SAND2012-3685P

Enabling Exascale Co-design with The Structural Simulation Toolkit

Gilbert Hendry

8953: Scalable Modeling and Analysis Systems

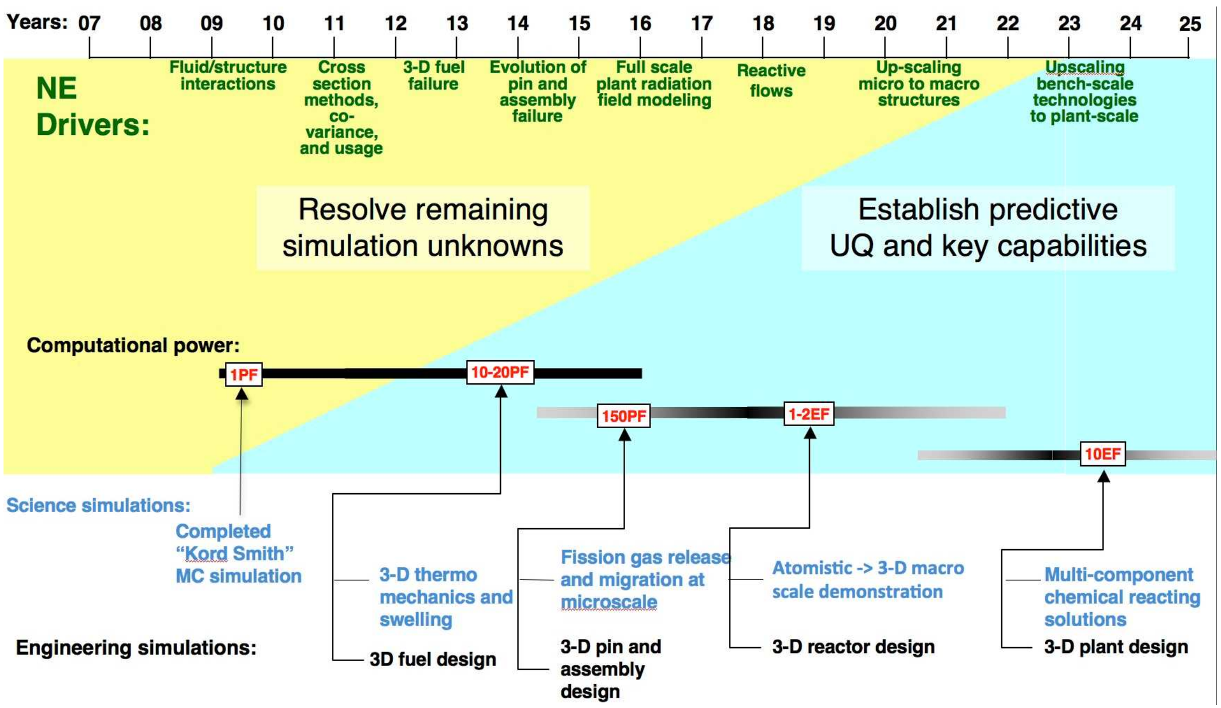
CIS External Panel Review

May 8-10, 2012

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Exascale Computing



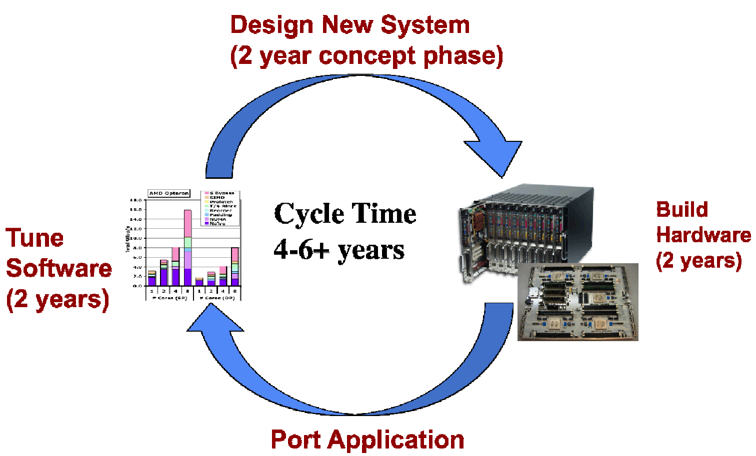
Source: John Shalf, Exascale Initiative Steering Committee

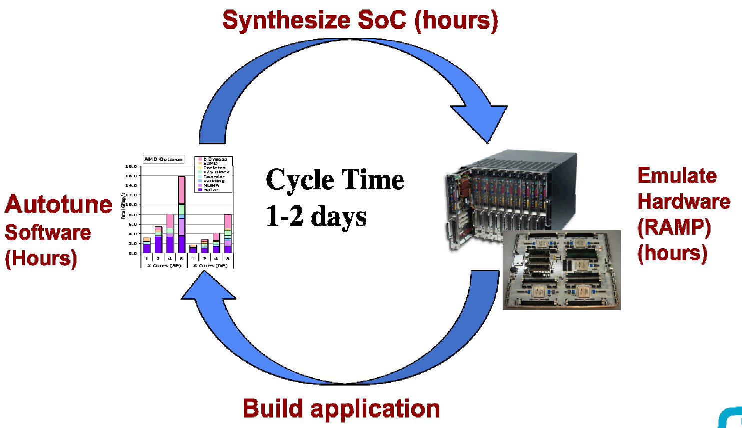
Exascale = 100x more computing = 100x more power and energy = 100x more $$

Sandia National Laboratories

Co-Design in the Exascale Era

- HPC is approaching some hard constraints, time to learn from embedded computing
- Co-Design is a process that involves tight coupling between hardware and software design which results in a more effective and efficient computing environment





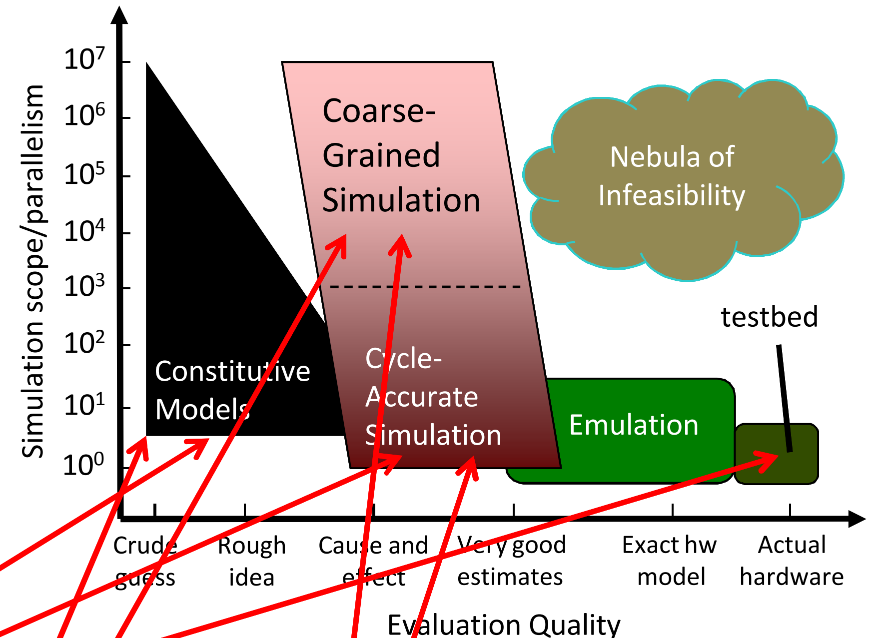
Sandia National Laboratories

Co-Design and Simulation at Sandia

- Sandia has a solid history of leadership in the design of HPC machines and software
- We have the scientific codes (the software) and the scientists who understand them.
- We buy the machines, in our interest to do our due diligence before dumping money in. (vendor proposal evaluation).
- We don't compete with anyone, have a neutral stance. Can also keep secrets (NDA).
- Can still work with both multiple industry partners and academia
- We have resources that academia usually doesn't have, providing better quality of software. We can work on something even if it doesn't produce a paper, and maybe is not very research-y.

Co-Design Evaluation Tools

- Constitutive Models – can be powerful in reasoning about system and tradeoffs, but hard to investigate new concepts and complex interactions
- Coarse-Grained Simulation – accurate, predicts trends, can scale, but many detailed studies not possible
- Cycle-Accurate Simulation – highly accurate for detailed studies, but can only scale so far
- Emulation – essentially exact and fast, but expensive to scale
- Testbed/prototype – provides real numbers, but time/cost a major factor



Example Issues:

Power Requirements

Programming Models and Application Transformation

Resilience, IO, and Performance

Simulation in the Exascale Design Space

SST/macro

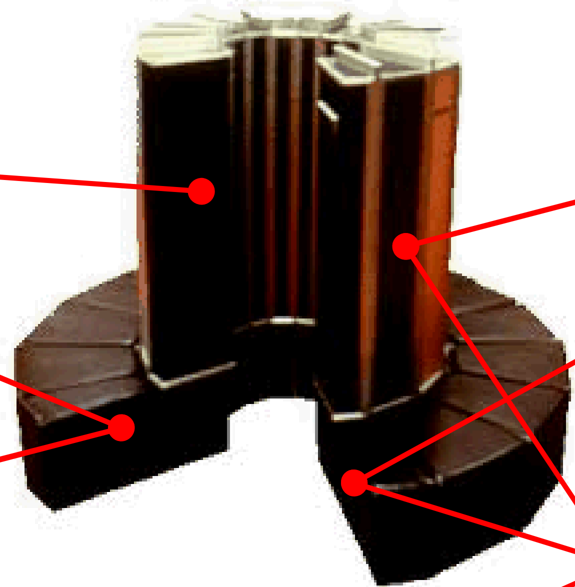
Network architecture/topology

Application development

Library/interface/services support

Machine fault tolerance

File system, I/O



(supercomputer)

SST/micro

Network switch implementation

Node Architecture

Memory technology

Processor/GPU

Power

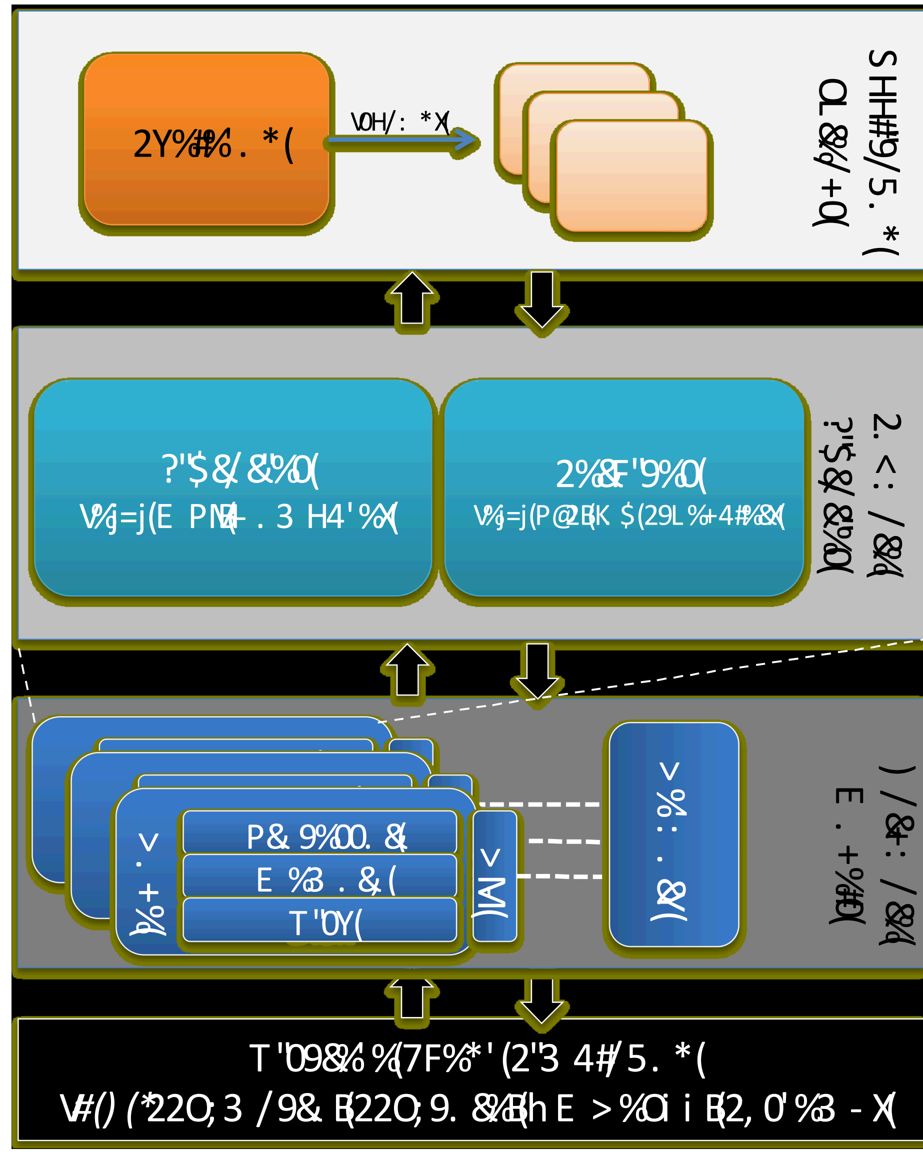
Meso-Scale Simulation

Ask implementation-related questions at scale.

SST/macro: Coarse-Grained Simulation

An application code with minor modifications

Our implementation of interfaces (MPI), which simulate execution and communication



Related Work:

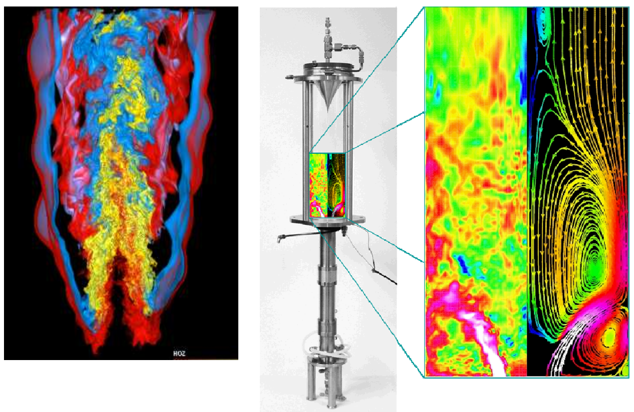
xSim (Oakridge) – only MPI

BigSim (UIUC) – only MPI, needs a big machine

SST/Macro Projects

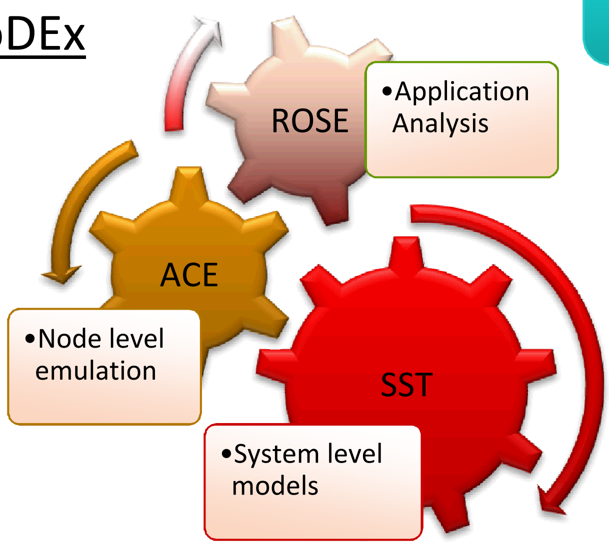
ASCR Execution Models

ASCR ExaCT: Combustion Codesign



SST/Macro

ASCR CoDEx



- SST/Macro is a key tool in a number of exascale/co-design efforts.
- It brings key capabilities that work synergistically with other tools to form a complete picture

Mini Apps: An efficient vehicle for Co-Design

<u>Mini-apps</u>:

(sometimes called compact apps, or reduced app)

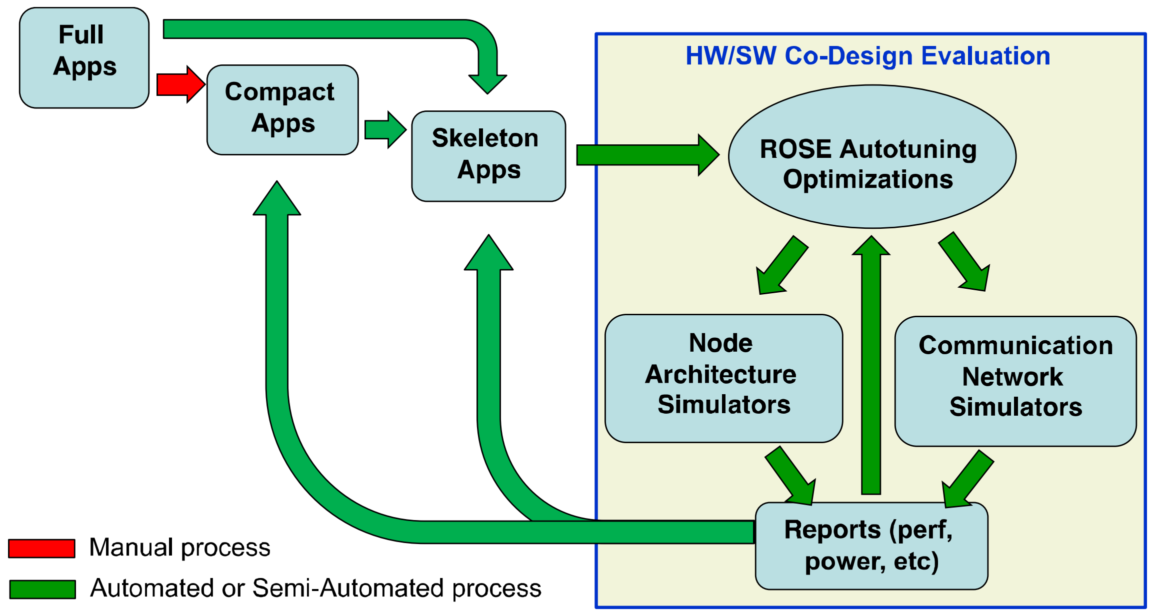
- A more efficient vehicle for co-design

<u>Skeletons:</u>

- A type of mini-app which strips out everything except communication and control
- Run 2-10x *faster* than real execution

(as opposed to 10-1000 *slower* in cycle-accurate)

- Design vehicle for fast prototyping



code:

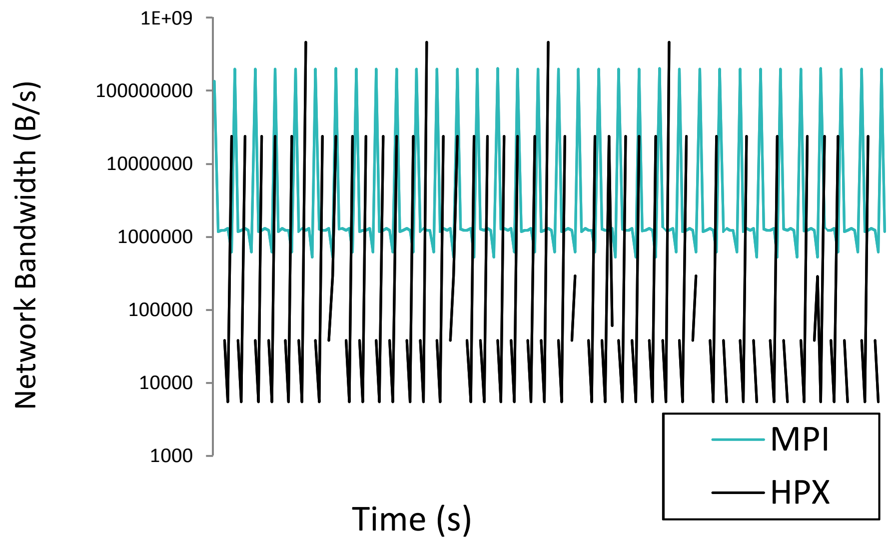
```
int x = rank % 2;

int count = 100;
int* buf = (int*)malloc(count * sizeof(int));

if(x == 0){
    for(int i = 0; i < count; i++){
        buf[i] = rand();
    }
    MPI_Send(buf, count, MPI_INT, rank + 1, 0, MPI_COMM_WORLD);
}else{
    MPI_Recv(buf, count, MPI_INT, rank - 1, 0, MPI_COMM_WORLD);
}
```

skeleton:

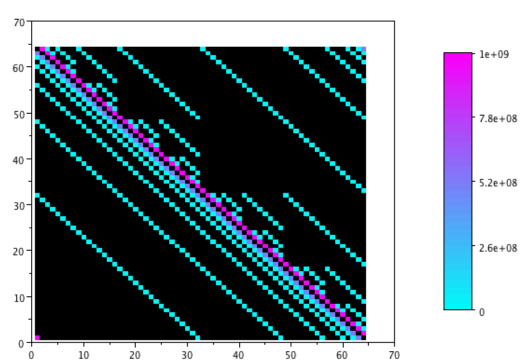
```
int x = rank % 2;

int count = 100;
/*model memory allocation here
 */
if(x == 0){
    /*model computation here
     */
    MPI_Send(NULL, count, MPI_INT, rank + 1, 0, MPI_COMM_WORLD);
}else{
    MPI_Recv(NULL, count, MPI_INT, rank - 1, 0, MPI_COMM_WORLD);
}
```

Using SST/macro – Execution Models

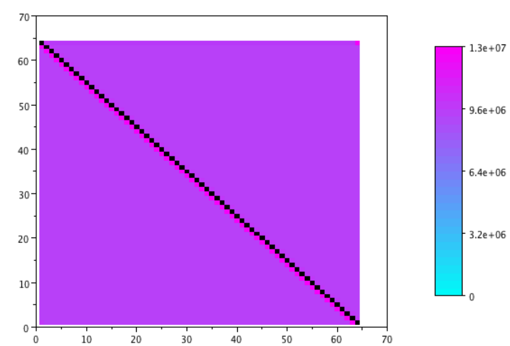
- Exploring execution models with HPX (Parallex)
- Using GTC (particle-in-cell fusion app) as test vehicle



MPI

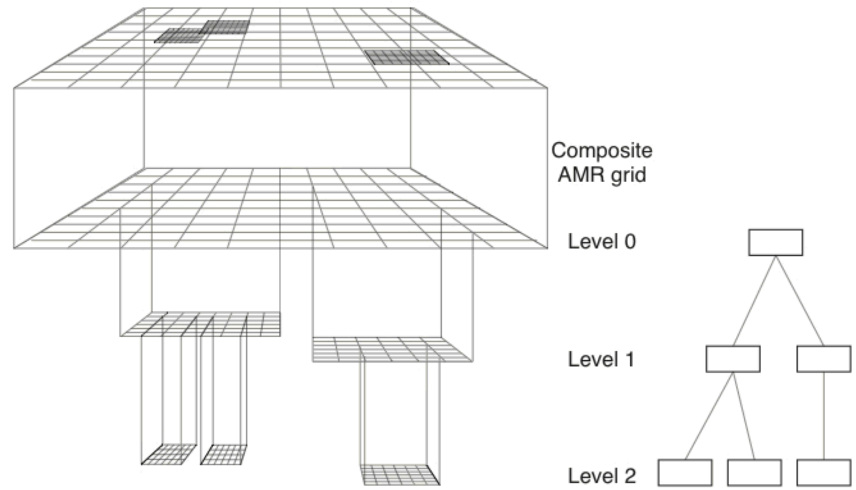


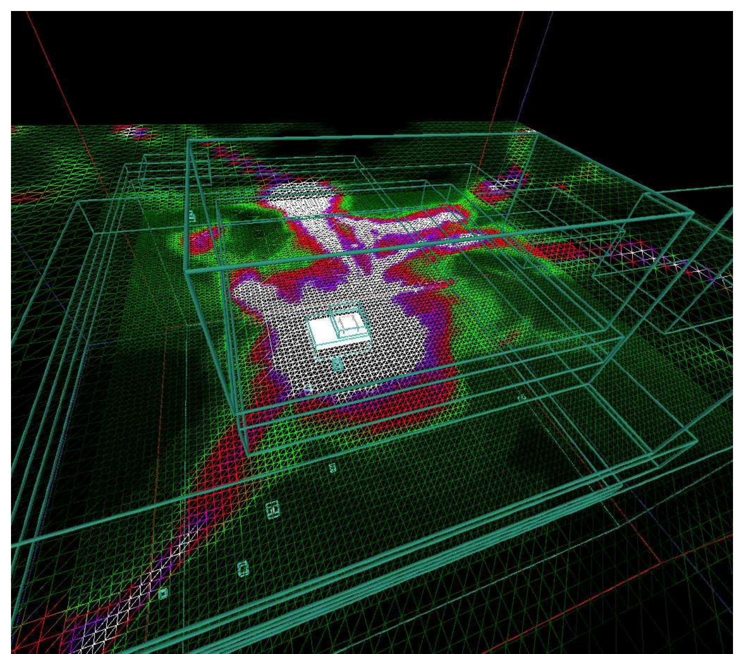
Parallex



Using SST/macro - Combustion

- Adaptive Mesh Refinement (AMR) important for efficiently looking at regions of space
- Very data-dependent, so hard to make a skeleton
- Porting parts of BoxLib (LBL) into SST/macro to investigate box generation and layout at large scales

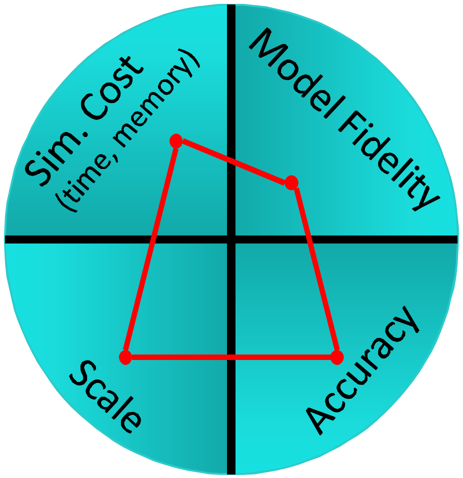




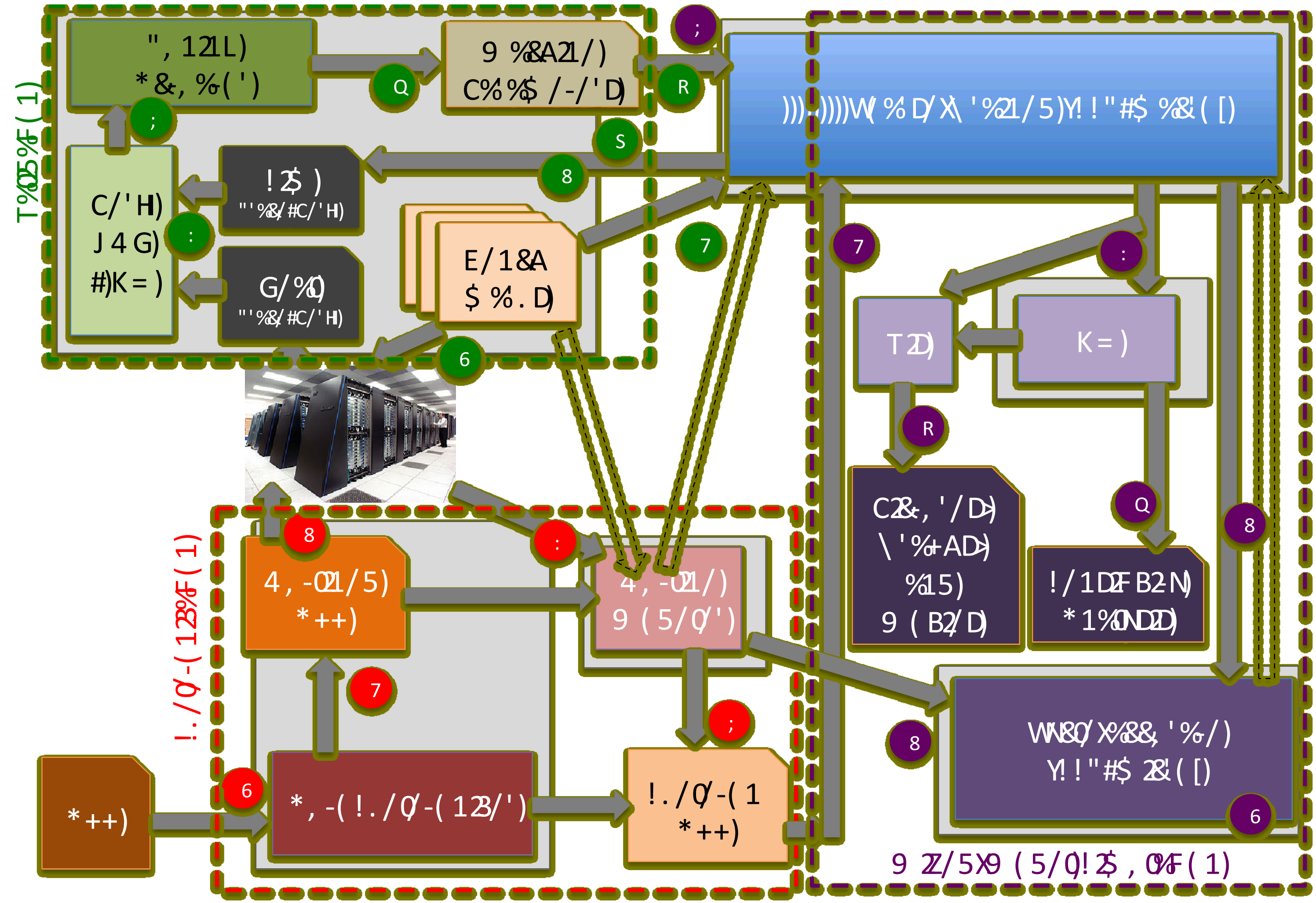
Sandia National Laboratories

Creating Efficient Simulation

- Example 1: What network topology is best for an application, a class of applications, or all applications of interest?
  - Use coarse-grained simulation
- Example 2: How do different memory technologies effect shared memory performance characteristics on a multi-core node?
  - Use cycle-accurate simulation
- Example 3: What hardware support would we want to see for many-tasking (threading) in very large, very parallel applications?
  - Need smart mix of simulation models.



Mixed-Model Simulation



Summary

- SST/macro is an integral part of the suite of simulation and evaluation tools necessary for doing exascale co-design
- SST/macro is fairly mature, currently being used to build simulations in interesting areas
- Looking forward, need to build framework and capability to quickly build mixed model simulation using automated processes