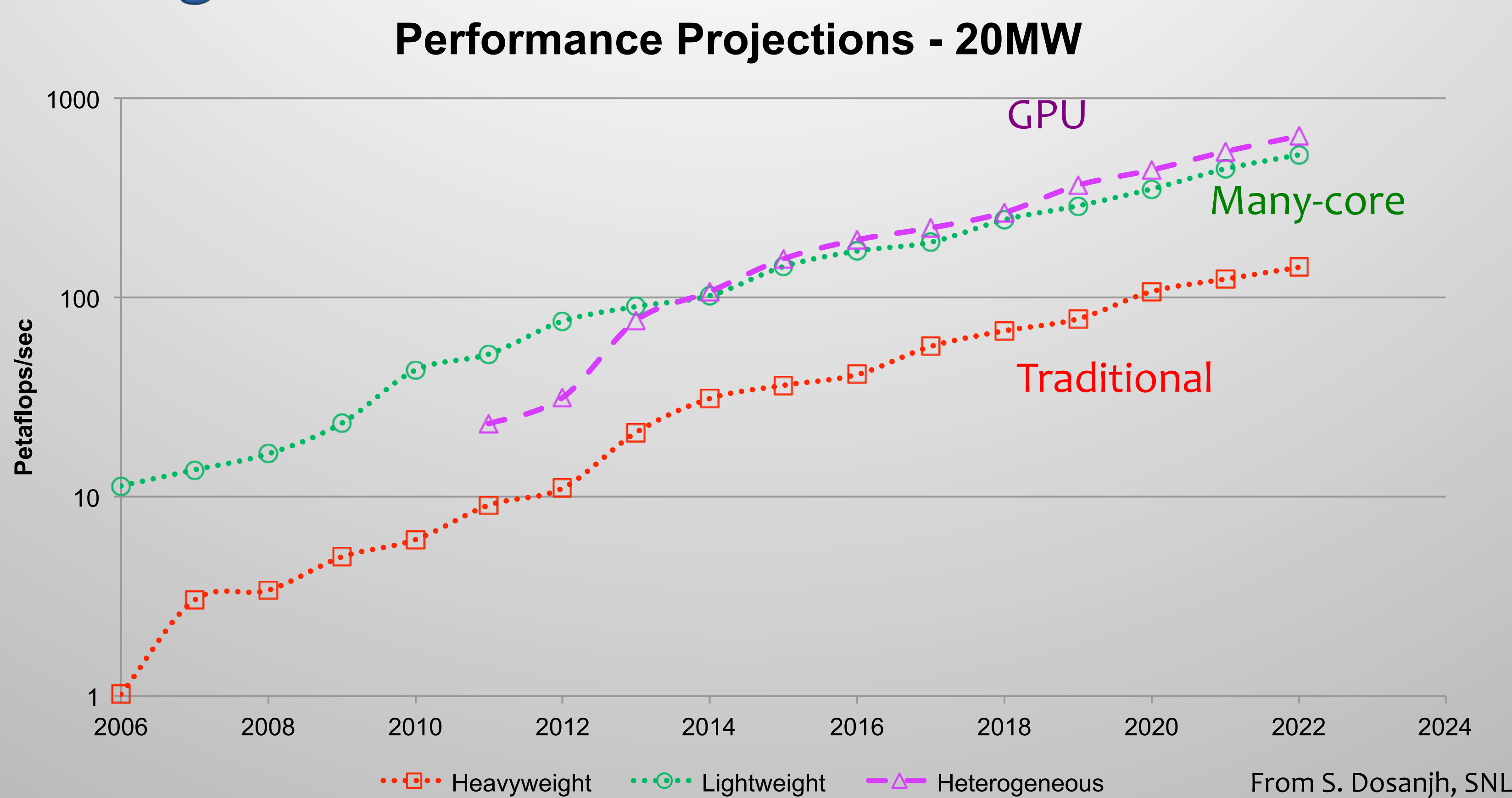
## Improved Resolution (multi-scale, time/space)

## Improved Understanding
(predictive capability)

## HEDP Example



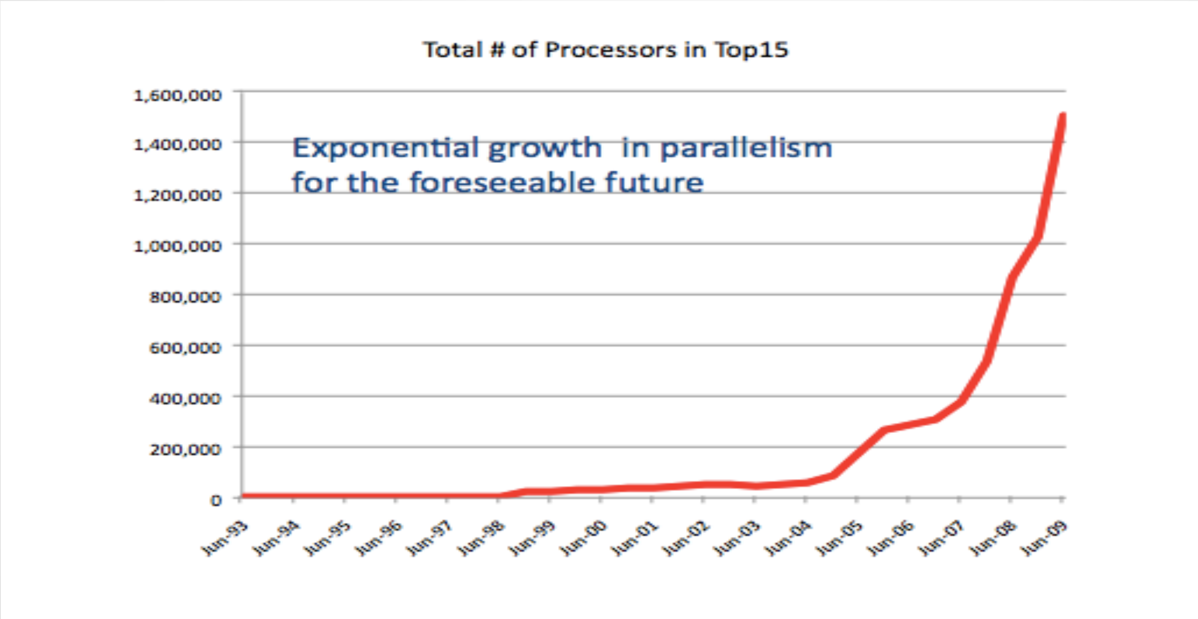
# Power constraints present a significant challenge to reaching exascale




From S. Dosanjh, SNL

**To meet power constraints, the architecture must change**

# Parallelism is exploding

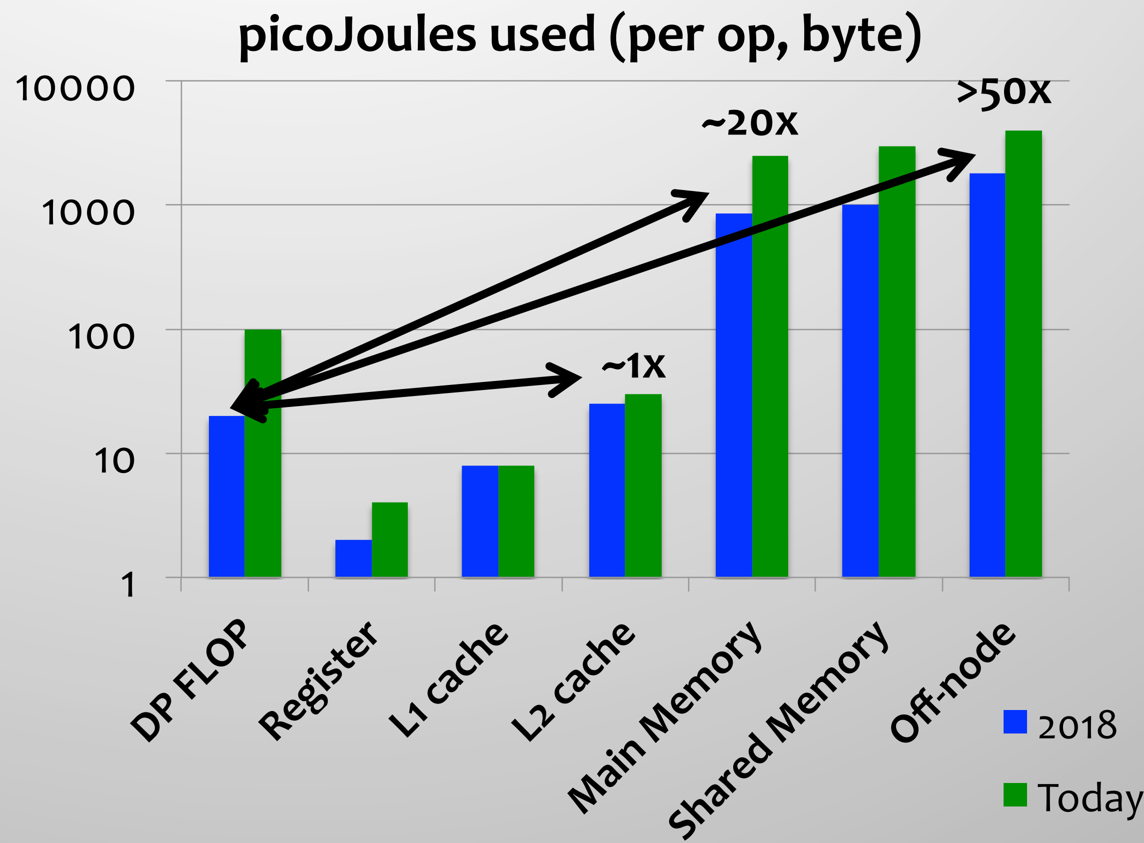



- **Many levels of parallelism are emerging (node, core, FPU)**
- **Heterogeneity is becoming common (GPU, accelerators, etc)**
- **Billion-way parallelism is expected at Exascale**
- **Weak scaling alone will not be sufficient to exploit additional parallelism**

**New programming models are needed**

# Computing is increasingly constrained by memory

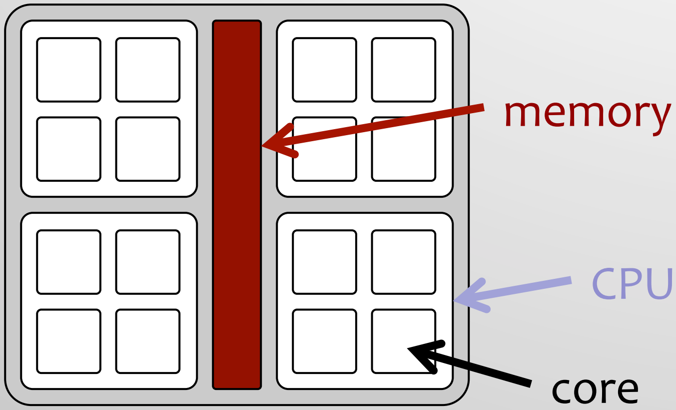
- **Core density** ***increasing dramatically***
- **Memory per node** ***increasing some***
- **Memory size and bandwidth per core** ***decreasing dramatically***
- **Data movement is the dominant power consumer**



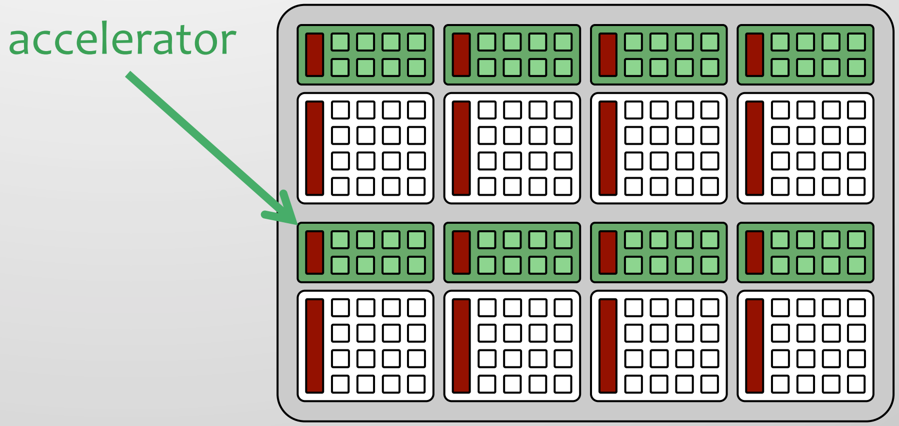

**Performance depends on managing data motion**

# Efficiently using exascale node hardware will provide a significant challenge

**Today's Homogeneous Node**




**Future Heterogeneous Node**

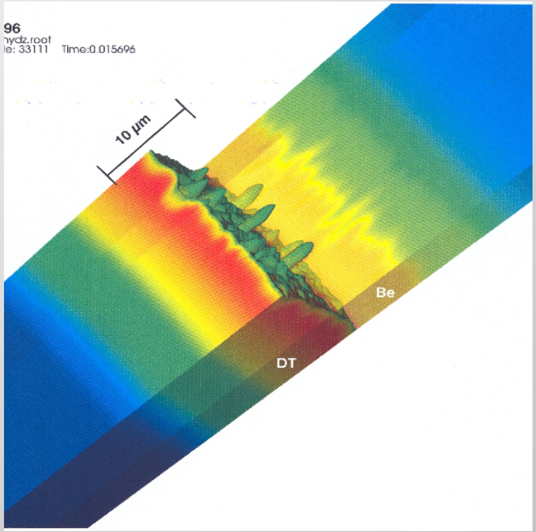



- Memory distributed across a node
- Nonlocal memory access introduces additional latency
- Performance depends on minimizing impact of that latency
- Locality matters and future programming models must express this

**Today: parallelize with MPI**
**Future: MPI + Memory Model + Threading**

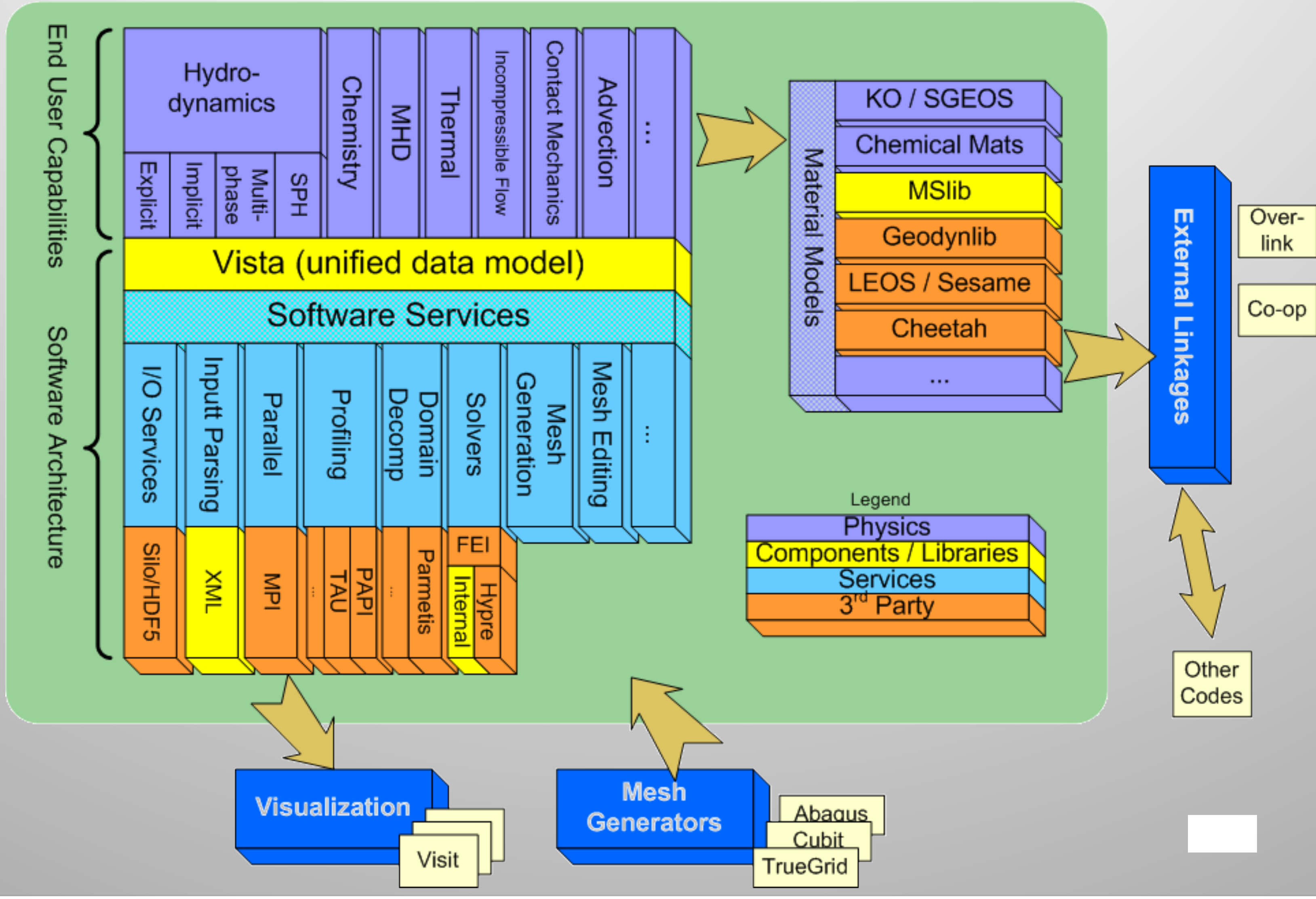
# Integrated Codes provide the greatest exascale challenge

- Often > 10 physics packages
- Many different spatial, temporal scales
- Algorithms tuned for minimal turn-around time instead of maximal computational efficiency
- Multi-language (Fortran, Fortran90, C, C++, Python)
- Variety of parallelism approaches
- Diverse memory and processor performance needs
- Steerable / interactive interfaces
- 30+ third party libraries
- Broad computational application space
- Long life-time projects with >1 million lines of code
- Restart/checkpoint up to 2Tb (today) – 1000x in future



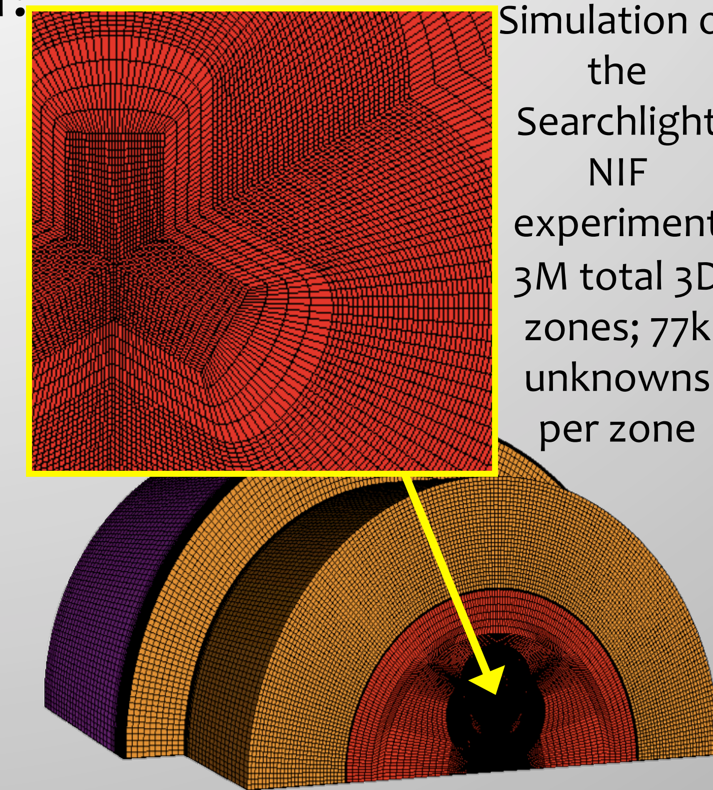

Complex hydrodynamics of an ICF capsule

# Example: ALE3D



# Substantial changes are required for multiphysics code performance at exascale

- Multiphysics codes are currently optimized for:
  - Large memory, large bandwidth
  - Manage flops
  - Store/fetch data better than re-compute
- Exascale architectures:
  - Memory/core, bandwidth limited (small)
  - Manage data
  - Re-compute better than store/fetch data?
- Biggest challenges
  - Millions of lines of trusted code
  - 10+ Physics packages
  - 30+ libraries

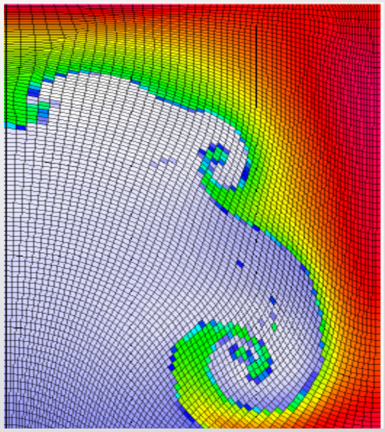


Simulation of the Searchlight NIF experiment 3M total 3D zones; 77k unknowns per zone

**Algorithm methodologies are needed to enable multiphysics codes to efficiently use exascale hardware**

# Hydrodynamics

- **LLNL relies principally on Arbitrary Lagrange-Eulerian (ALE) techniques**
- **Low order schemes typically used**
- **Exascale challenges:**
  - **Relatively good data locality, but flops per memory access is low**
    - — ***Would higher order schemes be more advantageous on future platforms?***
  - **Can have large spatial and temporal load imbalance**
    - — **E.g. AMR on unstructured meshes**



| Typical Characteristics | |
|---|---|
| Memory needs | 0.1 - 1 KB/zone |
| Access pattern | Regular with modest spatial and temporal locality |
| Communication pattern | **Point to point, surface communication** |
| Mflops per zone per cycle | 0.02 – 0.1 (can be 10X larger for iterative schemes) |
| EOS I/O (startup) | **20-160 MB currently sent to every process** |

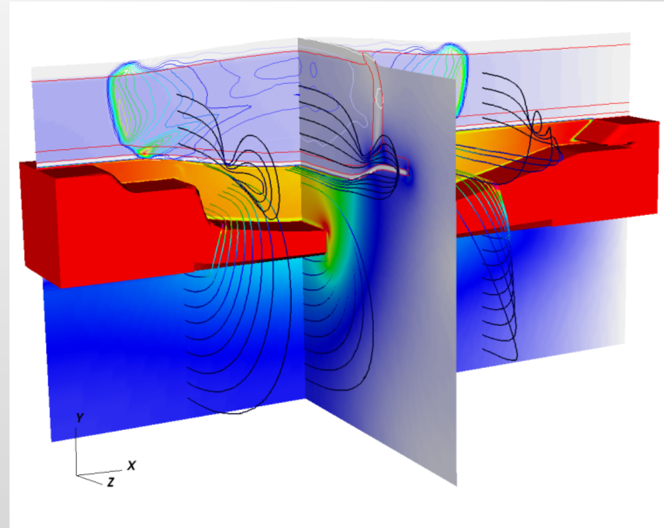
# Deterministic Transport

- **Exascale challenges (continued):**
  - **Often run on a mesh determined by hydro considerations**
    - ***Can the physics (radiation, neutrons, electrons, charged particles) be done on a separate mesh and accurately coupled to the hydro?***
      - ***If so, can higher order flux representations be used to reduce zone count while maintaining high flop rates?***
  - **Gather-summing of volumetric information will be costly**
    - ***How can data be arranged to minimize this?***

| **Typical Characteristics** | |
|---|---|
| Memory needs | **40 - 240 KB/zone** |
| Access pattern | Regular, low spatial but high temporal locality |
| Communication pattern | Point to point, some volume |
| Mflops per zone per cycle | 2 – 12 |
| Nuclear data I/O (startup) | 0.3 - 12 MB currently sent to every process |

# Diffusion processes

- Used for radiation, magnetic fields and thermal fields
- Often use a linear solver library (e.g. HYPRE, TRILINOS) to solve the coupled equations implicitly
- Exascale challenges:
  - Has significant communication, both collective and point-to-point, which will have to scale
  - Convergence generally slows as size of problem increases
  - Can new solvers be developed that minimize communication?



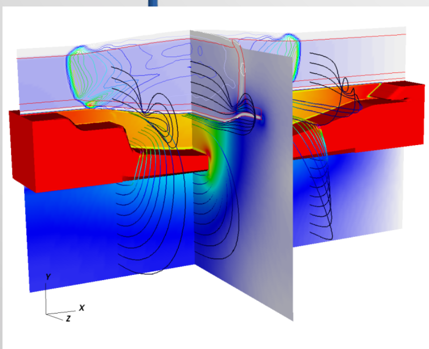

| Typical Characteristics | |
|---|---|
| Memory needs | 0.1 - 1 KB/zone |
| Access pattern | Regular, good spatial and temporal locality |
| Communication pattern | **Collective communications and point to point** |
| Mflops per zone per cycle | 0.1 - 3 |

# Monte Carlo transport

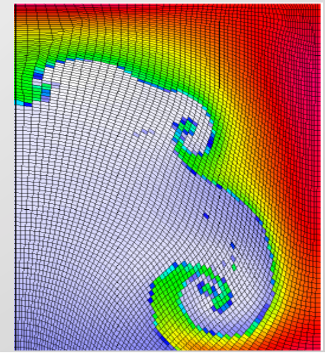
- **Parallelized over particles and space**
- **Some lessons learned on Roadrunner**
- **Exascale challenges:**
  - **Few flops per memory access places premium on data locality**
  - **Communication patterns are random with some volumetric communication**
  - **Load balance varies considerably over space and time**

| **Typical Characteristics** | |
|---|---|
| Memory needs | **3 - 30 KB/zone** |
| Access pattern | **Irregular, low spatial and temporal locality** |
| Communication pattern | Point to point, some volume |
| Mflops per zone per cycle | **.03 - .07** |
| Nuclear data I/O (startup) | **100 - 300 MB currently sent to every process** |

# Physics packages have differing computational requirements

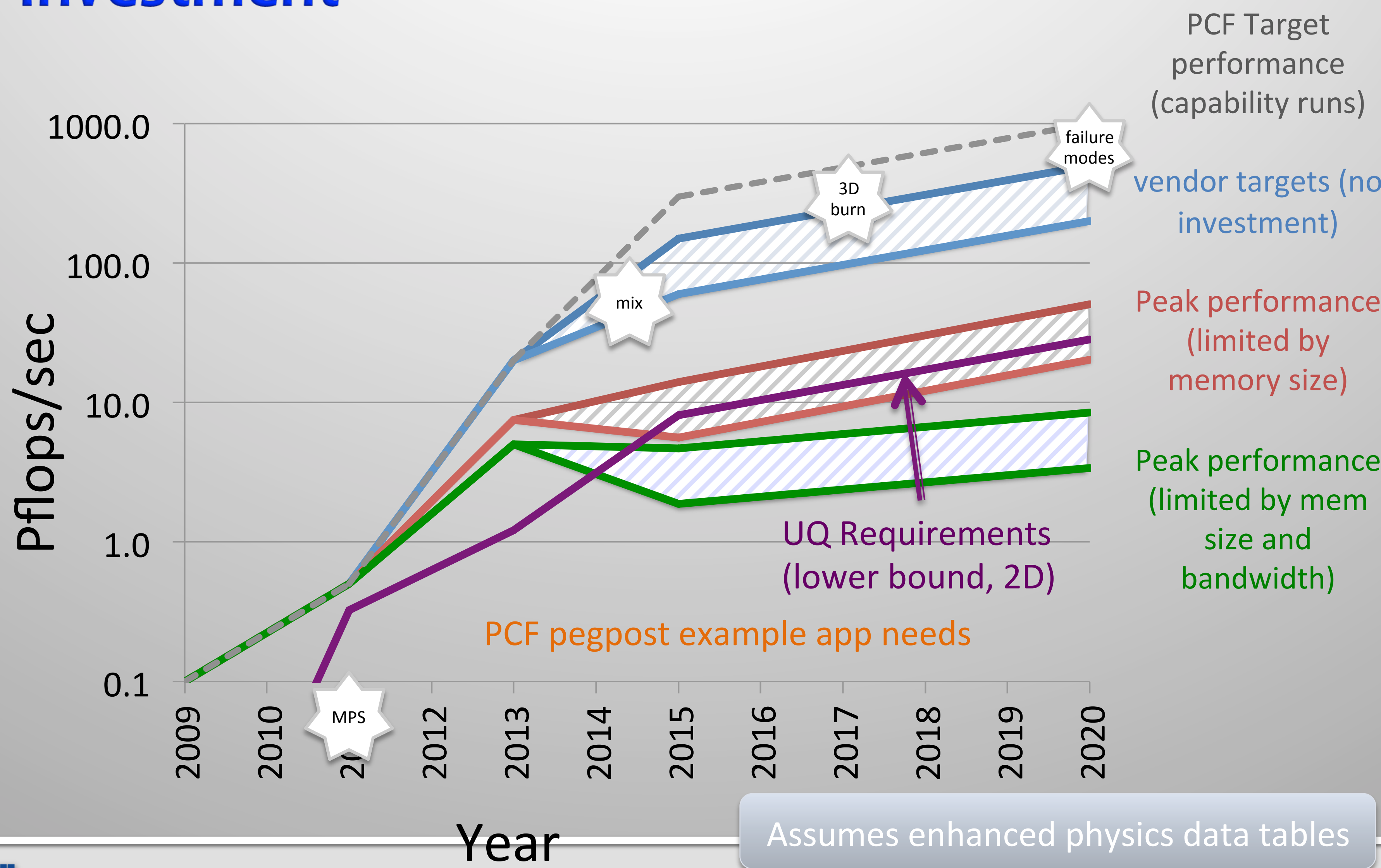


Multiphysics codes couple several different types of physics, some examples are



| Typical Characteristics | Hydrodynamics | Deterministic Transport | Monte Carle Transport | Diffusion |
|---|---|---|---|---|
| Memory needs | 0.1 - 1 KB/zone | **40 - 240 KB/zone** | 3 - 30 KB/zone | 0.1 - 1 KB/zone |
| Access pattern | Regular with modest spatial and temporal locality | Regular, low spatial but high temporal locality | **Irregular, low spatial and temporal locality** | Regular, good spatial and temporal locality |
| Communication pattern | **Point to point, surface communication** | Point to point, some volume | Point to point, some volume | **Collective communications** and point to point |
| Mflops per zone per cycle | 0.02 – 0.1 (10X for iterative schemes) | 2 – 12 | **.03 - .07** | 0.1 - 3 |
| I/O (startup data) | **20-160 MB (EOS)** | 0.3 - 12 MB (Nuclear) | **100 - 300 MB (Nuclear)** | 0.1 - 1 KB/zone |

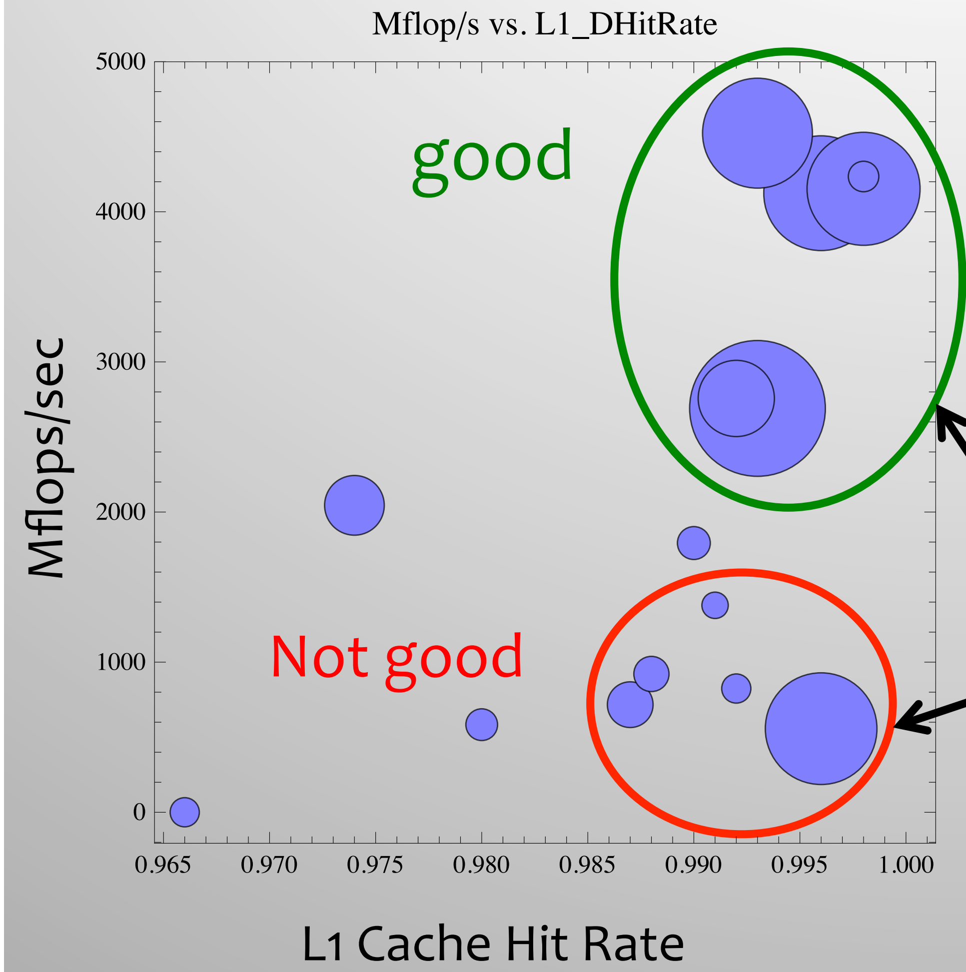
# Closing the gaps requires a Co-Design Investment




Assumes enhanced physics data tables

AATEMPS

# We developed metrics and applied to proxy apps to quantify impact of algorithmic transformations



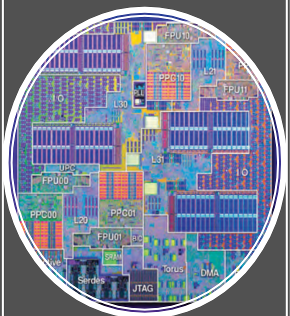

- **Each bubble represents a routine**
- **Bubble size is runtime spent in routine**

**This is the result of a good data structure versus a bad data structure**

# Addressing many concerns is critical for success

- **Architecture – floating point, integer performance and memory size/bandwidth**
- **System Software – fault detection/recovery, process migration, power management**
- **I/O and Networking services – checkpoint sets, data tables**
- **Viz and Data Analysis – in-situ methods and data exploration**
- **Tools – development/debugging, performance analysis**
- **Programming models – standardized for performance portability**
- **Solvers, Algorithms, Libraries – new data-aware methods**

# Key research investments directed toward exascale are needed inside an R&D framework

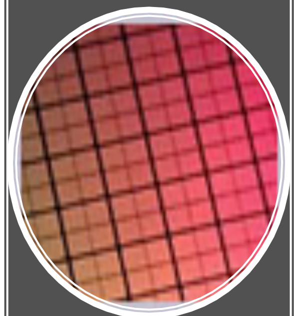


### Platform R&D

- Node architecture
- Systems architecture
- Interconnect
- I/O
- App performance
- App debugging
- System RAS
- Packaging

### Platforms

- Favorable contracting framework
- Flexible risk sharing contracting
- Prototype evaluation
- Flexible critical decision and EVMS processes
- Successful long term partnerships with industry

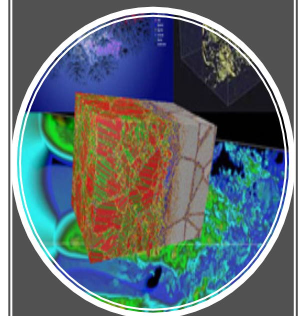


### Critical Technologies

- File systems
- Nonvolatile storage
- Networking
- Programming model

### Software and Environments

- System mgmt
- Thread mgmt
- Solvers
- App dev env
- Resource mgmt
- Sci vis data mgmt
- Fault oblivious computing
- Security
- App performance
- Power aware computing



### Applications

- ASCR co-design
- NNSA co-design
- Simulations
- Multi-physics at exascale
- Vanguard efforts – grand challenge
- Methods for co-design with sensitive apps

***With ASC we have a discipline of coordinating investments in computer R&D, computers, pathforward efforts, software, and integrated codes***

# Summary

- Scientific modeling of complex multi-physics phenomena is a key capability
- Coupled multiphysics simulation codes are very complex and represent a substantial investment
- Exascale technology will present a significant departure from current architectures
- Targeted R&D investments are needed to address key challenges
- Mitigating the risk involves identifying and managing the disruption as early as possible

Lawrence Livermore
National Laboratory