MVAPICH Still Saving the World!

Now Even Faster.

MVAPICH User's Group Meeting

August 26, 2014

Presented by Adam Moody

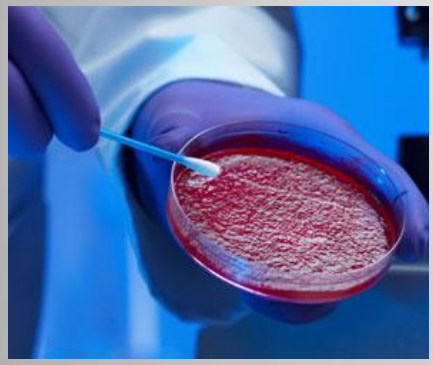
Lawrence Livermore National Laboratory

LLNL-PRES-659174

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

LLNL's mission is applying world-class science, technology, and engineering to national & global problems

Bio-Security



Counterterrorism

Defense

Energy

Intelligence

Nonproliferation

Science

Weapons

https://missions.llnl.gov

LLNL systems by purpose

Capability
Capacity
Visualization
Serial

System	Top500 Rank	Program	Manufactur e/ Model	OS	Inter-connect	Nodes	Cores	Memory (GB)	Peak TFLOP/s
Unclassified Network (OCF)									
Vulcan	9	ASC+M&IC+HPCIC	IBM BGQ	RHEL/CNK	5D Torus	24,576	393,216	393,216	5,033.2
Sierra	263	M&IC	Dell	TOSS	IB QDR	1,944	23,328	46,656	243.7
Cab (TLCC2)	94	ASC+M&IC+HPCIC	Appro	TOSS	IB QDR	1,296	20,736	41,472	426.0
Ansel		M&IC	Dell	TOSS	IB QDR	324	3,888	7,776	43.5
RZMerl (TLCC2)		ASC+ICF	Appro	TOSS	IB QDR	162	2,592	5,184	53.9
RZZeus		M&IC	Appro	TOSS	IB DDR	267	2,144	6,408	20.6
Catalyst		ASC+M&IC	Cray	TOSS	IB QDR	324	7,776	41,472	149.3
Edge		M&IC	Appro	TOSS	IB QDR	216	2,592	20,736	239.9
Aztec		M&IC	Dell	TOSS	N/A	96	1,152	4,608	12.9
Herd		M&IC	Appro	TOSS	IB DDR	9	256	1,088	1.6
OCF Totals	Systems	10							6,224.6
Classified Network (SCF)									
Pinot(TLCC2, SNSI)		M&IC	Appro	TOSS	IB QDR	162	2,592	10,368	53.9
Sequoia	3	ASC	IBM BGQ	RHEL/CNK	5D Torus	98,304	1,572,864	1,572,864	20132.7
Zin (TLCC2)	41	ASC	Appro	TOSS	IB QDR	2,916	46,656	93,312	961.1
Juno (TLCC)	399	ASC	Appro	TOSS	IB DDR	1,152	18,432	36,864	162.2
Muir		ICF	Dell	TOSS	IB QDR	1,296	15,552	31,104	168.0
Graph		ASC	Appro	TOSS	IB DDR	576	13,824	72,960	107.5
Max		ASC	Appro	TOSS	IB QDR	324	5,184	82,944	107.8
Inca		ASC	Dell	TOSS	N/A	100	1,216	5,120	13.5
SCF Totals	Systems	8							21,706.7
Combined Totals		18							27,931.3

System	Top500 Rank	Program	Manufacturer/ Model	OS	Inter-connect	Nodes	Cores	Memory (GB)	Peak TFLOP/s
Unclassified Network (OCF)									
Vulcan	9	ASC+M&IC+HPCIC	IBM BGQ	RHEL/CNK	5D Torus	24,576	393,216	393,216	5,033.2
Sierra	263	M&IC	Dell	TOSS	IB QDR	1,944	23,328	46,656	243.7
Cab (TLCC2)	94	ASC+M&IC+HPCIC	Appro	TOSS	IB QDR	1,296	20,736	41,472	426.0
Ansel		M&IC	Dell	TOSS	IB QDR	324	3,888	7,776	43.5
RZMerl (TLCC2)		ASC+ICF	Appro	TOSS	IB QDR	162	2,592	5,184	53.9
RZZeus		M&IC	Appro	TOSS	IB DDR	267	2,144	6,408	20.6
Catalyst		ASC+M&IC	Cray	TOSS	IB QDR	324	7,776	41,472	149.3
Edge		M&IC	Appro	TOSS	IB QDR	216	2,592	20,736	239.9
Aztec		M&IC	Dell	TOSS	N/A	96	1,152	4,608	12.9
Herd		M&IC	Appro	TOSS	IB DDR	9	256	1,088	1.6
OCF Totals	Systems	10							6,224.6
Classified Network (SCF)									
Pinot(TLCC2, SNSI)		M&IC	Appro	TOSS	IB QDR	162	2,592	10,368	53.9
Sequoia	3	ASC	IBM BGQ	RHEL/CNK	5D Torus	98,304	1,572,864	1,572,864	20132.7
Zin (TLCC2)	41	ASC	Appro	TOSS	IB QDR	2,916	46,656	93,312	961.1
Juno (TLCC)	399	ASC	Appro	TOSS	IB DDR	1,152	18,432	36,864	162.2
Muir		ICF	Dell	TOSS	IB QDR	1,296	15,552	31,104	168.0
Graph		ASC	Appro	TOSS	IB DDR	576	13,824	72,960	107.5
Max		ASC	Appro	TOSS	IB QDR	324	5,184	82,944	107.8
Inca		ASC	Dell	TOSS	N/A	100	1,216	5,120	13.5
SCF Totals	Systems	8							21,706.7
Combined Totals		18							27,931.3

Why MVAPICH?

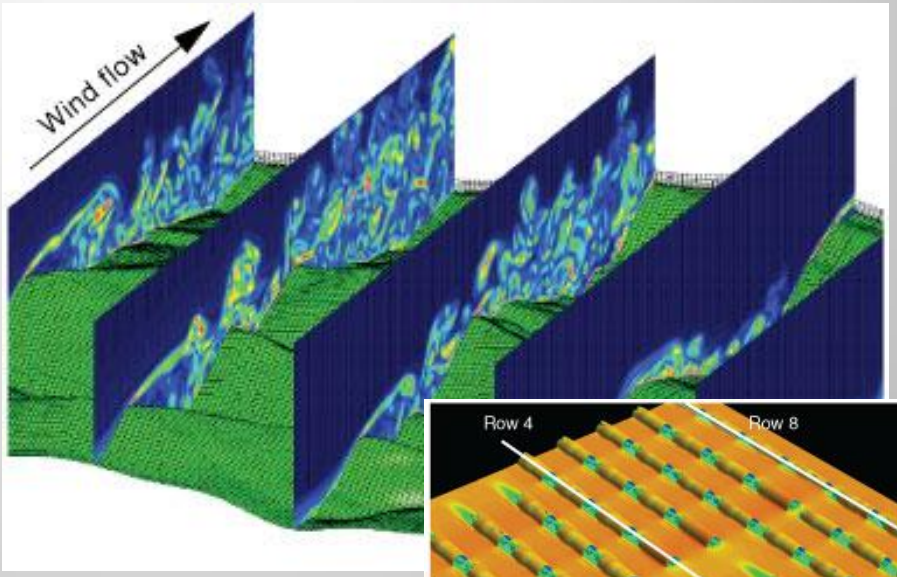
- First MPI available for IB
- Reliable and proven
- Fastest for many users
- Familiarity with MPICH code base
- Acceptance of feedback and patches
- Good ties and communication with OSU

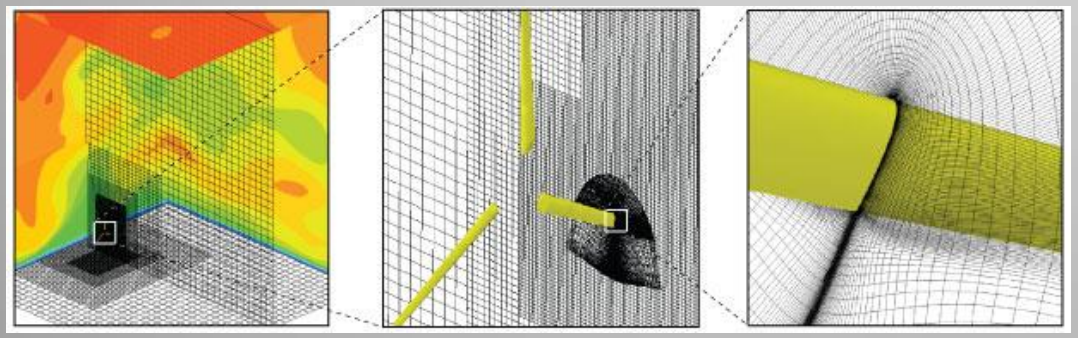
Livermore Computing / OSU: Successful history of collaboration

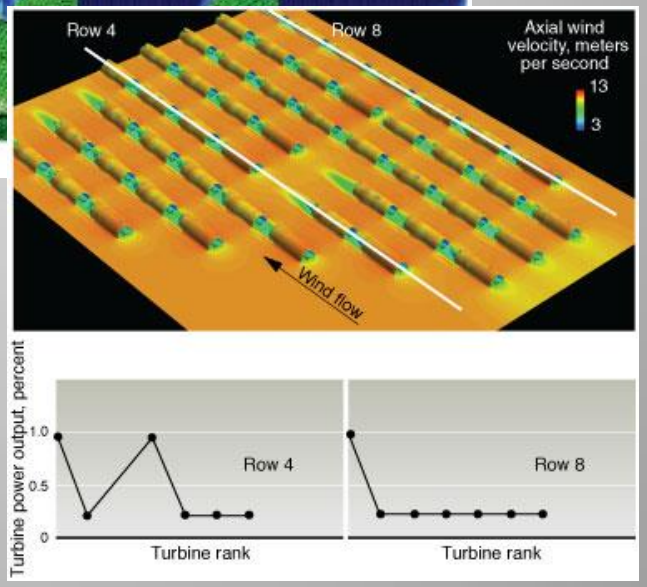
- Interns
  - Matt Koop
  - Hari Subramoni
  - Krishna Kandalla
  - Raghunath Rajachandrasekar
- Compute resources
  - “Collaborative Zone” systems
  - Hyperion




Integrated suite of codes (WRF, IBM, CGWind, HELIOS) estimate wind farm productivity and lifetime



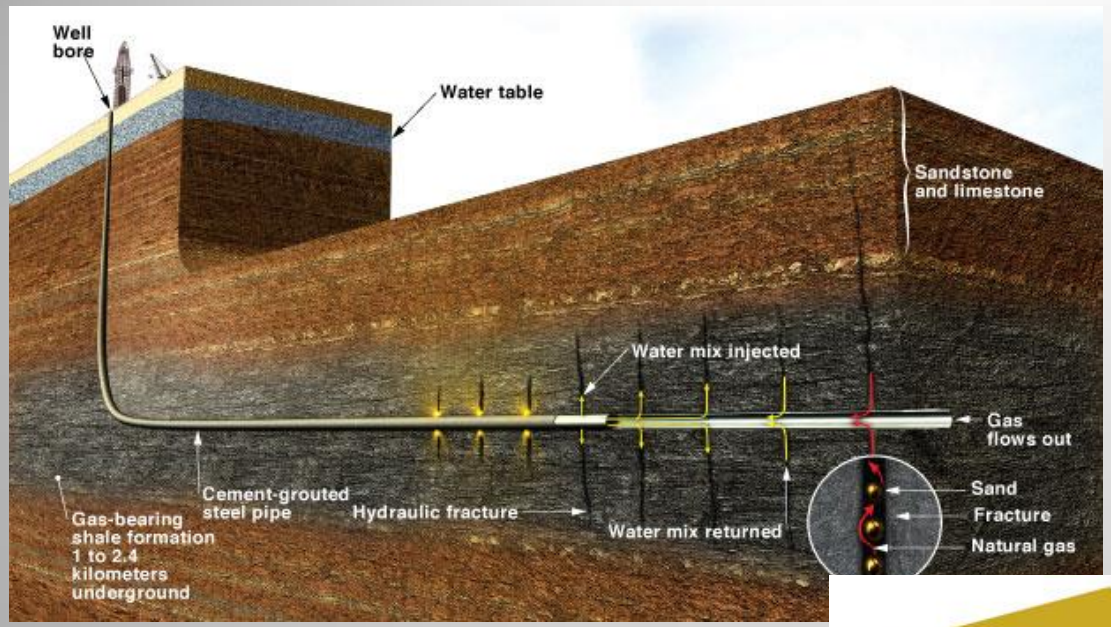


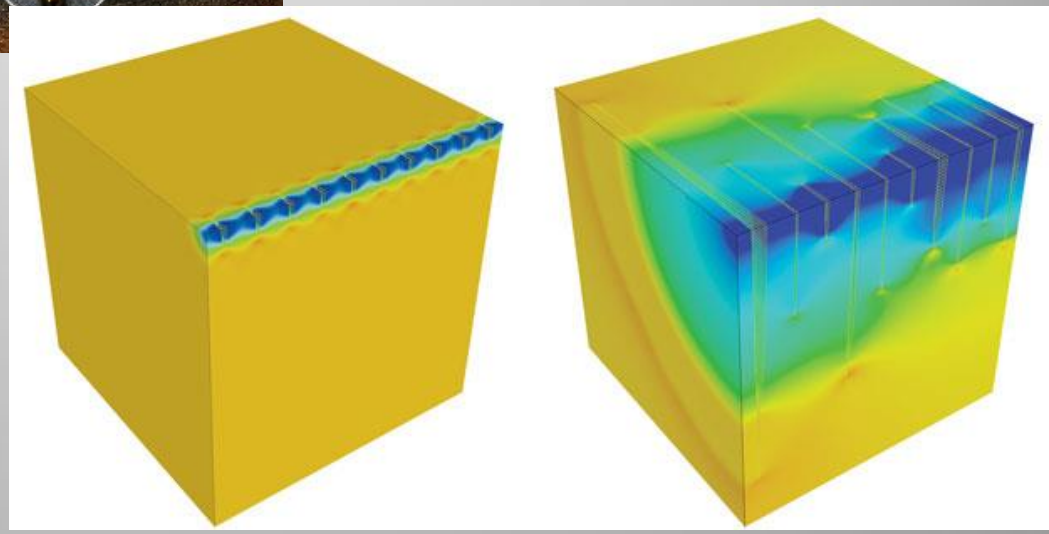


https://str.llnl.gov/april-2014/miller

Thanks: Wayne Miller, Katie Lundquist

GEOS optimizes hydraulic fracturing



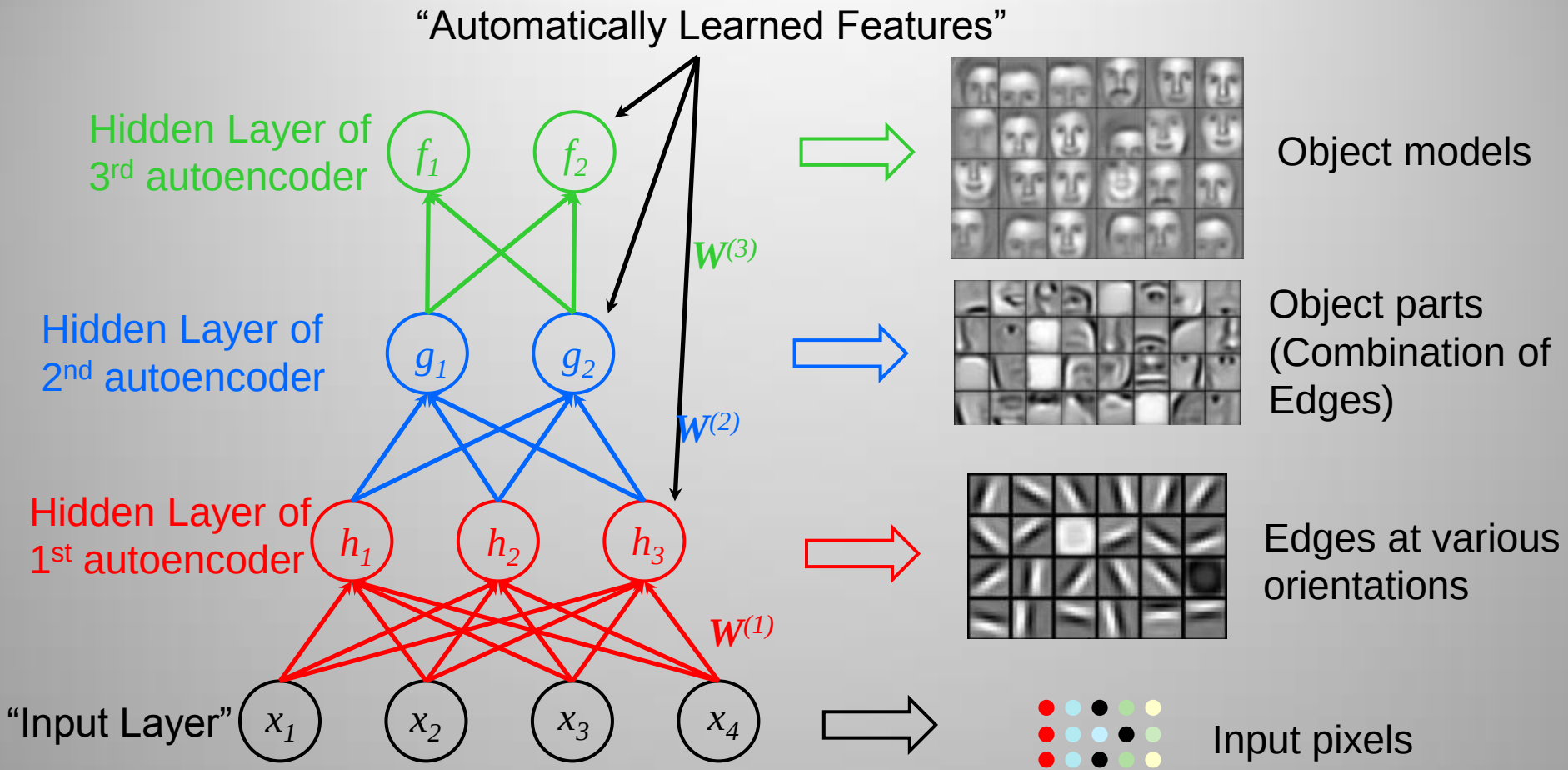


Uses a constant boundary condition at the center of each fracture and variations in pumping pressure of less than 0.5%. Running in explicit mode, the simulation used 960 processors and took 15 hours.

https://str.llnl.gov/july-2014/ryerson

Thanks: Rick Ryerson, Randy Settgast

Deep Learning Automatically Learns a Set of Hierarchical Basis Patterns ("Features")



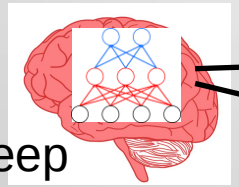
[Example from Honglak Lee]

Example: Train Deep Learning Network on images of faces

Livermore Brain: Fed by MVAPICH2 + CUDA

The overall goal of this project is to scale up Deep Learning algorithms using HPC and develop new Deep Learning approaches for finding inherent complex time-varying patterns

Unlabeled Static Inputs



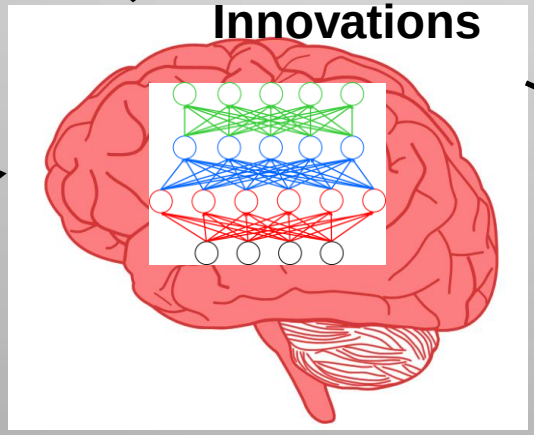
Deep Learning

Static Patterns Learned

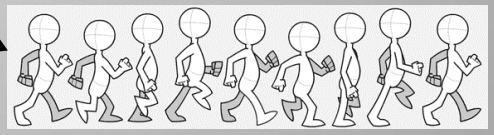
LLNL Team:
Kofi Boakye, Maya Gokhale, Brenda Ng, Karl Ni, Roger Pearce, Brian Van Essen, and Eric Wang

Stanford Collaborators:
Adam Coates, Brody Huval, and Andrew Ng

LLNL Innovations



Spatially/Time-Varying Inputs

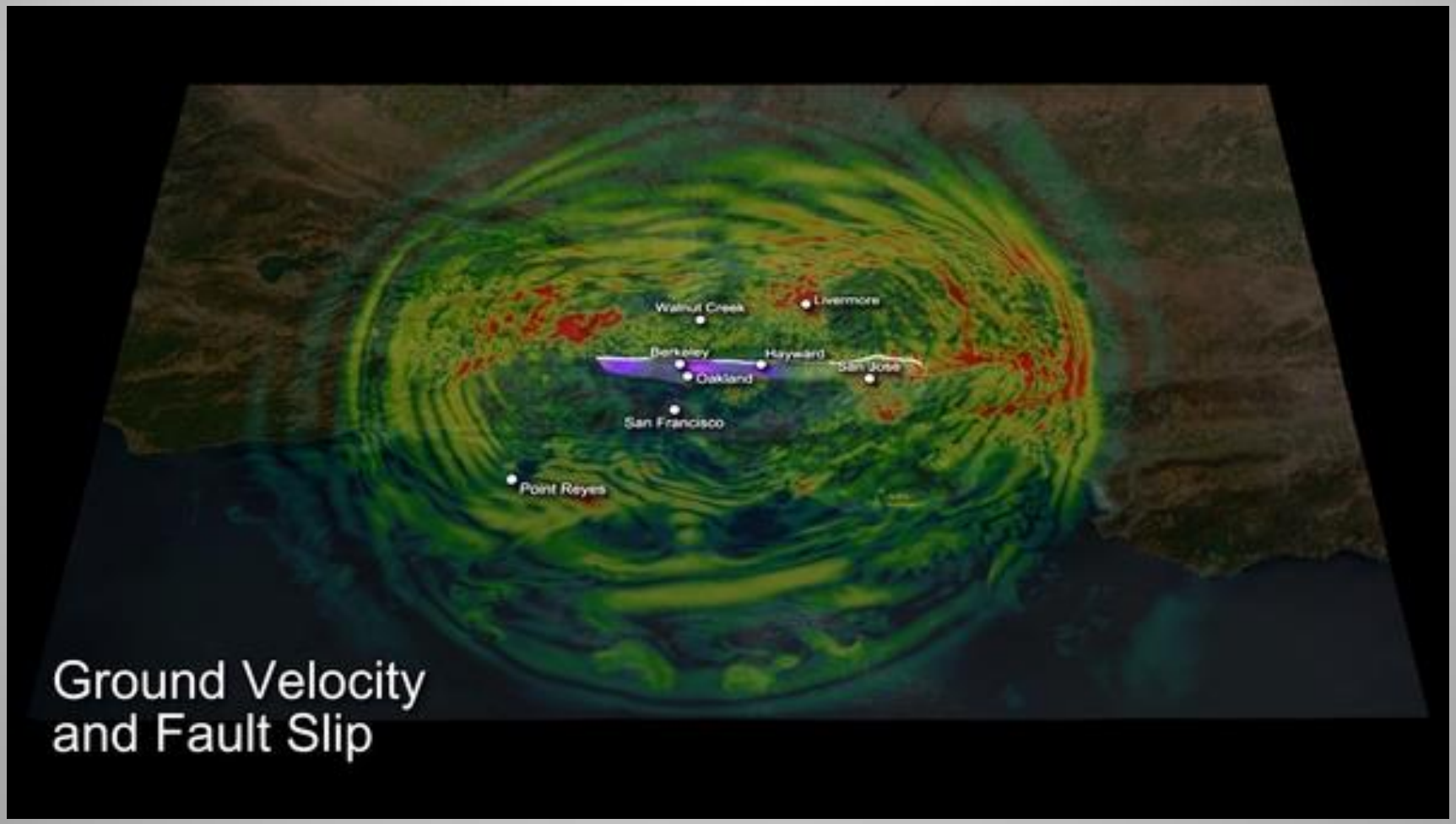


Complex Dynamic Patterns Learned

Thanks: Barry Chen, Roger Pearce

The Hayward Fault: Due for Major Earthquake

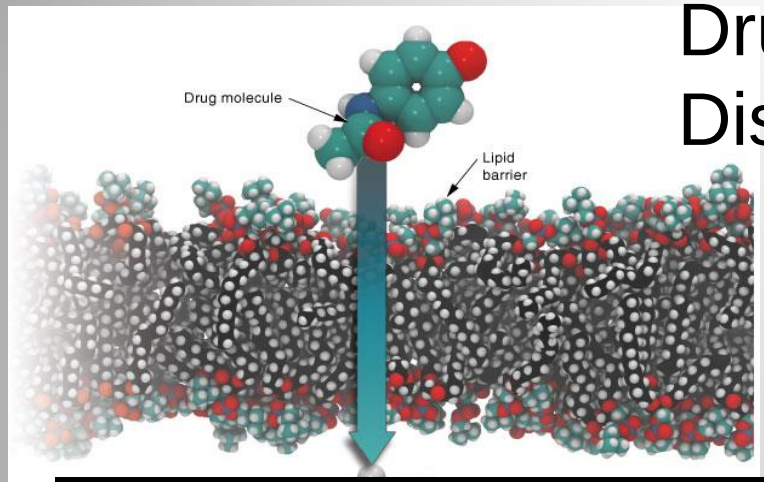
Hayward Fault Earthquake Simulation



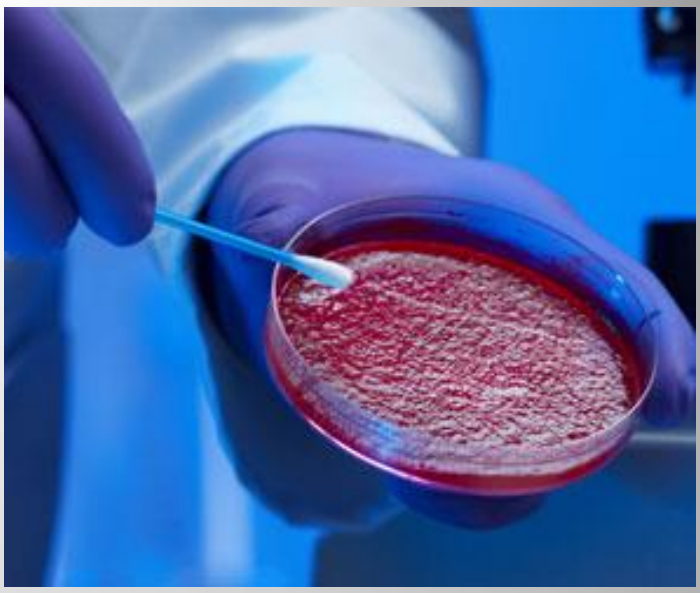
WPP open source code:
https://computation.llnl.gov/casc/serpentine/index.html

YouTube: "Supercomputing meets seismology in earthquake exhibit"

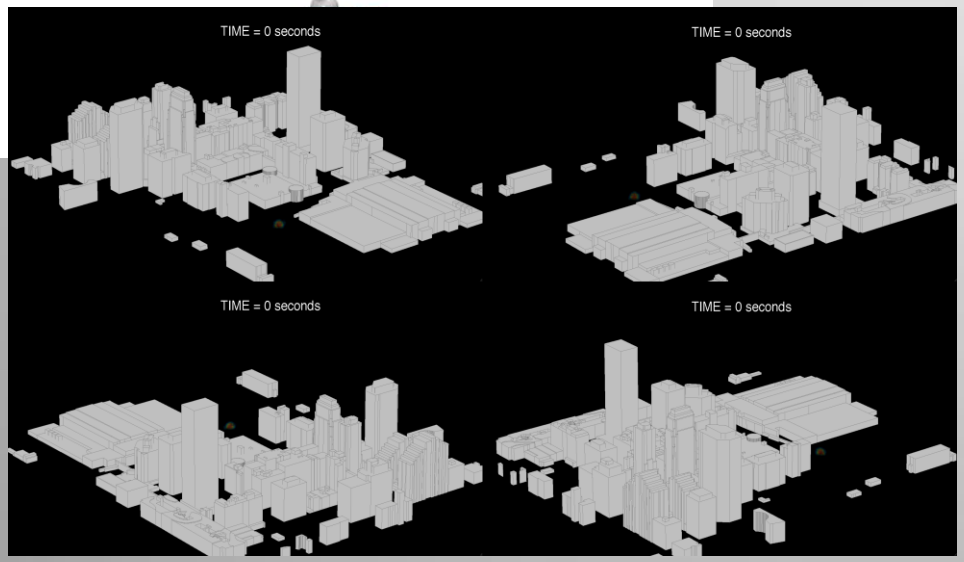
And lots more…



Drug Discovery

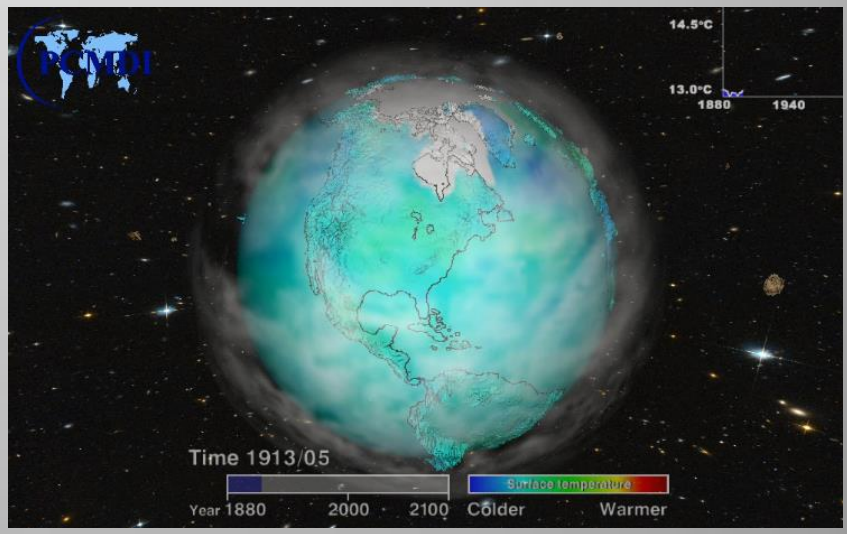


Pathogen Identification



Toxic Chemical Dispersion

Climate Change

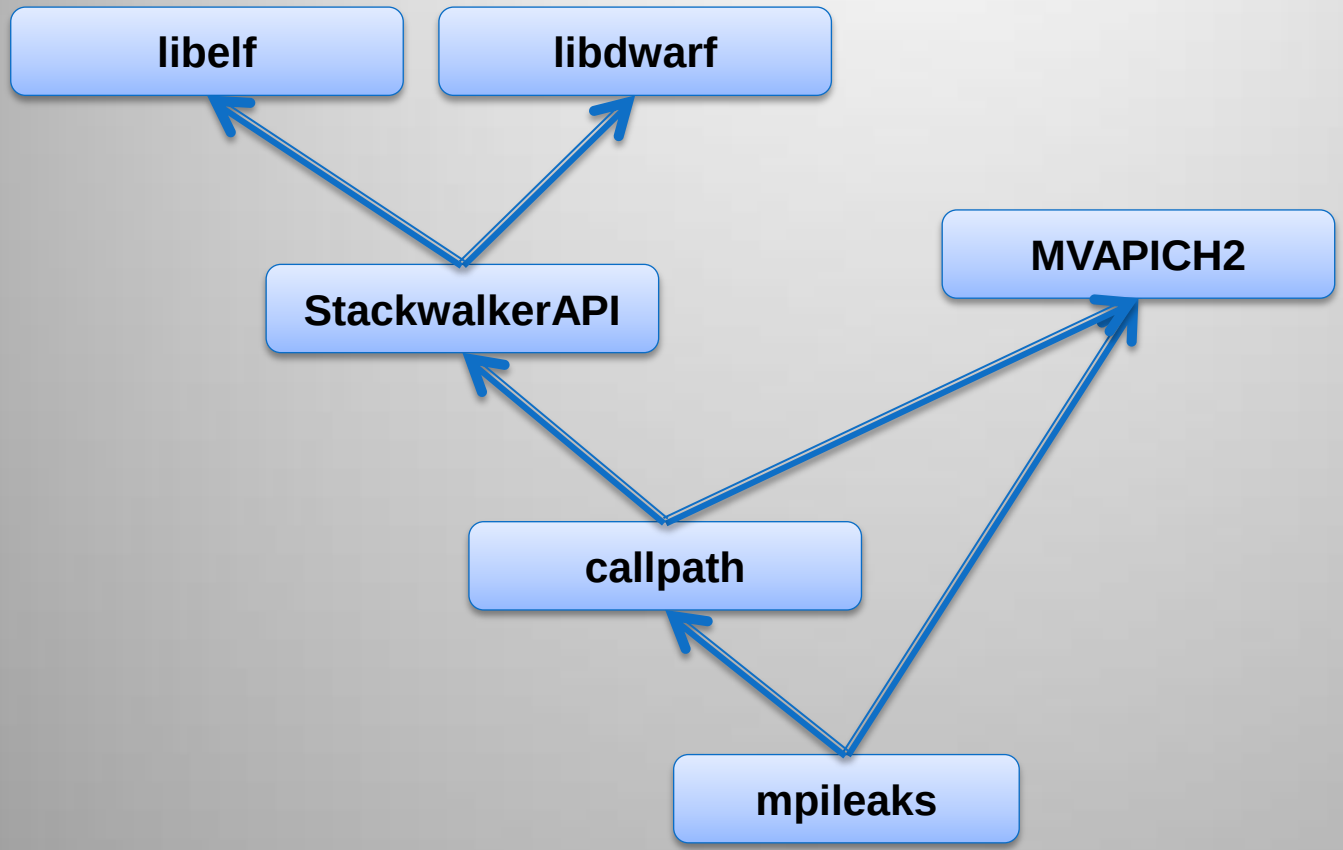


mpileaks – reports source line where code leaks MPI objects

- https://github.com/hpc/mpileaks
- MPI object handles are finite resources
  - Leaks eventually fatal
  - Hard to find source in large apps
- PMPI tool intercepts creation calls
  - MPI_Isend → request, MPI_Comm_split → comm
  - Capture callpath (stack trace)
  - Increment counter for callpath
  - Associate handle ID with callpath
  - Decrement callpath count when ID freed
- Report any callpaths with non-zero counts at MPI_Finalize
  - Sum across procs for scalability

```
>>: mpicc -g -O0 -o tests tests.c
>>: srun-mpileaks -n2 ./tests
---------------------------------------
START SECTION: LEAKED OBJECTS
---------------------------------------
Count: 9  ::  create_2level_comm.c:191:…
Count: 9  ::  create_2level_comm.c:266:…
Count: 7  ::  create_2level_comm.c:181:…
Count: 2  ::  tests.c:57:sendrecv
Count: 2  ::  tests.c:84:fileio
Count: 2  ::  tests.c:99:datatypes
Count: 2  ::  tests.c:120:errhandlers
Count: 2  ::  tests.c:131:keyvals
Count: 2  ::  tests.c:173:ops
Count: 2  ::  tests.c:299:comms
Count: 1  ::  tests.c:27:persistent
---------------------------------------
END SECTION: LEAKED OBJECTS
---------------------------------------
---------------------------------------
START SECTION: POSSIBLY LEAKED OBJECTS
---------------------------------------
Count: 1  ::  tests.c:26:persistent
Count: 1  ::  tests.c:27:persistent
---------------------------------------
END SECTION: POSSIBLY LEAKED OBJECTS
---------------------------------------
```

Spack (Supercomputing Package Manager) Automates download, build, and install of dependencies for HPC software

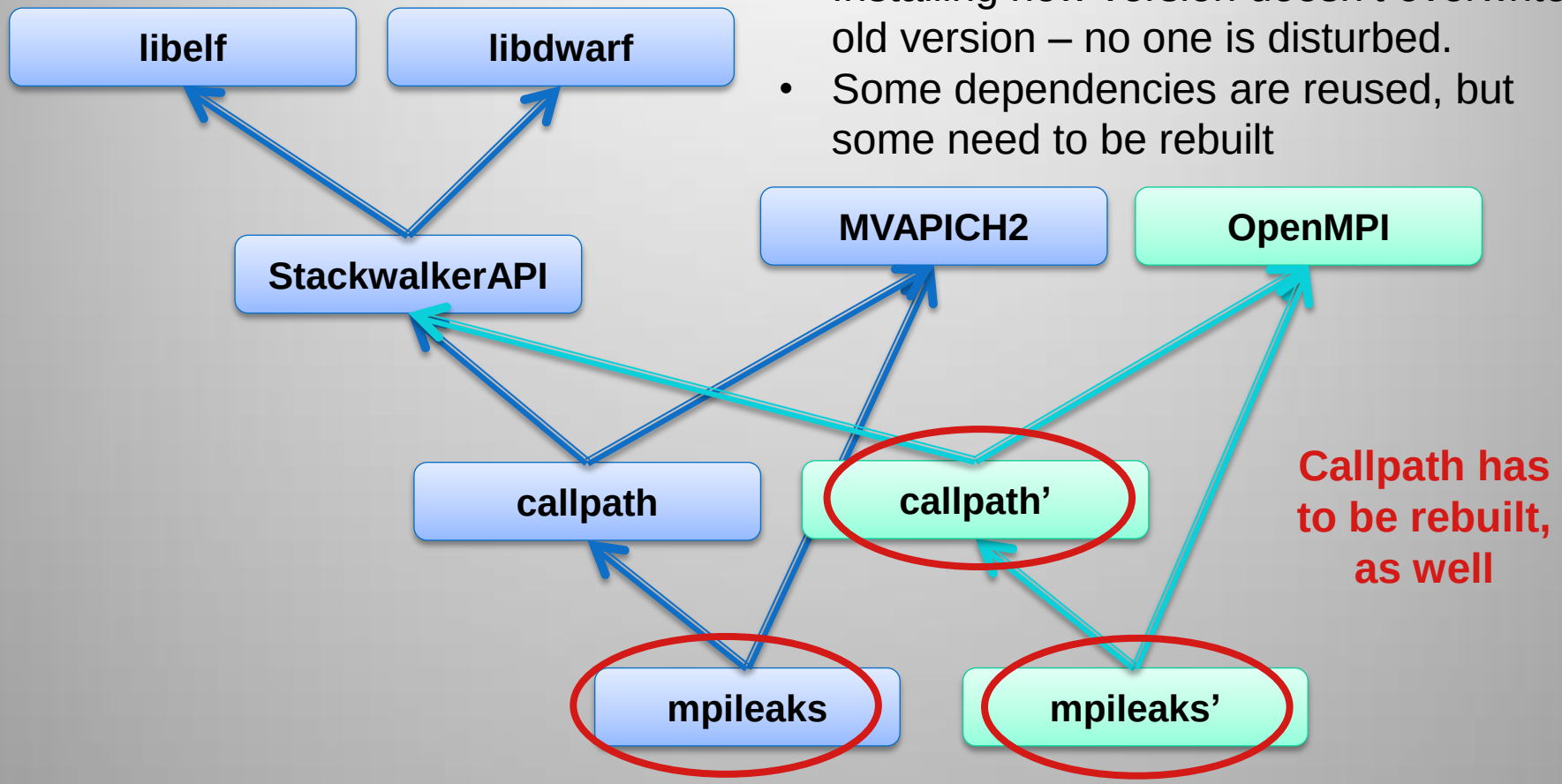


https://github.com/scalability-llnl/spack

```
(gluon):~$ spack install mpileaks
```

Adding a new mpileaks that depends on OpenMPI

- Installing new version doesn't overwrite old version – no one is disturbed.
- Some dependencies are reused, but some need to be rebuilt

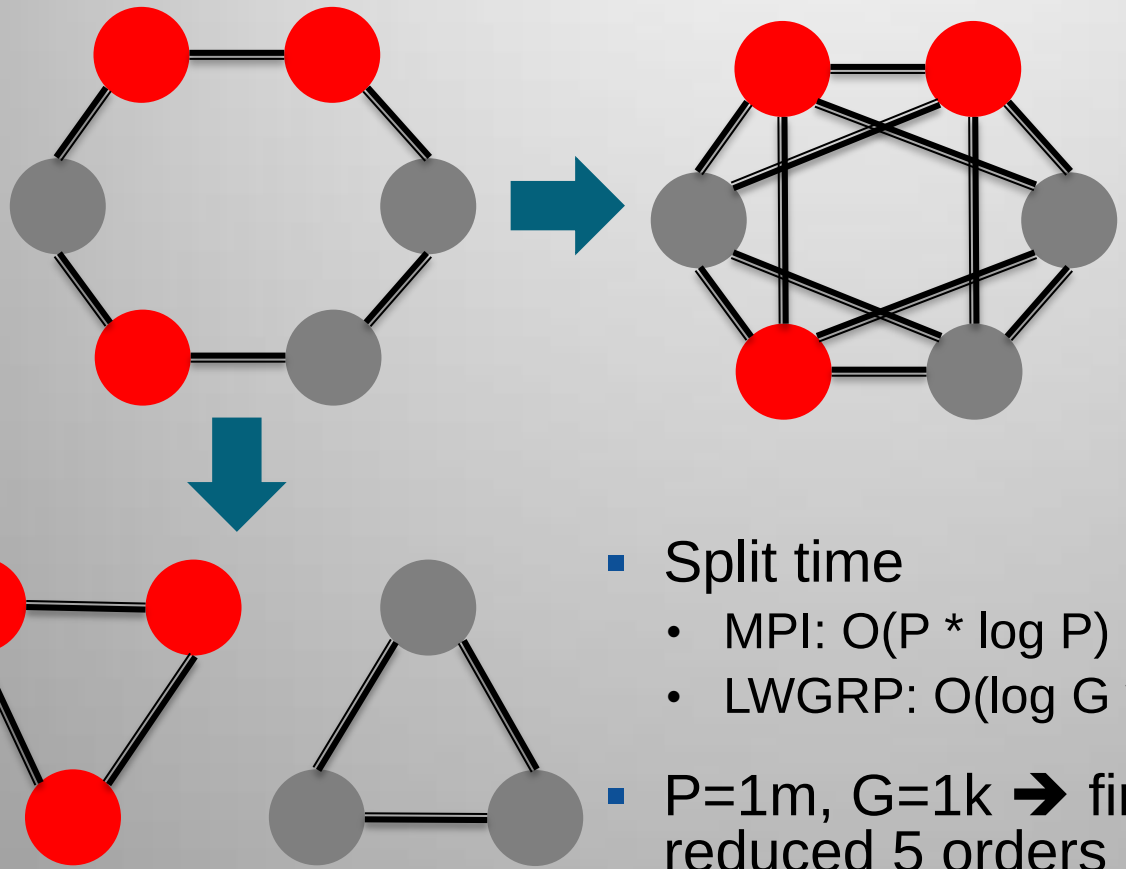


But now just LD_PRELOADING these will work as expected

```
(gluon):~$ spack install mpileaks ^openmpi
```

LWGRP (Light-weight group)
Create & destroy process groups faster than MPI communicators

https://github.com/hpc/lwgrp



- Group representation
  - MPI: O(P)
  - LWGRP: O(log P)
- P=1m ➔ memory reduced 5 orders of magnitude
- log(P) collectives: Barrier, Bcast, Allreduce, Scan, Allgather, Alltoall, etc.

- Split time
  - MPI: O(P * log P)
  - LWGRP: O(log G * log P)
- P=1m, G=1k ➔ first term reduced 5 orders of magnitude

DTCMP (Datatype comparison) User-defined comparison ops on MPI datatypes, including sort and segmented scan

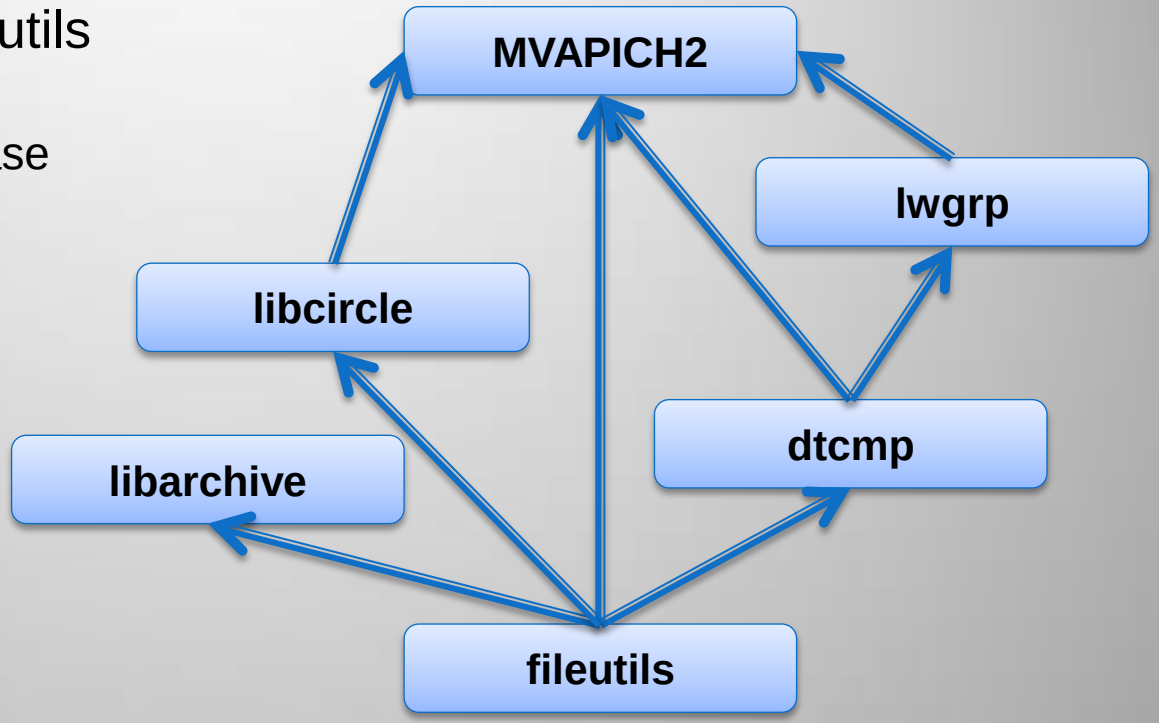
- https://github.com/hpc/dtcmp
- Collective on distributed data
  - Search, Partition, Merge, Select, Sort, Rank, Segmented scan
  - Different algorithm selected depending on data size and number of procs
- User-defnied and pre-defined ops
  - DTCMP_OP_INT_ASCEND, DTCMP_OP_FLOAT_DESCEND
- Supports
  - Arbitrary datatypes (almost)
  - Lexicographic comparisons
  - Satellite data

```
int inbuf[1000] = {... 1000 random ints ...};
int outbuf[1000];


DTCMP_Sort(
   inbuf, outbuf, 1000, MPI_INT, MPI_INT,
   DTCMP_OP_INT_ASCEND, DTCMP_FLAG_NONE,
   MPI_COMM_WORLD
);
```

Distributed FileUtils: MPI-based tools for managing large files / large sets of files

- https://github.com/hpc/fileutils
  - Multi-site collaboration
  - Private repo, awaiting release
- Suite of tools based on common library:
  - dwalk – parallel list
  - dcp – parallel copy
  - drm – parallel remove
  - dcmp – parallel compare
  - dsync – parallel rsync
  - dtar – parallel tar
  - dsh – parallel shell



Lustre directory tree w/ 400k small files

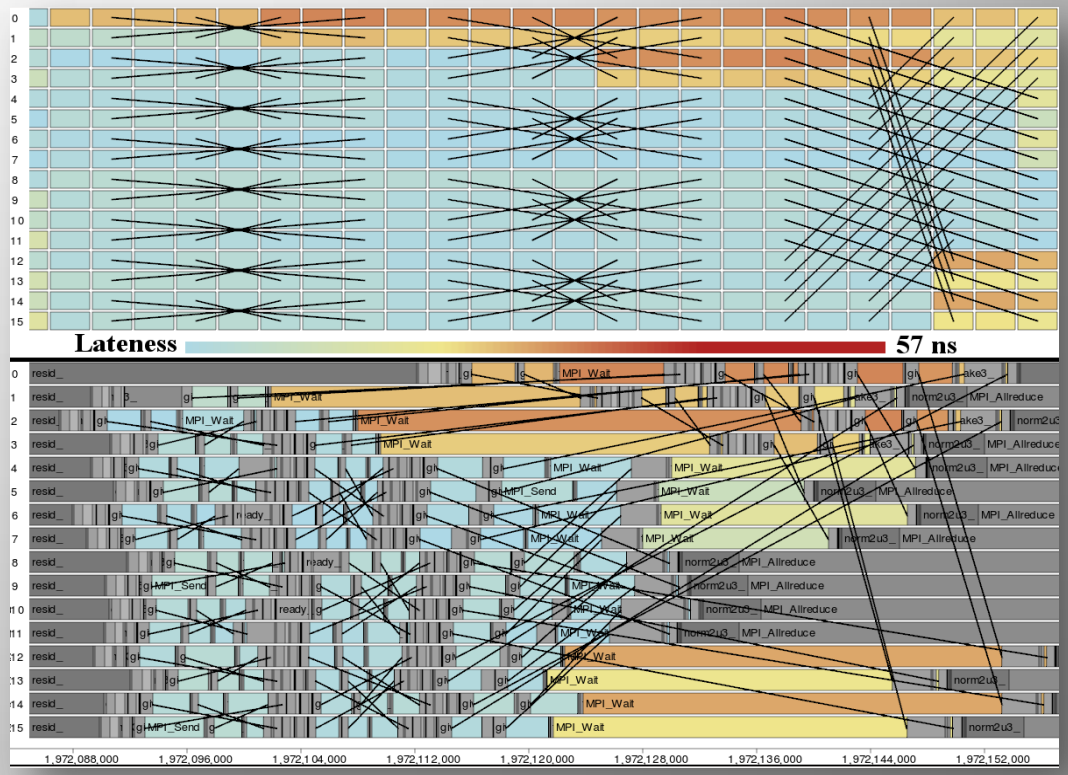
cp -pr
5 hours

dcp
16 tasks: 46 min
32 tasks: 34 min
64 tasks: 16 min
128 tasks: 14 min
256 tasks: 10 min
512 tasks: 7 min

drm
16 tasks: 10:44
32 tasks: 7:04
64 tasks: 4:41
128 tasks: 3:41
256 tasks: 2:51
512 tasks: 2:28

Ravel: Visualize Using Virtual Time

- Diagram with logical time instead of physical time
- Color logical time by differences in physical time
- NAS MG Benchmark on 16 processes



"Combing the Communication Hairball: Visualizing Parallel Execution Traces using Logical Time"

Katherine Isaacs, Peer-Timo Bremer, Ilir Jusufi, Todd Gamblin, Abhinav Bhatele, Martin Schulz, Bernd Hamann LLNL & UC Davis

mpicc complicates build & run

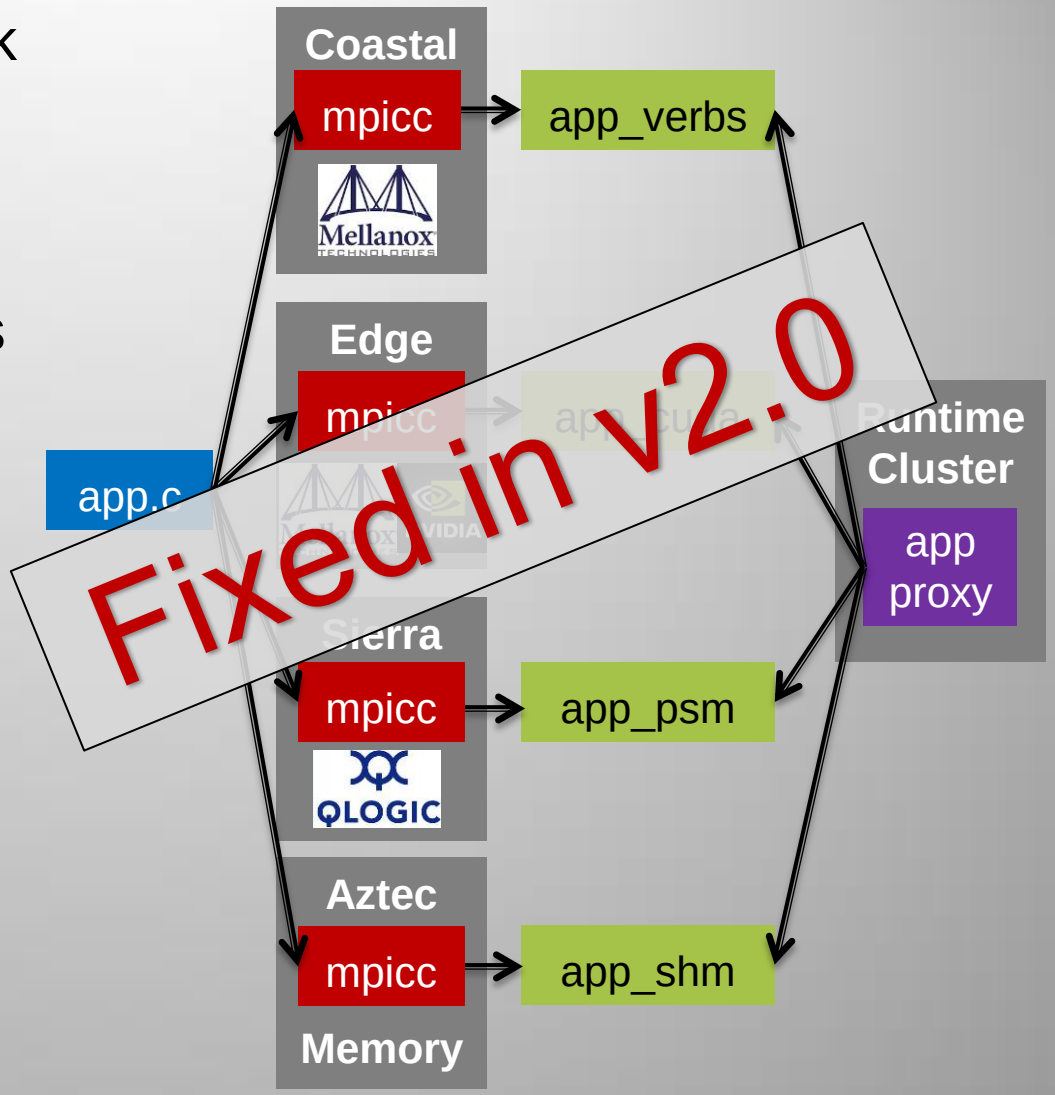
- MPI wrappers link network libs

  >>: mpicc -show

  gcc … -lmpich ... -libverbs

- LLNL has 4 network types
  - 4 different application binaries
  - 4 versions of every lib the app uses
  - Built on 4 different clusters
  - Documentation and/or scripting so user gets the right binary

- Cumbersome and error prone for app developers and their users



mpicc complicates build & run

- MPI wrappers link network libs

  >>: mpicc -show

  gcc … -lmpich ... -libverbs

- LLNL has 4 network types
  - 4 different application binaries
  - 4 versions of every lib the app uses
  - Built on 4 different clusters
  - Documentation and/or scripting so user gets the right binary
- Cumbersome and error prone for app developers and their users

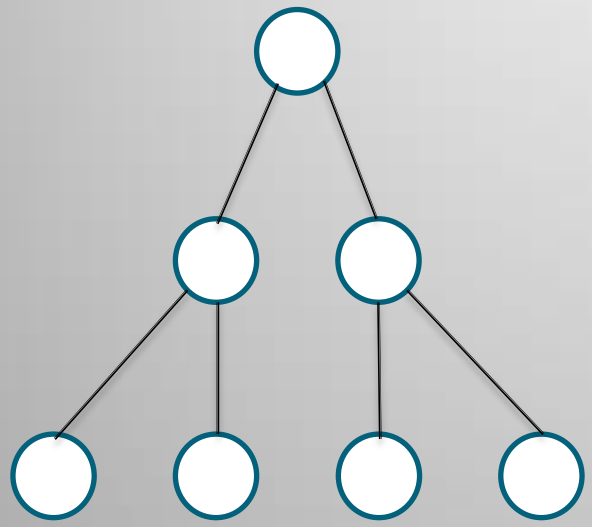
Building MPI: A Nightmare of Permutations

- Multiple compilers
  - GNU, Intel, PGI
  - several versions of each
- Multiple MPI implementations
  - MVAPICH, MVAPICH2, Open MPI
  - 2-3 versions of each
  - normal + debug
- Multiple system types

MPI	Open MPI	MVAPICH2
Compilers	3	3
x MPI Versions	3	3
x (Normal + Debug)	2	2
x Platforms	1	4
= Total	18	72 !!!

Avalaunch and MVAPICH2 Startup

- Avalaunch process tree
  - One per node
  - Children connect back to parent, get info, start their own children



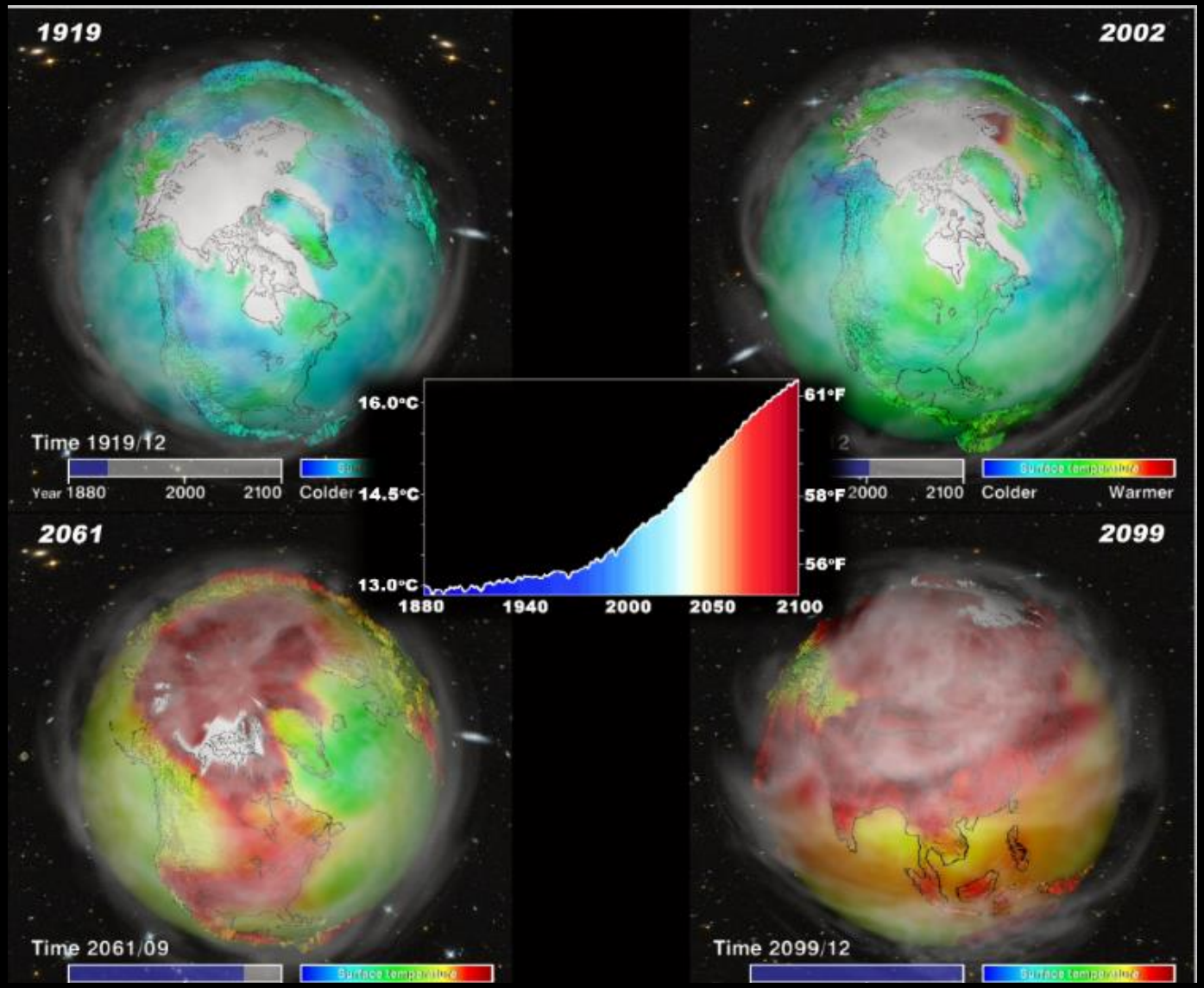
Procs	504	1000	2000	3920
Ava+ Ring	0.52	0.85	1.01	1.26
Ring	0.22	0.29	0.40	0.43

- Avalaunch experiments
  - degree of tree
  - spawn_net: TCP vs IBUD
  - path search on root
  - copy avalaunch binary from local storage of parent to local storage on child, before starting via ssh
  - bcast application and its libs from root to local storage on all nodes (SPINDLE-lite)
  - PMI_Ring vs PMI allgather
- MVAPICH2 experiments
  - spawn_net IBUD + lwgrp collectives vs PMI
  - CM address request packets for on-demand address lookup vs PMI
- Modified SLURM to support new PMI_Ring
  - “PMI Extensions for Scalable MPI Startup”, Sourav Chakraborty, Hari Subramoni, Jonathan Perkins, Adam Moody, Mark Arnold, DK Panda, EuroMPI 2014.

More Challenges

- ABI compatibility between versions
  - LLNL codes take 1 day to recompile
  - Breaks: MV1-1.2 → MV2-1.7 → MV2-1.9 → MV2-2.1
  - Example solution: MPI-Adapter (Japan) or MorphMPI
- Faster MPI_Comm_split
  - see LWGRP
- Find and eliminate every O(P) scaling term!
  - P > 1m too costly
  - Probably worthy of a PhD (or 2)
- Efficient MPI_THREAD_MULTIPLE
  - MPI Endpoints?

LLNL PCMDI IPCC



23 models, 1880-2099

MVAPICH Still Saving the World!
Now even faster.