



# Experiences in Performance Modeling: The Krak Hydrodynamics Application

**Kevin J. Barker**

**Scott Pakin and Darren J. Kerbyson**

**Performance and Architecture Laboratory (PAL)**

**<http://www.c3.lanl.gov/pal/>**

**Computer, Computational, and Statistical Sciences Division**

**Los Alamos National Laboratory**



What characteristics should a good performance model have?

## 1. Predictive capability

- Variations in component performance (network, processor)
- Variations in system size
- Variations in network architecture/topology

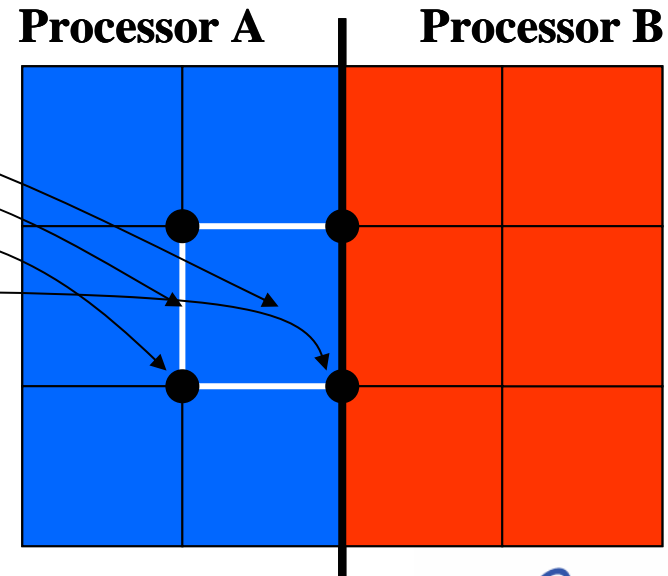
## 2. Simplicity

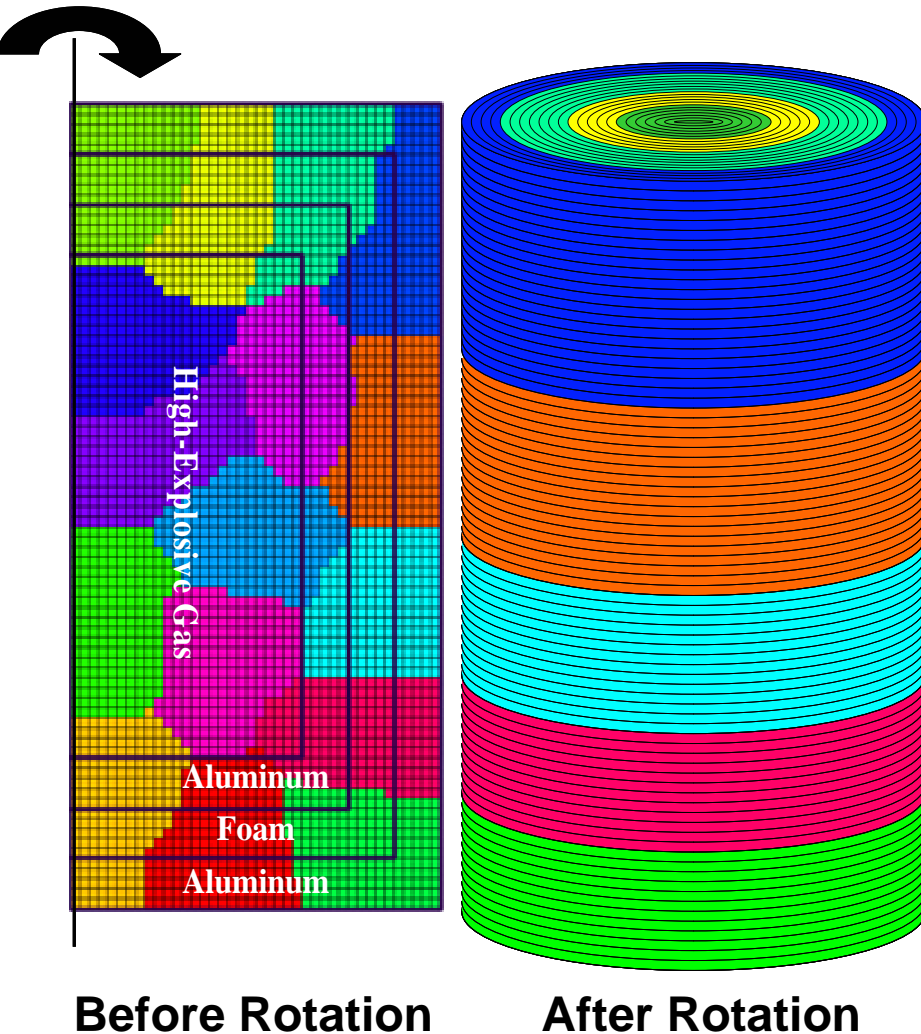
- The performance model should only capture those elements which actually impact application performance

## 3. Accuracy

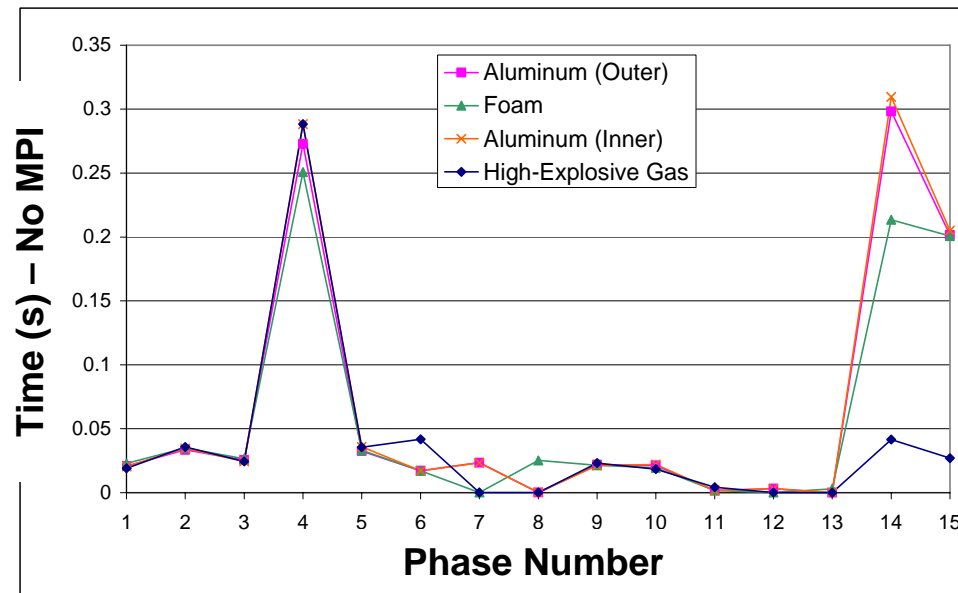
- How well do the model's predictions compare against measured application runtimes on current systems?

- **Production hydrodynamics code developed at LANL**
  - Simulates forces propagating through objects composed of multiple materials
  - >270K lines of code, >1600 source files
  - Object-oriented Fortran dialect
- **Typically executes in strong-scaling mode (fixed global domain size)**
- **Objects mapped onto grid**
  - Grid composed of “cells”
  - Cell defined by “faces”
  - Faces connect “nodes”
  - “Ghost nodes” on PE boundary
- **Processing flow moves through series of time-steps that calculate object deformation caused by high-energy forces**





- **Three grid sizes studied**
  - Small : 3,200 Cells
  - Medium : 204,800 Cells
  - Large : 819,200 Cells
- **Cells contain one of three material types**
  - Aluminum
  - Foam
  - High Explosive (HE) Gas
- **Regular grid decomposed into irregular sub-domains (colors – shown for 16 processors)**
- **Metis partitioning optimized for edge-cuts leads to irregular domain shapes and sizes**



**Spatial grid of 65,536 cells on 256 PEs**

- **Computation divided into 15 phases per iteration, each modeled separately**
  - Separated by global reduction operations
  - 9 involve only computation
  - 6 involve communication as well
- **Computation time varies with**
  - Number of cells per sub-domain
  - Cell material composition
  - Phase

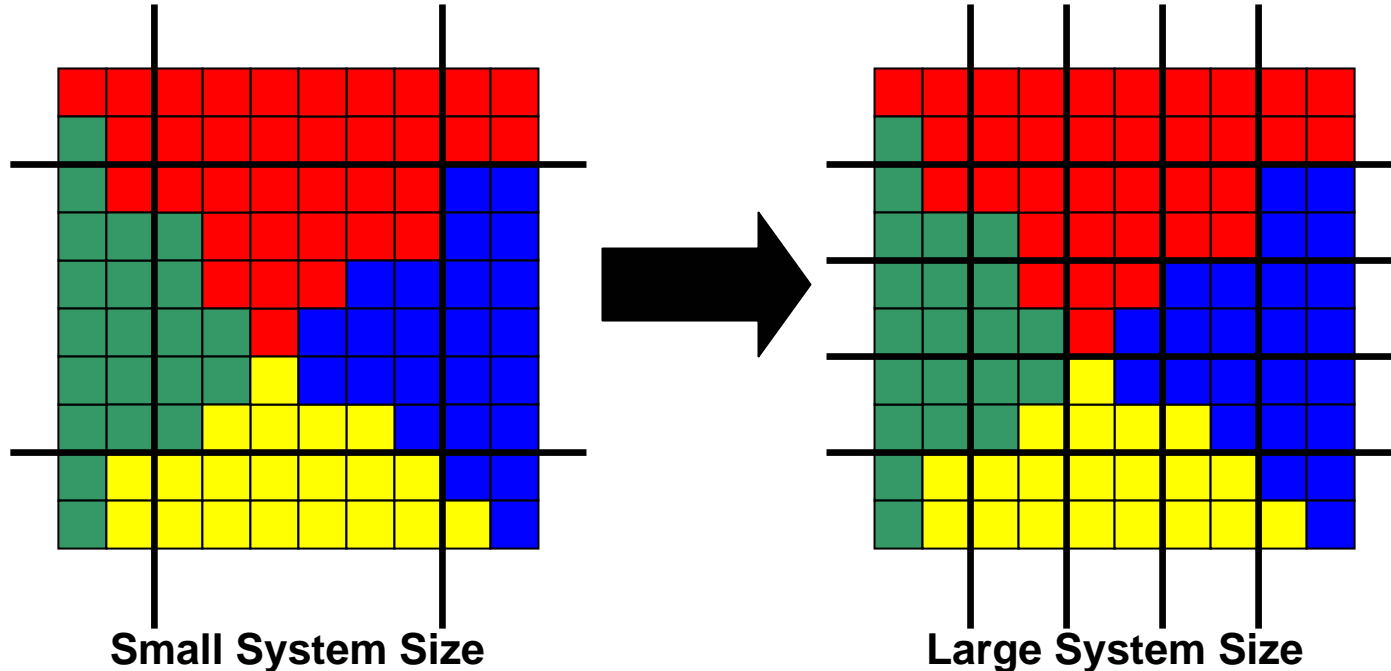
**Given the phase number, material composition, and local sub-grid size, the computation time per cell can be calculated using a piecewise linear equation**



- **What would a Krak performance model depend on?**
  - **Computation**
    - Per cell computation cost of each material
    - Number of cells of each material in each sub-grid
  - **Communication**
    - Boundary length between sub-grids
    - Collectives
- **All of these are determined by the exact partitioning of the input spatial grid, which cannot be known in advance**
- **Any resulting model would not satisfy goals of simplicity and predictive ability**
- **Abstraction is the key...**

Due to Strong Scaling:

1. Sub-grids become more homogeneous as system size increases (figure below)
2. Assuming each sub-grid to be square is reasonable at large system sizes

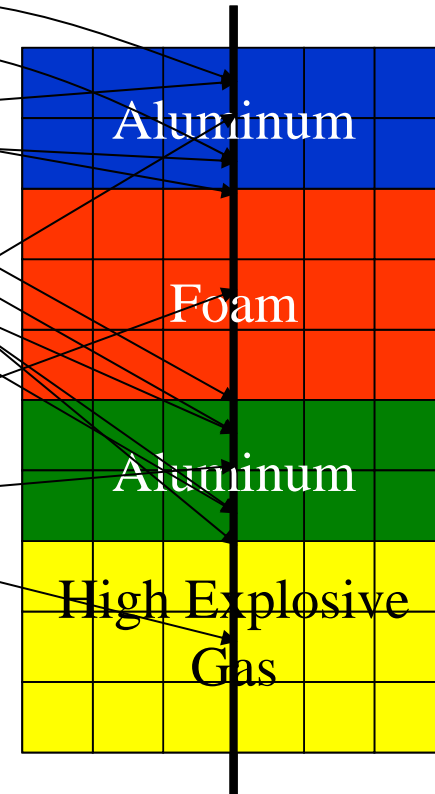




**Abstractions result in simplified performance model:**

- **Computation**
  - Each sub-grid contains the same number of cells
  - All cells are of the most computationally intensive material
  - All sub-grids are square in shape
  - Per-cell cost derived from measuring compute times of sub-domains of varying sizes
- **Communication**
  - Each sub-grid is modeled with four neighbors in 2D
  - All boundaries are the same length
  - All boundary faces touch only a single material
  - Three communication types:
    - Boundary Exchange
    - Ghost Node Updates
    - Collectives

Material	Msg. Count	Size of Each Msg (bytes) ( $N_B = 12$ bytes)
Aluminum (both)	2	$84 = N_B(2+2+3)$
	4	$48 = N_B(2+2)$
Foam	2	$60 = N_B(3+2)$
	4	$36 = N_B 3$
H.E. Gas	2	$48 = N_B(3+1)$
	4	$36 = N_B 3$
All	6	$120 = N_B(2+3+2+3)$



- $N_{\text{materials}} + 1$  total steps, each comprising 6 msgs
- Message sizes depend on the number of material boundary faces and ghost nodes
  - 12 bytes per boundary face
  - 12 bytes per ghost node on material boundary

# Ghost Node Updates

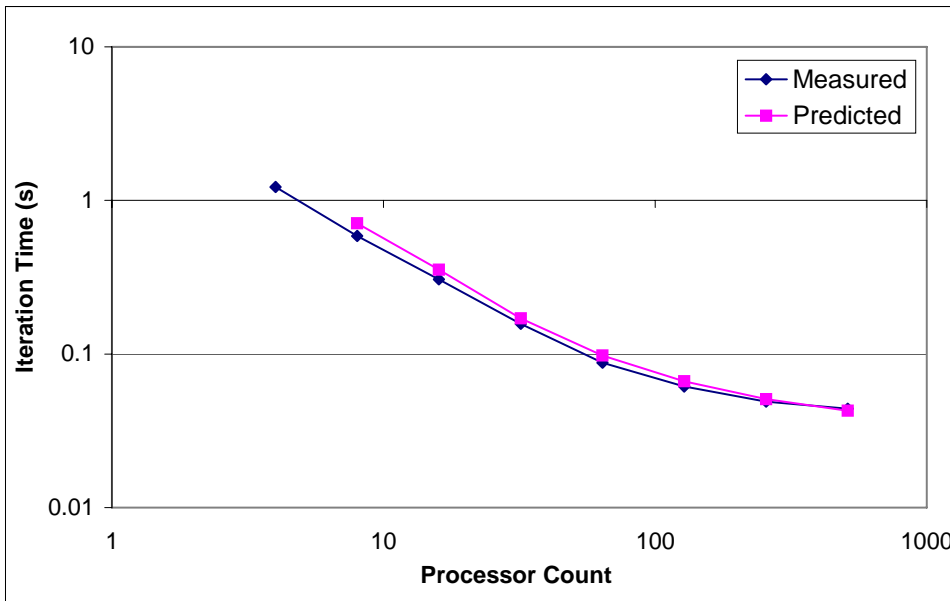


- **Ghost nodes exist on processor sub-grid boundaries**
- **Each ghost node is described as:**
  - Local to the “owning” processor
  - Remote to all other processors whose boundaries contain that ghost node
- **Ghost node updates take place in three application phases**
  - Phase 4
    - 8 bytes for each local ghost node to all neighbors
    - 8 bytes for each remote ghost node to all neighbors
  - Phases 5 and 7
    - 16 bytes for each local ghost node to all neighbors
    - 16 bytes for each remote ghost node to all neighbors

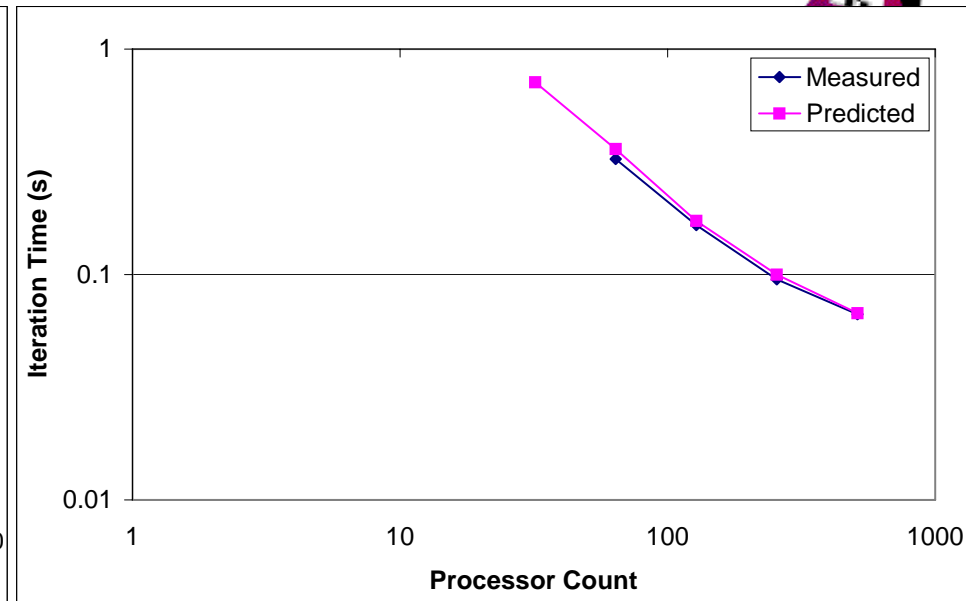


- Three types of collective communication per iteration
- Collectives modeled as “fan-in” or “fan-in, fan-out” point-to-point messages
- Message sizes and counts are independent of input deck

Type	Count	Size (bytes)
MPI_Bcast()	3	4
	3	8
MPI_Allreduce()	9	4
	13	8
MPI_Gather()	1	32



### Medium Problem Size



### Large Problem Size

- **Homogeneous material distribution more realistic for large processor counts**
- **Error less than 3% at 512 processors**
- **Communication overheads overwhelm benefits of increased parallelism at large processor counts**



- **Described an analytic performance model for a complex hydrodynamics code**
- **Challenges:**
  - Irregular mesh partitioning among processors
  - Variations in per-cell times based on material properties
  - Variation by phase
- **Careful abstractions lead to a simplified, accurate, and predictive performance model**
  - Approximate sub-grid composition and shape
  - Accurate representation of reality at large processor counts
  - Low error (< 3%) at large scale

# Acknowledgements



CCS-3

- The ASC program at Los Alamos National Laboratory
- The authors wish to thank Hank Alme and Scott Runnels at Los Alamos National Laboratory for their assistance in obtaining Krak source code and input decks

## Thank You! Questions?