SAND2020-1307PE

Improving the Scalability and Performance of the Sandia I/O Software Stack

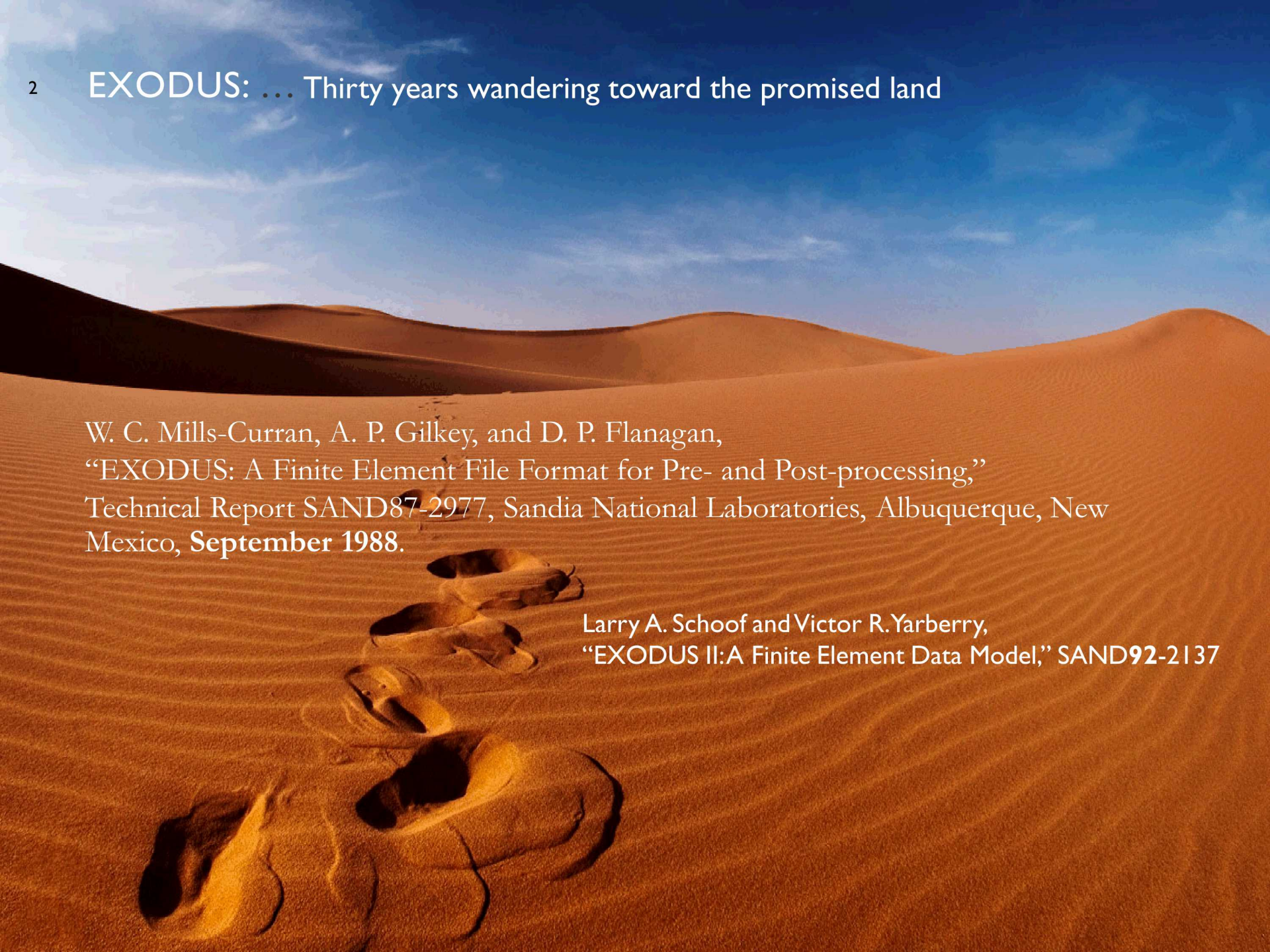
COMPSIM

ACS Meeting, LLNL, February 24-27

Gregory Sjaardema, Engineering Sciences Center, SNL

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & 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.

EXODUS: … Thirty years wandering toward the promised land



W. C. Mills-Curran, A. P. Gilkey, and D. P. Flanagan,
"EXODUS: A Finite Element File Format for Pre- and Post-processing,"
Technical Report SAND87-2977, Sandia National Laboratories, Albuquerque, New Mexico, **September 1988**.

Larry A. Schoof and Victor R. Yarberry,
"EXODUS II: A Finite Element Data Model," SAND**92**-2137

Exodus History

- Genesis 1986, Exodus 1987
  - fortran-unformatted read/write files
- Exodus II 1992
  - device-independent, random access database. Based on NetCDF
- Exodus now refers to the Exodus II version

- Single format used by all Sandia Finite Element Codes
- Used externally by National Labs, Universities, and Commercial companies

- Nodes, Homogenous Element Blocks, NodeSets, Sidesets
  - Transient Data on Nodes and Elements.

- SEACAS: A Suite of mesh generation, preprocessing, postprocessing, visualization, and translation tools developed to generate and modify exodus files.

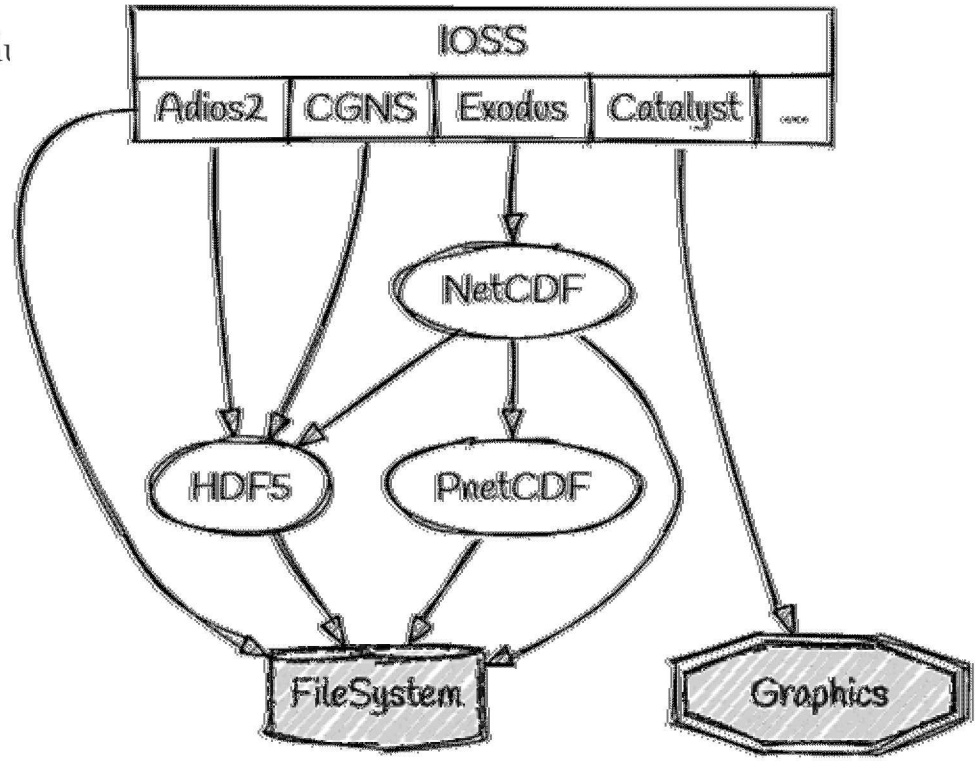
- Backward Compatibility Essential

Y-MP with eight vector processors,
6 ns clock cycle time (167 MHz).
333 megaflops per processor.
128, 256 or 512 MB of SRAM.

4 MB memory board for the VAX 8600

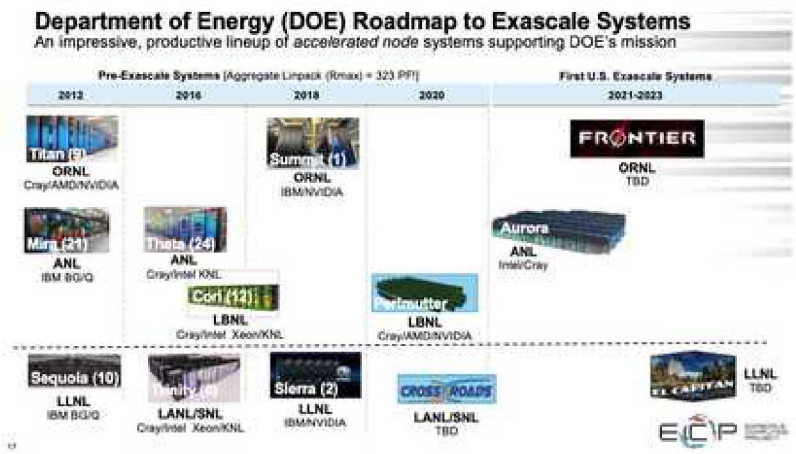
I/O Subsystem – IOSS Library

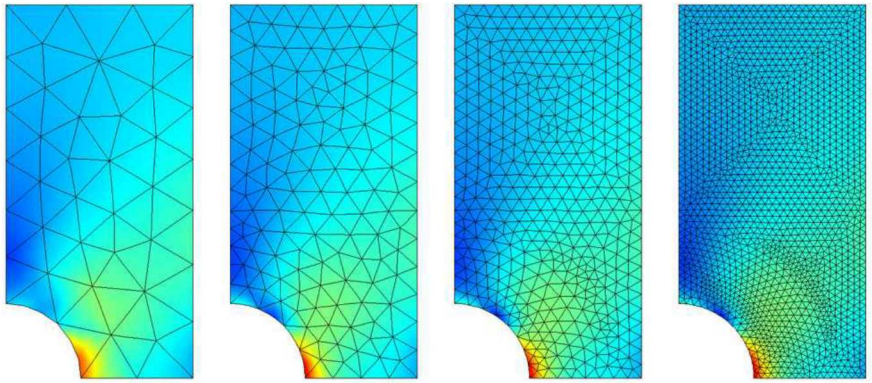
- Started as the IO component of the Sierra project – 12/1999
- Provide a database-independent interface to applications
  - (Exodus, CGNS, SAF, XDMF, ADIOS2, …)
- Also functions as a "pseudo-C++ API" for exodu
- Supports Advanced HPC Capabilities:
  - Kokkos Data
  - Burst Buffer
  - Data Warehouse (FAODEL)
  - Embedded Visualization



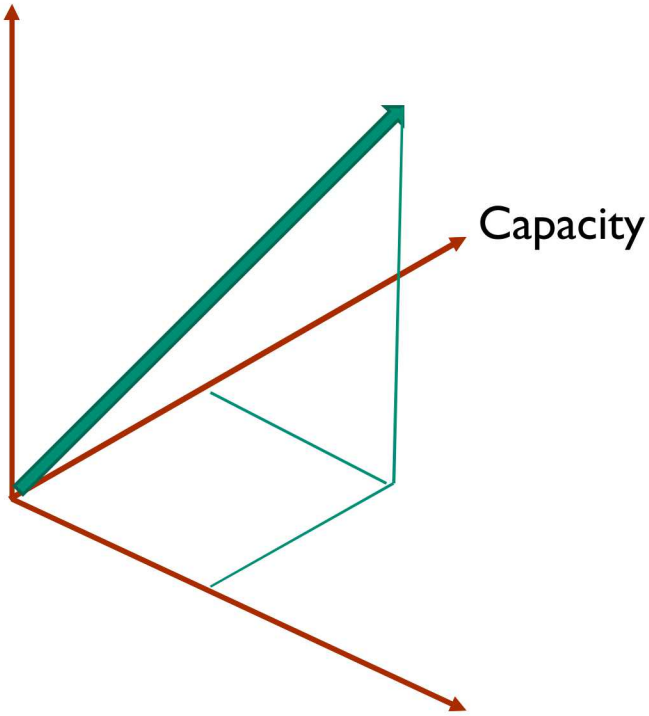
- Auto-decomposition
  - Replaces legacy file-per-processor mode
  - Uses either HDF5 or PnetCDF for parallel input
  - Uses decomposition methods in Zoltan and ParMETIS
  - Supports Exodus and CGNS (Structured and Unstructured)

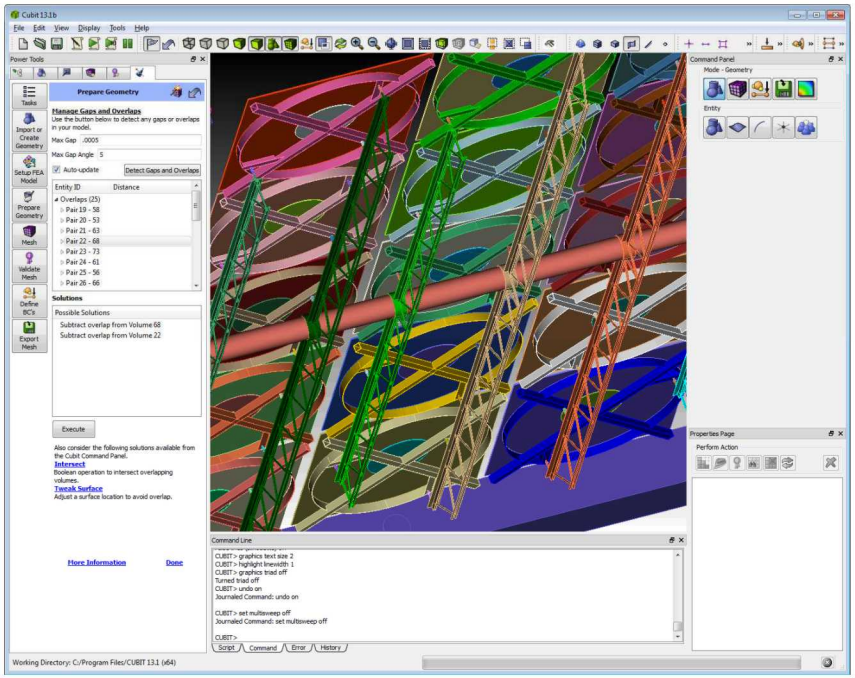
- Auto-join (single file output) option
  - Uses HDF5 or PnetCDF for parallel output
  - Scalability issues.... Being addressed.





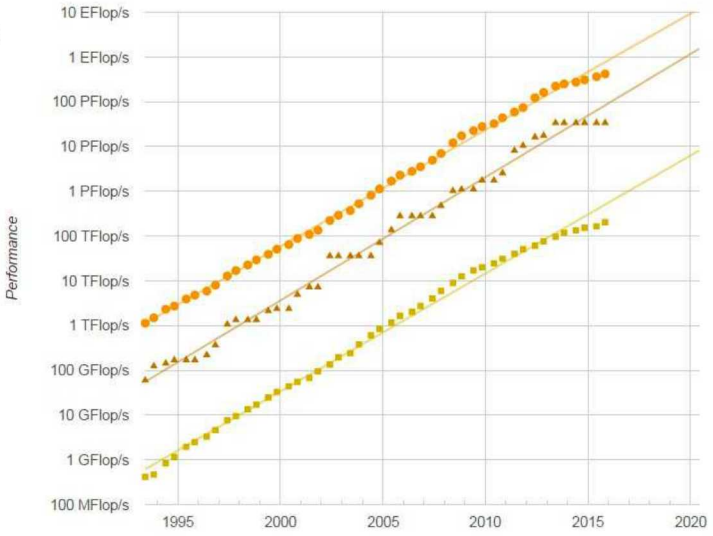
Scalability





Exodus Evolution -- Capacity

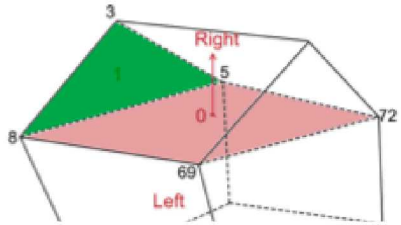
- Original: ~34 Million Nodes/Elements (~NetCDF-2.3.X, CDF1)
- 2004: ~500 Million Nodes/Elements (NetCDF-3.6.0, CDF2)
  - Internal changes to exodus format to reduce dataset sizes, new NetCDF variant
- 2008: 2.1 Billion Nodes/Elements (NetCDF-4, HDF5 based)
- 2012: 64-bit Integer changes ?? Billion Nodes/Elements
- 2017: NetCDF-4.5.1
  - No NetCDF modifications required for use with Exodus
  - NC_MAX_DIMS, NC_MAX_VARS modifications no longer required
  - Can build exodus using a system-installed NetCDF library
  - Can pull and build from NetCDF repository with no modifications
- **Backward Compatibility Essential**

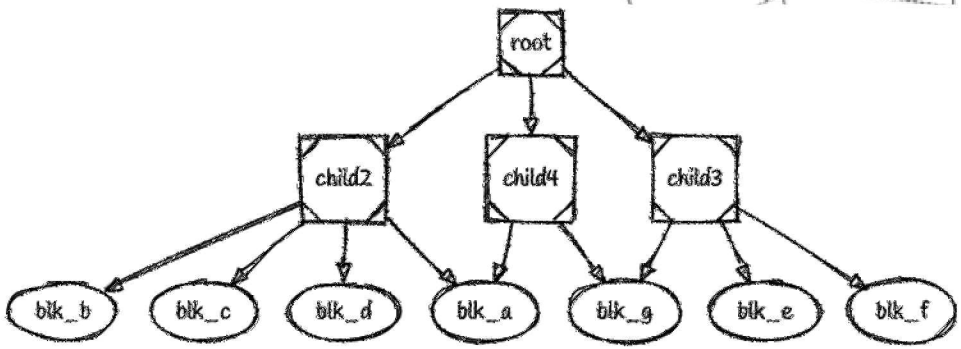


Exodus Evolution – Capability → Complexity

Capability:

- Named blocks, sets, attributes, and maps
- Transient variables on all blocks and sets
- "unlimited" string size replaced fixed 32 character limit
- Full topology support (Element->Face->Edge->Node)
- Arbitrary Polyhedral element support
- Compression
- Assemblies
- Blobs
- Entity Attributes (Key-Value pairs)
- Reduction Variables
- Thread-Safe





Exodus Evolution -- Complexity

—8,192 element block model

Routine	Original	Current	Ratio
Total Execution	36,329,205,355	6,842,180,452	5.3
NC3_inq_dimid	14,114,692,688	1,418,784	~10,000
get_element_block	3,318,934,624	422,122,570	7.9
NC3_def_dim	1,077,126,927	1,443,169	~750
NC3_inq_varid	712,439,902	2,064,893	~345
write_meta_data	3,540,642,840	639,142,440	5.5

1,044 blocks,
12 element variables/block:

I applied your patch and committed it to the CTH source.

Without the patch a test calculation took ***~25minutes.***

With the patch the same calculation took ***~1 minute****.*

This is a huge improvement!

Reduced io_shell time (read/write)

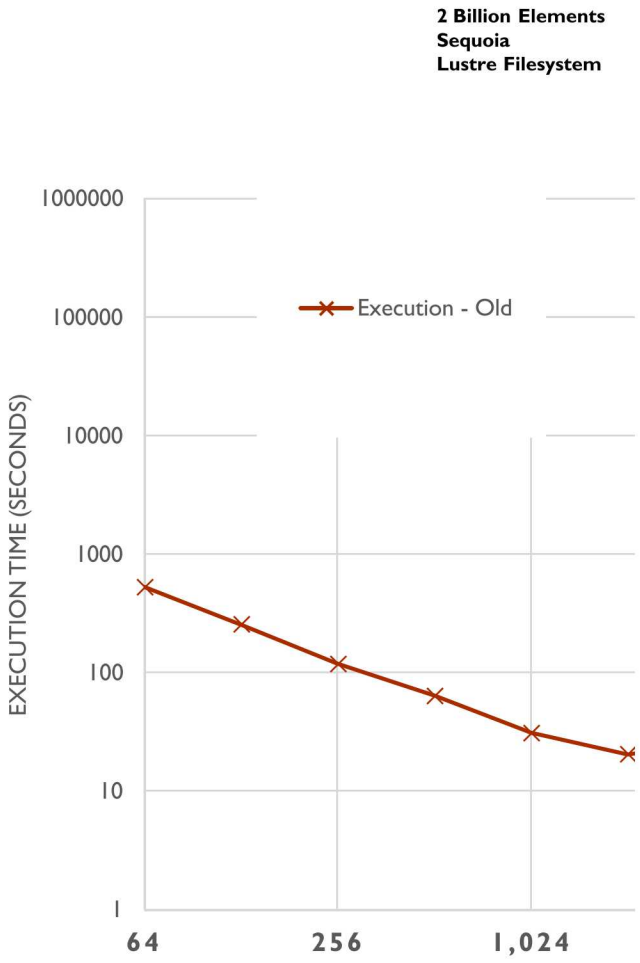
2176 element variable mesh:

before fix: 10m32.052s
after fix: 9.674s

Combination of
--- internal NetCDF changes
--- application usage changes

Can reduce time by an additional~25% -- 40%
Some easy, some will need some convincing

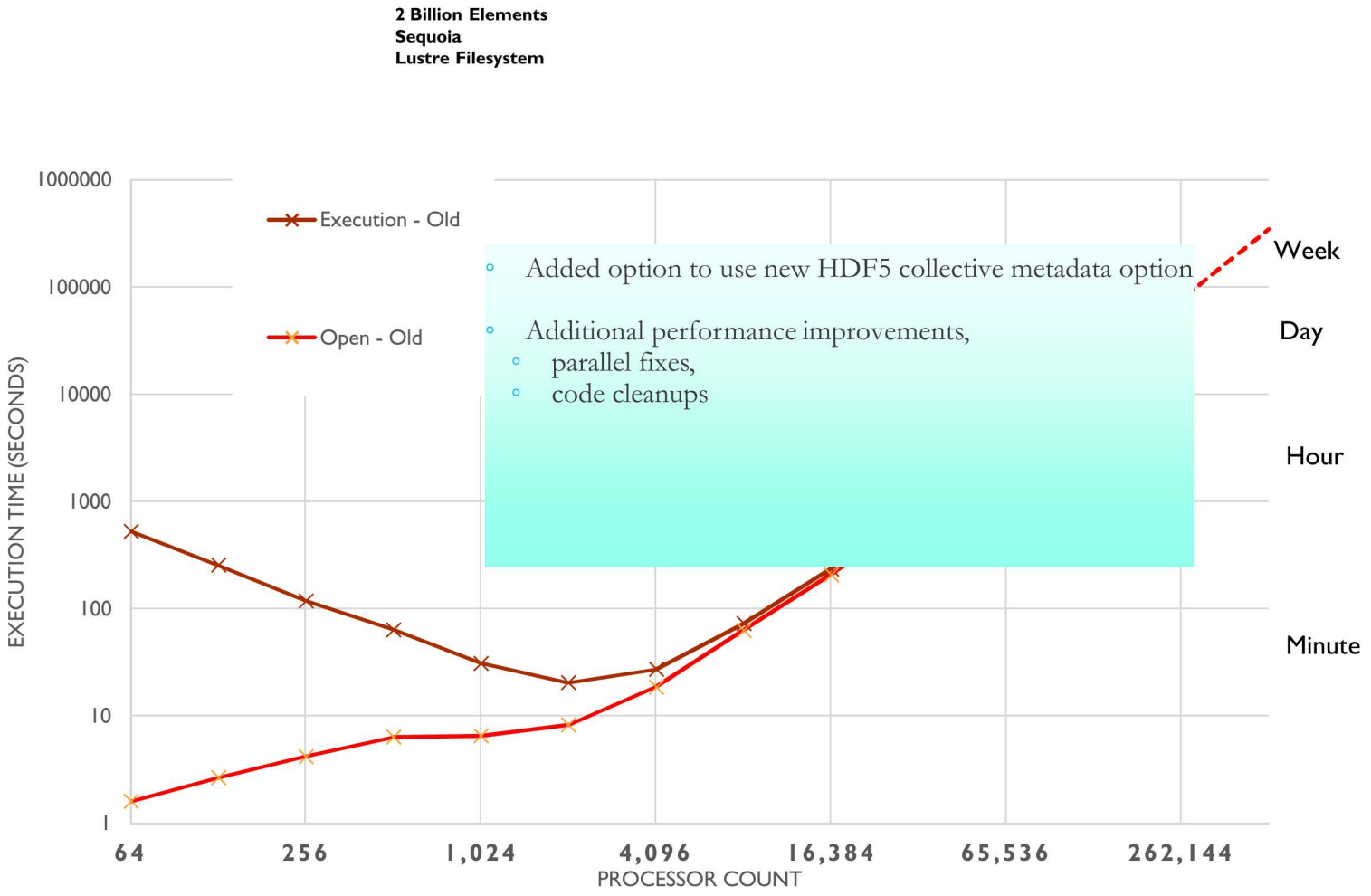
Evolution -- Parallel Scalability



Large model being run on Sequoia
Application reported "analysis is hanging"
Ran same mesh in simulator with tracing enabled to determine what was happening

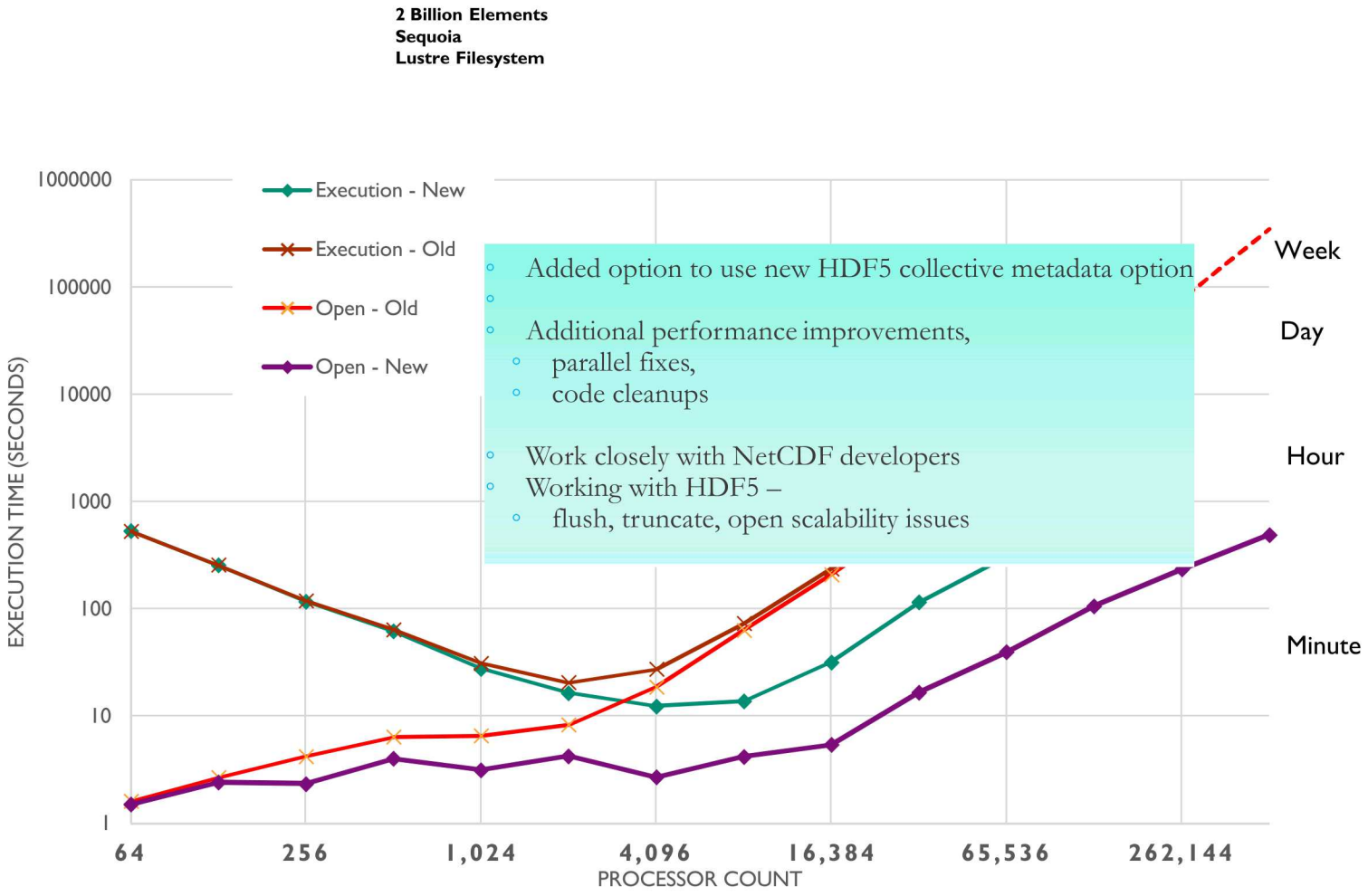
Initial results looked promising…

Parallel Scalability



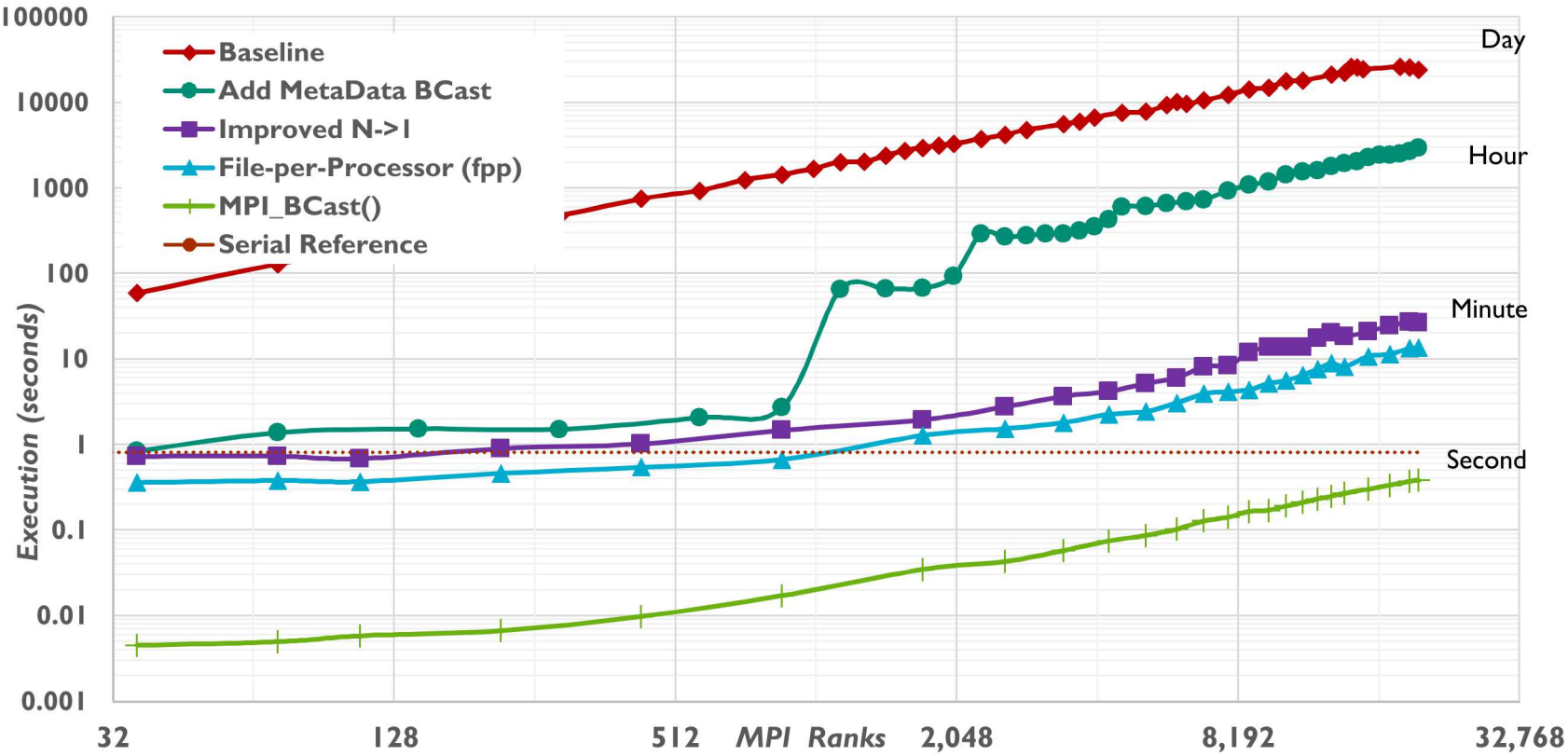
A supercomputer is a device for turning compute-bound problems into I/O-bound problems

Parallel Scalability

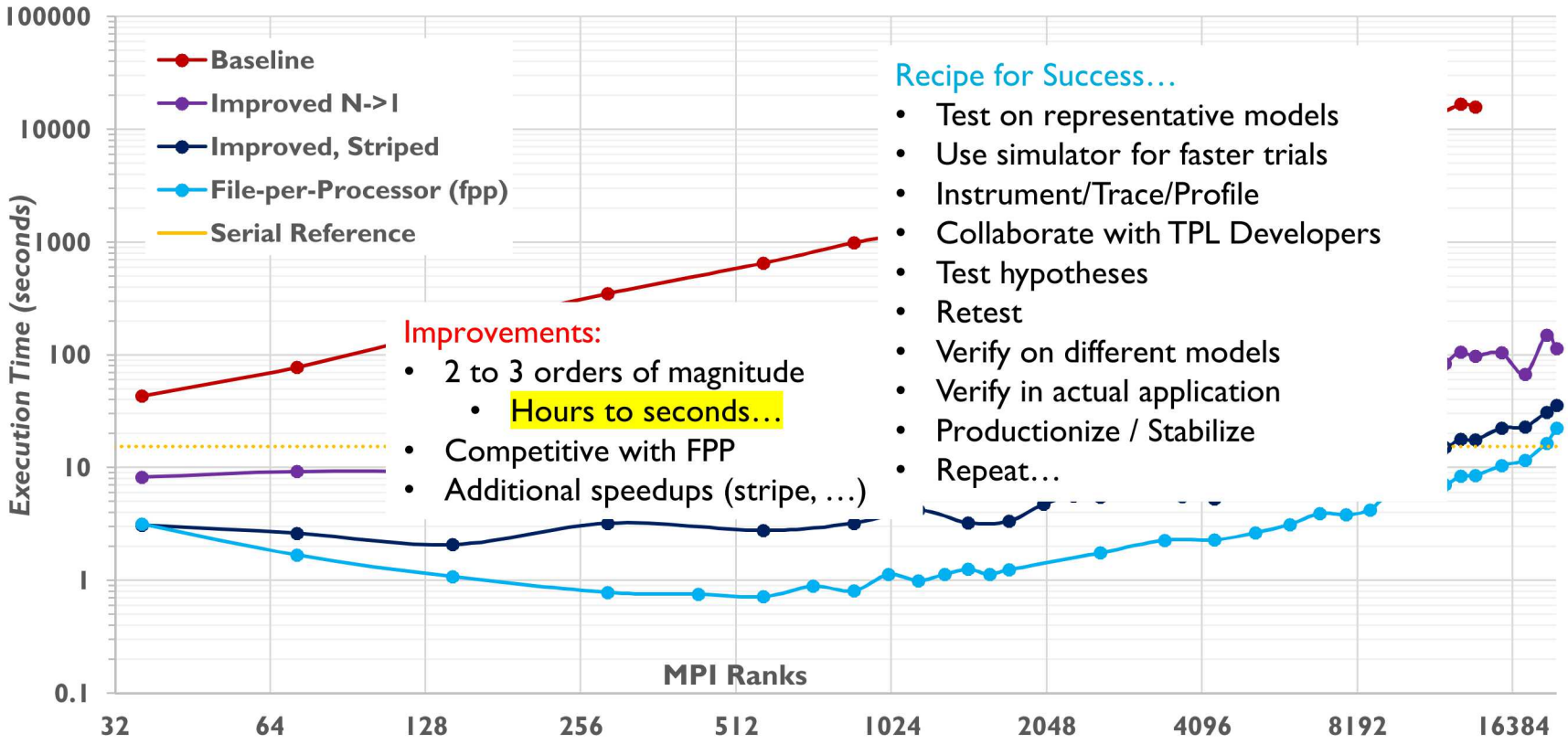


A supercomputer is a device for turning compute-bound problems into I/O-bound problems

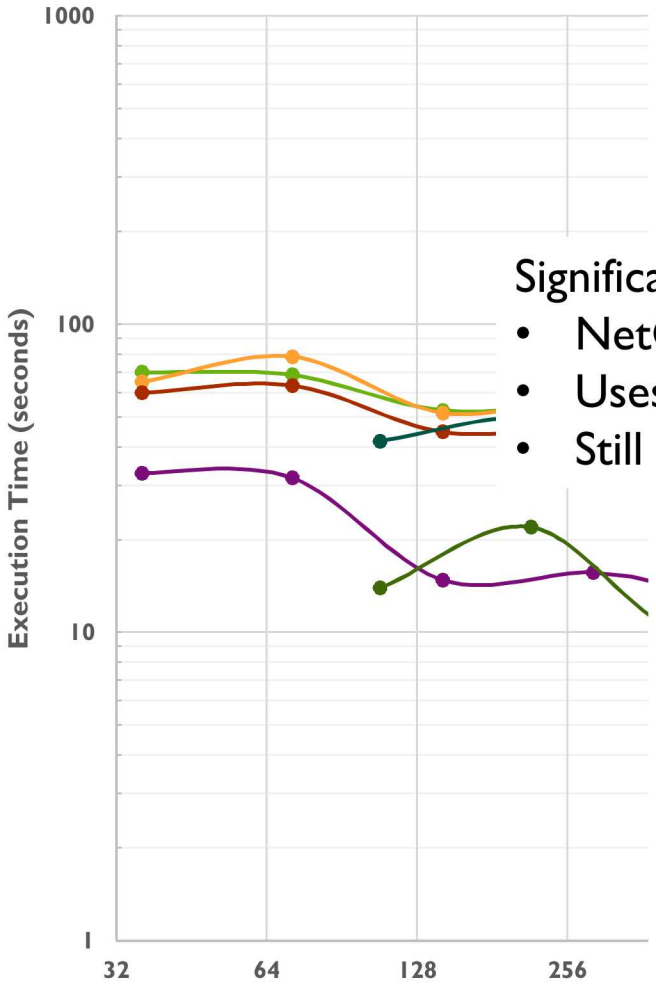
Input/Output performance improvements
CGNS Structured Mesh



Input/Output performance improvements – Large Model CGNS Structured Mesh



Exodus speedups…



Exodus format file in NetCDF-4 (HDF5) format

- 64-bit integers, 64-bit doubles
- 48.3 Gbyte file

- 390 Million finite element nodes
- 49 Million 27-node hexahedral elements
- 1 time step with 8 variables per node

Significant Speedup Realized

- NetCDF PR1570, merged for next release.
- Uses existing API function
- Still slower than FPP by >2X; more profiling needed

- Output file stripe count = 36

Run on "serrano" CTS-1 system

- 2.1 GHz processors
  - Dual sockets with 18 cores each
  - Intel Broadwell® E5-2695 v4
  - ~1.2 TFLOPs per node
- 128 GB RAM per node (3.55 GB per core)
- Intel Omni-Path high speed interconnect

Summary

Significant Improvements are possible; even in production code

Recipe for Success…

- Test on representative models
- Test at or beyond expected scale
- Use simulator for faster trials
- Instrument/Trace/Profile
- Collaborate with TPL Developers

- Test hypotheses
- Question assumptions

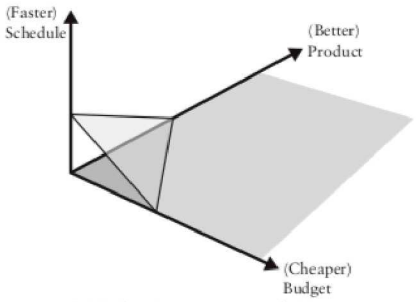
- Retest
- Verify on different models
- Verify in actual application
- Productionize / Stabilize
- Repeat…

- NO Code is sacred; examine all levels

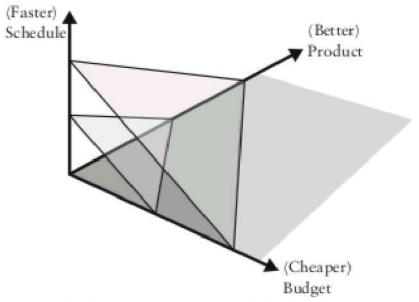
Backward Compatibility Essential

Hyrum's Law
An observation on Software Engineering
Put succinctly, the observation is this:

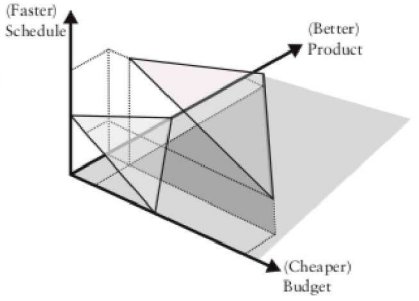
With a sufficient number of users of an API,
it does not matter what you promise in the contract,
all observable behaviors of your system
will be depended on by somebody.



(a) Original project-management decision space



(b) Quantitatively improved decision space



(c) Qualitatively improved decision space

A supercomputer is a device for turning compute-bound problems into I/O-bound problems Lakos, Large-Scale C++

16

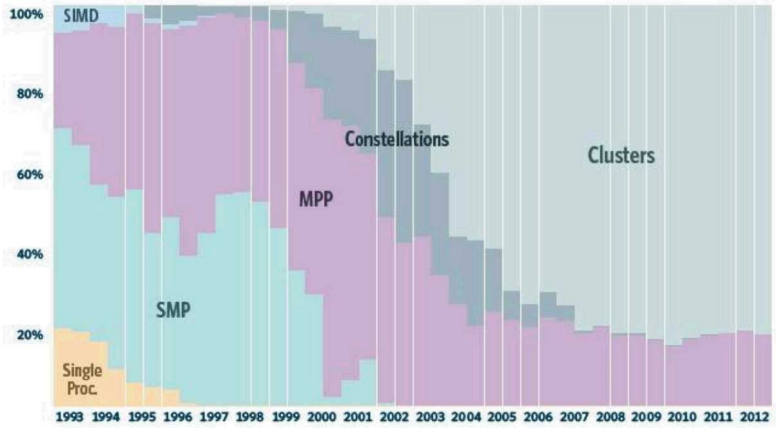
Exodus Evolution – Parallel

Original workflow -- file-per-processor [~1998]

- External tools to split to parallel and join from parallel
- Extension library "nemesis" provide parallel data structures
- Each "processor" file is valid exodus file.

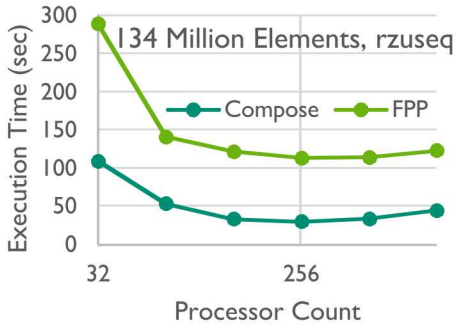
Exodus becomes "parallel-aware"

- Nemesis Embedded in Exodus (ne_? Changed to ex_?)
- PNetCDF and HDF5 provide parallel IO capabilities
- Can support auto-decomposition and auto-join
- Added several "partial" read/write functions



Parallel Scalability -- AutoJoin

Improving parallel scalability of auto-join (single file output)
- Patch NetCDF/HDF5 to avoid unneeded data access (PR 336)
- Code review of Ioss "autojoin" routines



Preliminary results look promising, more to do:
- Darshan Profiling
- MPI Profiling
- MPI-IO Tuning
- Filesystem Tuning

Working with HDF5 –
- Flush, truncate, close scalability issues