Getting Up to Speed with DOE's Exascale Computing Effort(s)

K. J. Roche
High Performance Computing Group, Pacific Northwest National Laboratory
Nuclear Theory Group, Department of Physics, University of Washington

Credits: Dan Hitchcock, Barb Helland, Lucy Nowell, Thuc Hoang, David Turek, Ulrich Drepper, Steve Koonin, OLCF Staff, NERSC staff

Outline:

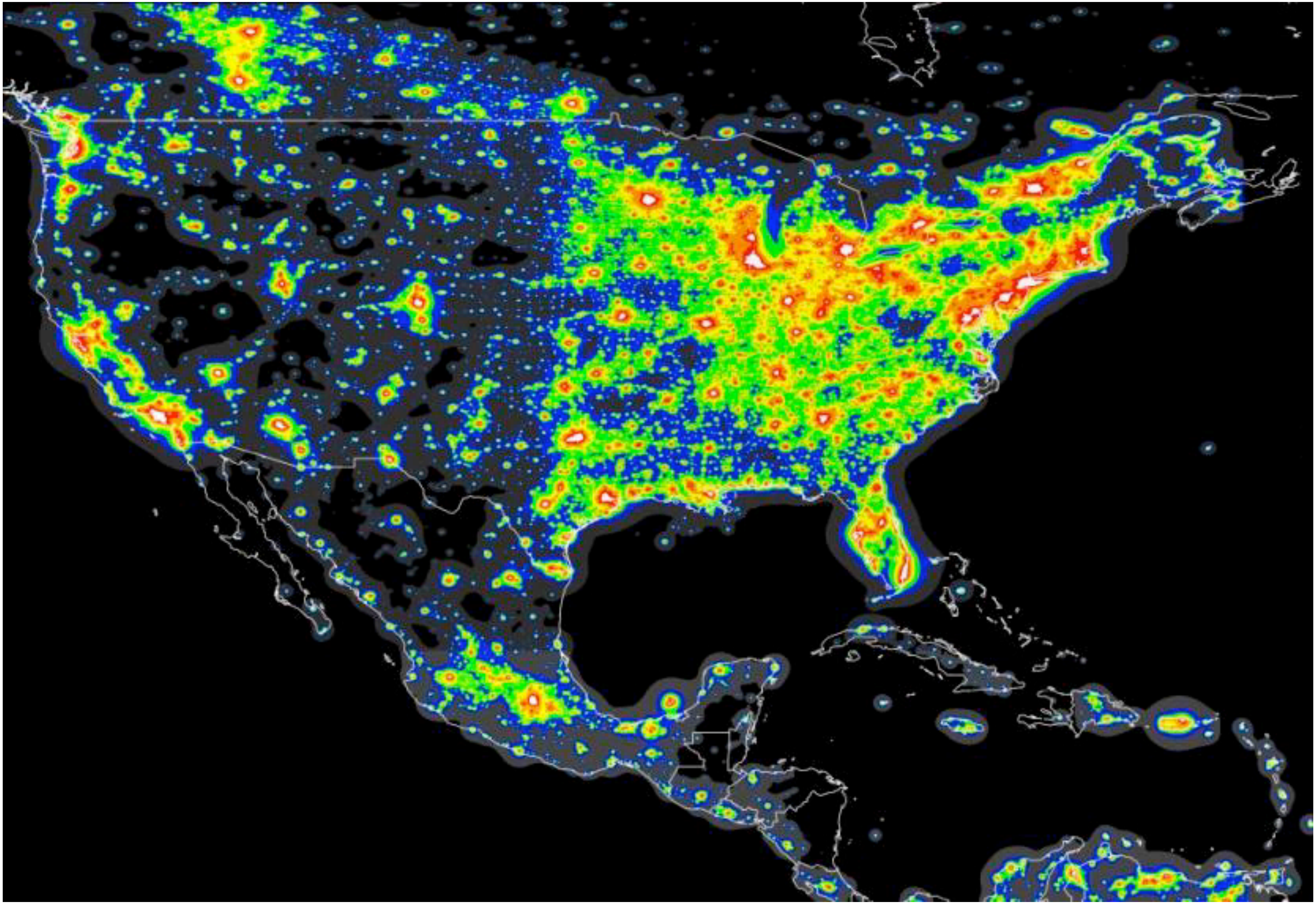
- ASCR's OMB PART PMM software metric
  - where we are NOW
- Some comments on exascale developments
- Questions / Discussion

**the contents of this talk reflect my opinions -not cleared for public consumption by DOE

RADAR

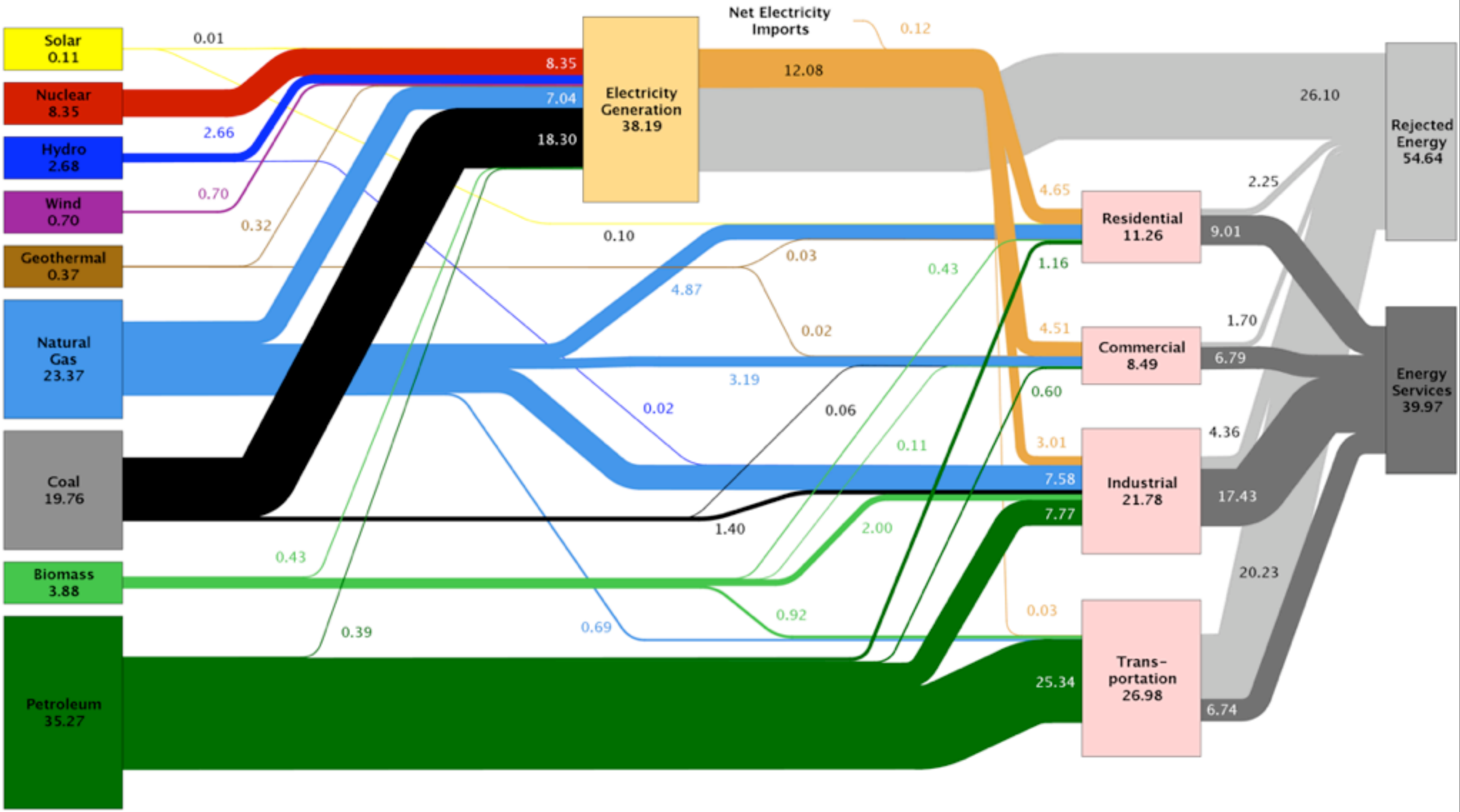
⟵ go this way ⟶

energy use spatial distribution ~ population density distribution



Lawrence Livermore National Laboratory

Estimated U.S. Energy Use in 2009: ~94.6 Quads



Source: LLNL 2010. Data is based on DOE/EIA-0384(2009), August 2010. If this information or a reproduction of it is used, credit must be given to the Lawrence Livermore National Laboratory and the Department of Energy, under whose auspices the work was performed. Distributed electricity represents only retail electricity sales and does not include self-generation. EIA reports flows for non-thermal resources (i.e., hydro, wind and solar) in BTU-equivalent values by assuming a typical fossil fuel plant "heat rate." The efficiency of electricity production is calculated as the total retail electricity delivered divided by the primary energy input into electricity generation. End use efficiency is estimated as 80% for the residential, commercial and industrial sectors, and as 25% for the transportation sector. Totals may not equal sum of components due to independent rounding. LLNL-MI-410527

Clean Energy and Related Research

- **Materials by design** using nanoscale structures and syntheses for: carbon capture; radiation-resistant and self-healing materials for the nuclear reactor industry; highly efficient photovoltaics; and white-light emitting LEDs.
- **Biosystems by design** combining the development of new molecular toolkits with testbeds for the design and construction of improved biological components or new bio-hybrid systems and processes for improved biofuels and bioproducts.
- **Modeling and simulation** to facilitate materials and chemistry by design and to address technology challenges such as the optimization of internal combustion engines using advanced transportation fuels (biofuels).

•***Climate Change*: Understanding and mitigating the effects of global warming**

–Sea level rise

–Severe weather

–Regional climate change

–Geologic carbon sequestration

•***National Nuclear Security*: Maintaining a safe, secure and reliable nuclear stockpile**

–Stockpile certification

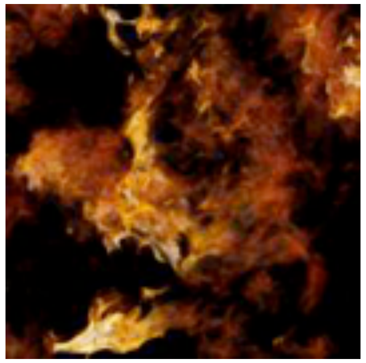
–Predictive scientific challenges

–Real-time evaluation of urban nuclear detonation

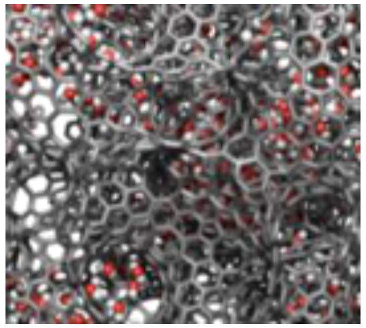
•***Energy*: Reducing U.S. reliance on foreign energy sources and reducing the carbon footprint of energy production**

–Reducing time and cost of reactor design and deployment

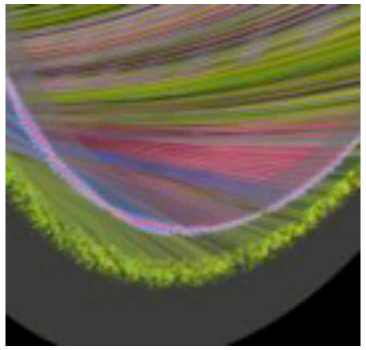
–Improving the efficiency of combustion energy sources



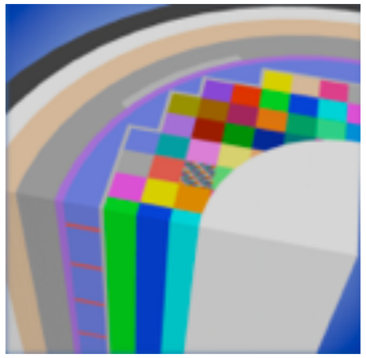
Turbulence
Understanding the statistical geometry of turbulent dispersion of pollutants in the environment.



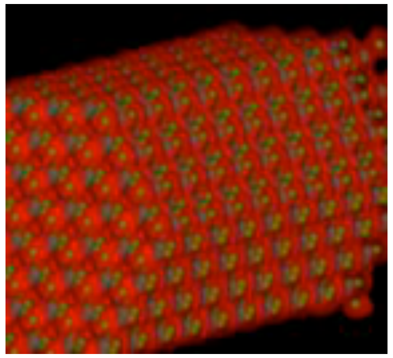
Energy Storage
Understanding the storage and flow of energy in next-generation nanostructured carbon nanotube supercapacitors



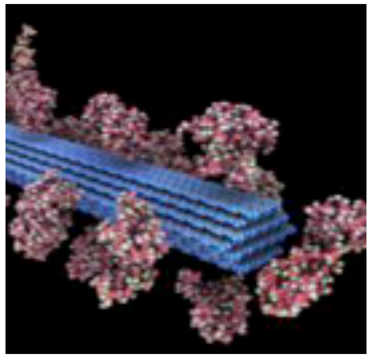
Fusion Energy
Substantial progress in the understanding of anomalous electron energy loss in the National Spherical Torus Experiment (NSTX).



Nuclear Energy
High-fidelity predictive simulation tools for the design of next-generation nuclear reactors to safely increase operating margins.

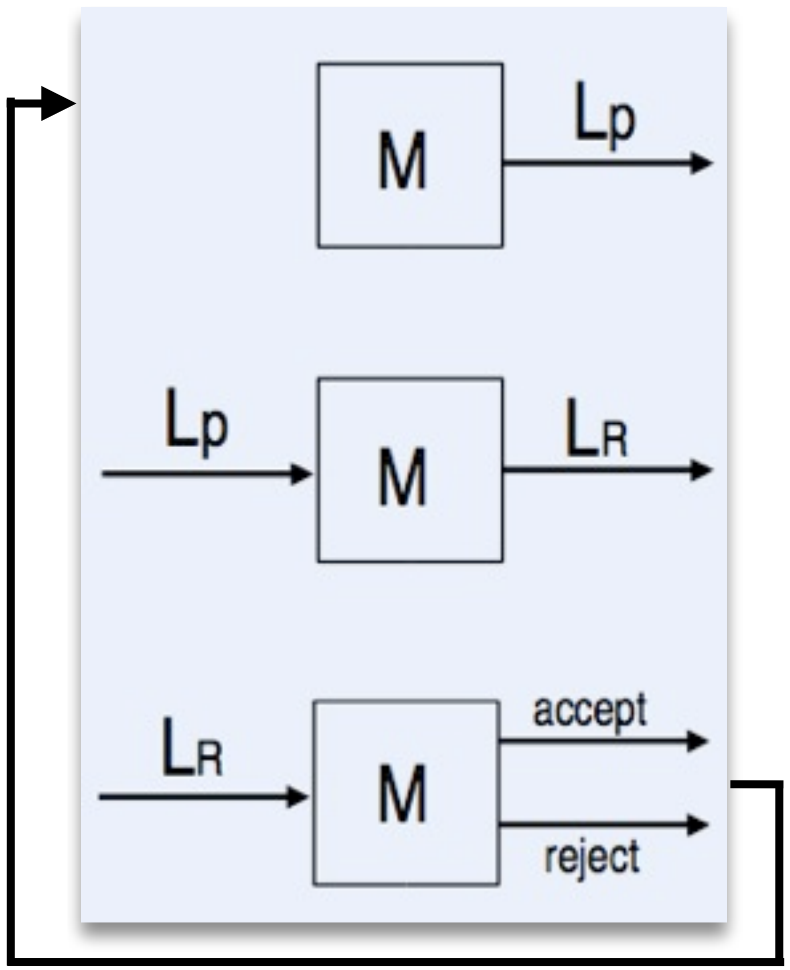


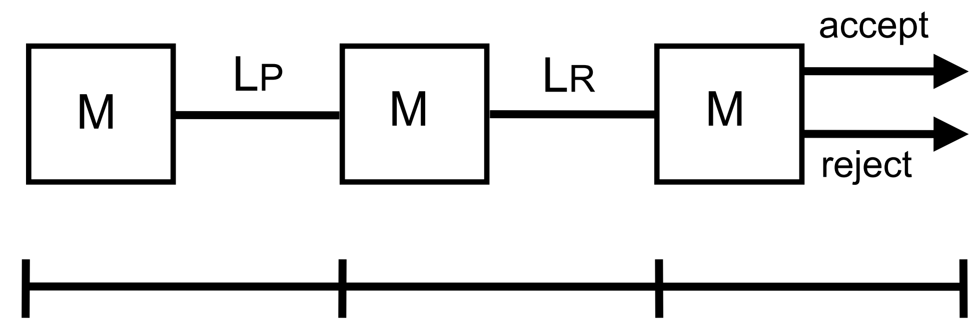
NanoScience
Understanding the atomic and electronic properties of nanostructures in next-generation photovoltaic solar cell materials.



Biofuels
A comprehensive simulation model of lignocellulosic biomass to understand the bottleneck to sustainable and economical ethanol production.

- COMPLEXITY
  - PROBLEMS
  - ALGORITHMS
  - MACHINES





Measured time for machine M to generate the language of the problem plus time to generate the language of the result plus the time to accept or reject the language of the result.

Asking questions, solving problems is recursive process

Accepting a result means a related set of conditions is satisfied

$$S = S1 \wedge S2 \wedge \ldots \wedge Sn$$

How Are Mission Applications Performing on Today's Systems

US OMB PART DOE SC ASCR Annual
GPRA / PMM Goal with Quarterly Updates

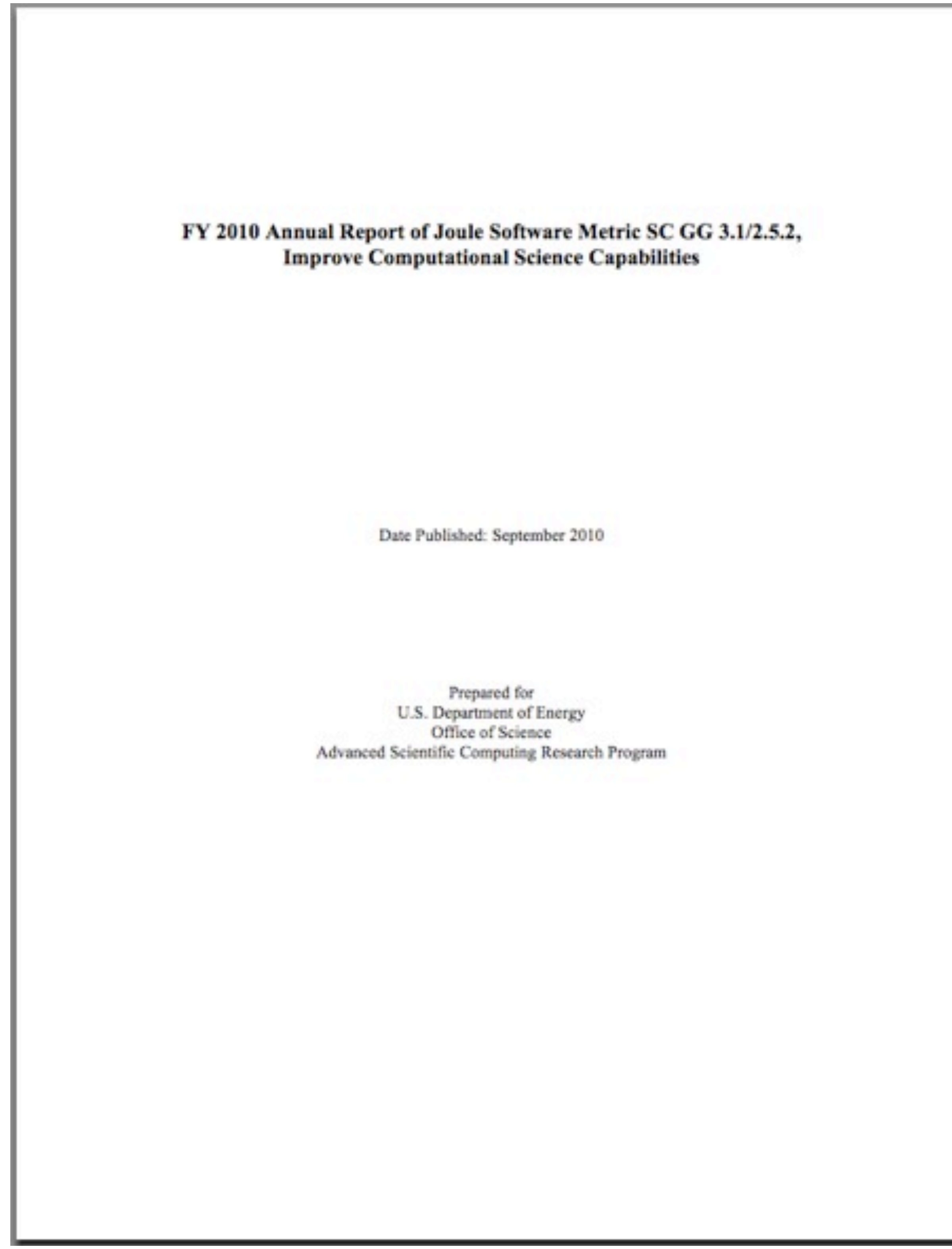
(SC GG 3.1/2.5.2) Improve computational science capabilities, defined as the average annual percentage increase in the computational effectiveness (either by simulating the same problem in less time or simulating a larger problem in the same time) of a subset of application codes. Efficiency measure: X%

- Description of Problem Domain, Target Problems
- Description of Application Software, Algorithm Implementation

- Benchmark Parameters Q2, Q4
  - -problem instance
  - -build environment, build
  - -runtime environment, run script

- Benchmark Results Q2, Q4
  - -performance data
    - --wall time
    - --machine events
  - -simulation results

- Comparative Analysis of Q2 and Q4 results
  - -description of problem related findings
  - -description of software enhancements



FY 2010 Annual Report of Joule Software Metric SC GG 3.1/2.5.2, Improve Computational Science Capabilities

Date Published: September 2010

Prepared for
U.S. Department of Energy
Office of Science
Advanced Scientific Computing Research Program

"simulating the same problem in less time"

Algorithm, machine strong scaling :

Q4 problem := Q2 problem
Q4 algorithm := Q2 algorithm
Q4 machine ~ k * Q2 machine
Q4 time ~ 1/k * Q2 time

Algorithm enhancements, performance optimizations:

Q4 problem := Q2 problem
Q4 algorithm ~ enhanced Q2 algorithm
Q4 machine := Q2 machine
Q4 time ~ 1/k * Q2 time

*Could consider other variations: algorithm and machine are varied to achieve reduction of compute time

"simulating a larger problem in same time"

Algorithm, machine weak scaling (100%):

Q4 problem ~ k * Q2 problem
Q4 algorithm := Q2 algorithm
Q4 machine ~ k * Q2 machine
Q4 time := Q2 time

Algorithm enhancements, performance optimizations:

Q4 problem ~ k * Q2 problem
Q4 algorithm ~ enhanced Q2 algorithm
Q4 machine := Q2 machine
Q4 time := Q2 time

*Could consider other variations: problem, algorithm and the machine are varied to achieve fixed time assertion

Computational Efficiency

- Total elapsed time to execute a problem instance with a specific software instance (algorithm) on a machine instance
- Parallel
  - e(n,p) := Tseq (n) / (p * T(n,p))

Examples: Machine Perspective of Performance Enhancements

Strong Scaling

Machine Events	Q2	Q4
INS	2.147E+15	2.1130E+15
FP_OP	5.896E+14	5.8947E+14
PEs	5632	11264
Time[s]	121.252233	57.222988

INS:
2113046508030116 /
2146627269408190 = **.9843**

FP_OP:
589469277576687 /
589624961638025 = **.9997**

PEs: 11264 / 5632 = **2**

Time[s]:
57.222988 / 121.252233 = **.472**

Weak Scaling

Machine Events	Q2	Q4
INS	5.18E+17	1.93E+18
FP_OP	4.63E+17	1.81E+18
PEs	7808	31232
Time[s]	25339	23791

INS: 3.72

FP_OP: 3.92

PEs: 4

Time[s]: .938

NB: k= T(Q4)*PEs(Q4)/
T(Q2)*PEs(Q2) ~ 3.756

Improve Efficiency

Machine Events	Q2	Q4
INS	3.16E+12	4.37E+11
FP_OP	5.50E+11	5.53E+11
PEs	1	1
L2DCM	823458808	34722900
Time[s]	826.494142	79.414198

INS: 0.1381 (7.239x)

FP_OP: 1.0053 (0.99475x)

PEs: 1

L2DCM: 0.0422 (23.715x)

Time[s]: 0.0961 (10.407x)

Results Summary: FY10 Benchmark Exercises

Application	TD-SLDA	POP	LS3DF	Denovo
Problem	***Q2 : Nuclear 198W study*** • Z=74, N=124 • 40 x 40 x 40 lattice • 7,466 p-quasiparticle • 8,946 n-quasiparticle • 200 time steps • 0.75fm spacing • 100MeV cutoff ***Q4 : Nuclear 238U study*** • Z=92, N=146 • 40 x 40 x 64 lattice • 67,118 p-quasiparticle • 69,508 n-quasiparticle • 200 time steps • 1.25fm spacing • 100MeV cutoff	***3 simulated days, ocean-only model*** • 0.1-degree tripole global grid (3600×2400) • 42 vertical levels • 10 minute time steps • High-frequency output time slice	***Self-consistent DFT calculation for ZnO nanorod*** • 2776 atoms • 24220 valence electrons, d-electrons in valence band • 720×300×300 numerical grid	***Q2 : Full Core EDF PWR900 benchmark*** • 17x17 fuel assemblies • 17x17 fuel pins per assembly • 2x2 cells per pin cell • 3 fuel enrichments • 45 homogenized pin cell materials per assembly • 135 different pin cell materials • 233,858,800 (578x578x700) cells • 168 angles, 1 moment, 2 energy (fast and thermal) groups • 7.86×10^{10} total unknowns ***Q4 : Full Core EDF PWR900 benchmark*** • 168 angles, 1 moment, 44 energy (fast and thermal) groups • 1.73×10^{12} total unknowns
Hardware (cores)				
Q2	(s)73,728; (td)16,414	4,800	43,200	17,424
Q4	(s)217,800; (td)136,628	9600	86,400	112,200
Time (seconds)				
Q2	(s)6538.5, (td)2084.4	957.8	13,932	11,260.8
Q4	(s)18393.2, (td)2031.5	290.3	5328	1121.6
Metric target	(s)Q2:Q4 efficiency ≥ 1.0 ; (td)Q2:Q4 time ≥ 1.0	Q2:Q4 time ≥ 2.0	Q2:Q4 time ≥ 2.0	Q2:Q4 efficiency ≥ 1.0
Metric result	(s)Q2:Q4 efficiency = 2.11 (td)Q2:Q4 time = 1.026	Q2:Q4 time = 3.2992	Q2:Q4 time = 2.6	Q2:Q4 efficiency = 31

Results Summary: FY09 Benchmark Exercises

Application	VisIt		CAM	XGC1	RAPTOR
Metric	Image construction/display time	Image construction/display time	Simulation time	Grind time and particle rate Time per time step Particles pushed per second	Grind time Time per cell per time step
Problem	Isosurface • 1,024 × 1,024 pixels • Iso @ 0.001, 0.01, 0.1, 1.0, 10.0, 100.0 • Q2 dataset: 103.7M cells, 4,096 cores, 27 groups • Q4 dataset: 321.1M cells, 12,720 cores, 27 groups	Volume render • 1,024 × 1,024 pixels • 2,000 samples per ray • Q2 dataset: 103.7M cells, 4,096 cores, 27 groups • Q4 dataset: 321.1M cells, 12,720 cores, 27 groups	1 simulated month • T341 mesh • 150 sec time step • 26 vertical levels • Spectral Eulerian core	DIII-D experimental tokamak • 13.5B particles • Q2: 4,000 time steps • Q4: 16,000 time steps	DLR-A configuration • 50 time steps • 110 × 40 jet diam in axial and radial directions • Q2: 10,285,056 cells • Q4: 24,261,120 cells
Hardware (cores) **Q2** **Q4**	 4,096 12,720	 4,096 12,720	 8,192 8,192	 29,952 119,808	 47,616 112,320
Time (seconds) **Q2** **Q4**	 0.01778 per contour 0.01686 per contour	 28.729 6.378	 6,481.724 3,241.144	 86,400 75,600	 1,034.0 444.0
Metric target	Q2:Q4 contour time ≥ 1.0	Q2:Q4 time ≥ 3.10	Q2:Q4 time ≥ 2.0	Q2:Q4 grind time ≥ 1.0 Q2:Q4 particle rate ≥ 4.0	Q2:Q4 grind time ≥ 1.0
Metric result	1.05	4.50	2.10	1.14 4.57	2.34

Results Summary: FY08 Benchmark Exercises

Application	DCA++	GYRO	PFLOTRAN
Metric	time / disorder configuration	timesteps / second / process	time / dof / PE
Problem	$N_{dis} = 64, N_c = 16, N_t = 150$	$\mu = 30$, 10 timesteps	$64.8M$ DOFs, 200 flow, transport steps
Hardware Used	7808 PEs	4608 PEs	4000 PEs
Walltime	25339 s	17.23 s	2594 s
Instructions	5.1805×10^{17}	2.2410×10^{14}	2.2222×10^{16}
Floating Point Ops	4.6270×10^{17}	6.8320×10^{13}	1.2898×10^{15}

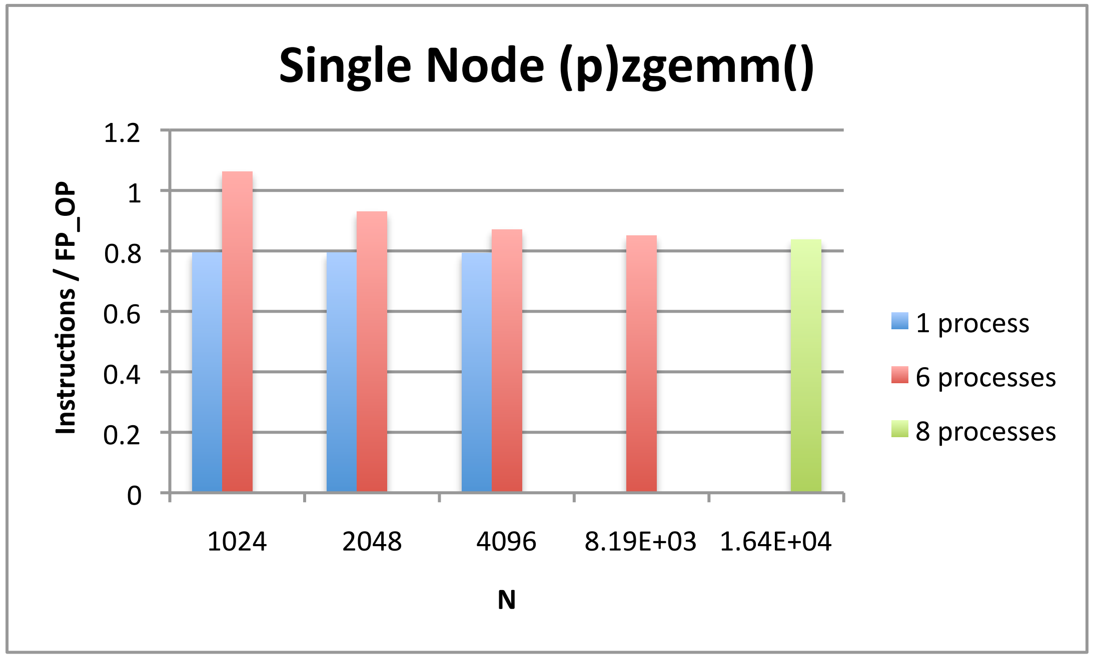
Application	DCA++	GYRO	PFLOTRAN
Metric	time / disorder configuration	timesteps / second / process	time / dof / PE
Problem	$N_{dis} = 256, N_c = 16, N_t = 150,$	$\mu = 40$, 10 timesteps	129, 635, 520 DOFs, Q2 stepping
Hardware Used	31232 PEs	24576 PEs	8000 PEs
Walltime	23791 s	152.75 s	2958.36 s
Instructions	1.9300×10^{18}	1.2202×10^{16}	5.0374×10^{16}
Floating Point Ops	1.8126×10^{18}	6.0882×10^{15}	2.8603×10^{15}

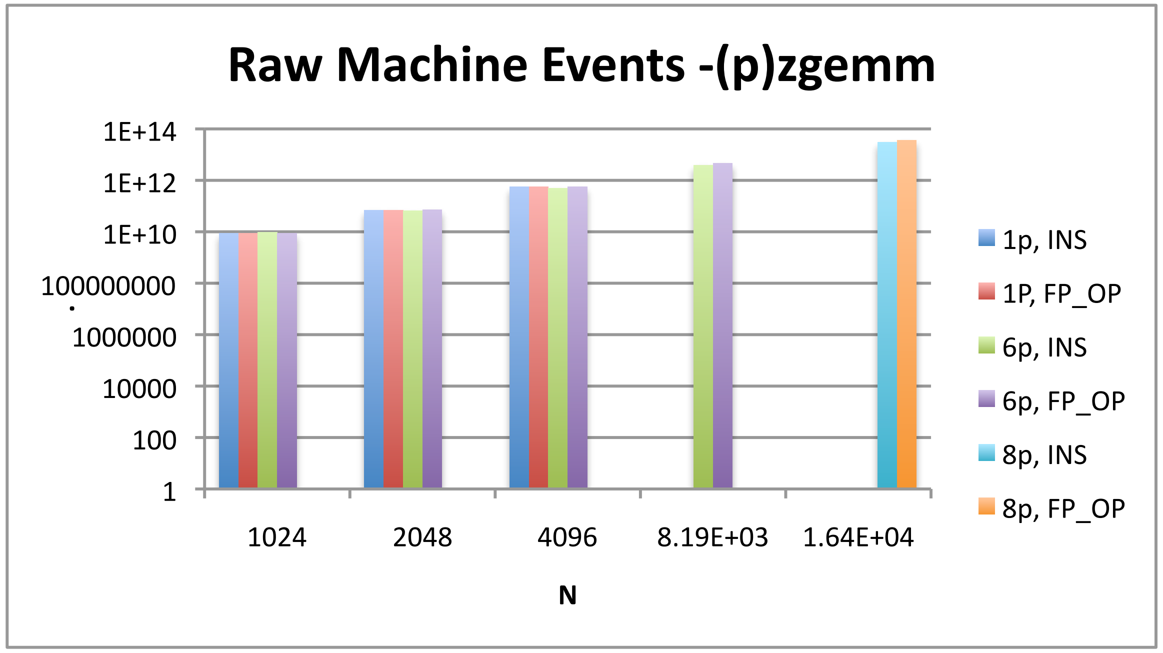
TOTALS	Q2	Q4	ratio (Q4 : Q2)
$\sum$ Walltime	27950.23 s	26902.11 s	.9625
$\sum$ PEs	16416	63808	3.8869
$\sum$ Instructions	5.4049×10^{17}	1.9925×10^{18}	3.6866
$\sum$ Floating Point Ops	4.6405×10^{17}	1.8215×10^{18}	3.9253

Floating Point Intensity of DOE Mission Applications: Are We Really Dominated by FLOPs?

Application	1	2	3	4	5	6	7
Instructions Retired	1.99E+15	8.69E+17	1.86E+19	2.45E+18	1.24E+16	7.26E+16	8.29E+18
Floating Point Ops	3.52E+11	1.27E+15	1.95E+18	2.28E+18	6.16E+15	4.15E+15	3.27E+17
INS / FP_OP	5.64E+03	6.84E+02	9.56	1.08	2.02	17.5	25.3

REFERENCE FLOATING POINT INTENSE PROBLEM :: Dense Matrix Matrix Multiplication
C <--- a A B + b C :: OPERATIONAL COMPLEXITY :A[m,n] , B[n,p] , C[m,p] :: [8mpn + 13mp] FLOP
E.g. m=n=p=1024 ---> 8603566080 FLOP , measure 8639217664





Benchmark Aggregated Computational Costs

i.e. how much does it cost to improve our applications?

Fiscal Year*	Benchmark CPU-Hours
2005	24,814
2006	211,888
2007	314,459
2008	2,718,788
2009	39,300,189
2010	78,289,735

*FY04 numbers are available but unreliable

Fiscal Year	CPU-Hours Awarded
2010	150M
2011	100M + Dirac at NERSC

Remaining Time Goes to Applications for Production

Results Summary: FY10 Benchmark Exercises

"**Real-Time Dynamics of Quantized Vortices in a Unitary Fermi Superfluid**," Science, 10 June 2011: Vol. 332 no. 6035 pp. 1288-1291 DOI: 10.1126/science.1201968

Application	TD-SLDA	POP	LS3DF	Denovo
	Q2 : Nuclear 198W study • Z=74, N=124 • 40 x 40 x 40 lattice • 7,466 p-quasiparticle • 8,946 n-quasiparticle • 200 time steps [illegible] • 69,508 n-quasiparticle • 200 time steps • 1.25fm spacing • 100MeV cutoff	***3 simulated days, ocean-only model*** • 0.1-degree tripole global grid (3600×2400) • 42 vertical levels • 10 minute time steps [illegible]	***Self-consistent DFT calculation for ZnO nanorod*** • 2776 atoms • 24220 valence electrons, d-electrons in valence band • 720x300x300 numerical [illegible]	***Q2 : Full Core EDF PWR900 benchmark*** • 17x17 fuel assemblies • 17x17 fuel pins per assembly • 2x2 cells per pin cell • 3 fuel enrichments [illegible] groups • 7.86×10^{10} total unknowns ***Q4 : Full Core EDF PWR900 benchmark*** • 168 angles, 1 moment, 44 energy (fast and thermal) groups • 1.73×10^{12} total unknowns
Hardware (cores)				
Q2	(s)73,728; (td)16,414	4,800	43,200	17,424
Q4	(s)217,800; (td)136,628	9600	86,400	112,200
Time (seconds)				
Q2	(s)6538.5, (td)2084.4	957.8	13,932	11,260.8
Q4	(s)18393.2, (td)2031.5	290.3	5328	1121.6
Metric target	(s)Q2:Q4 efficiency ≥ 1.0 ; (td)Q2:Q4 time ≥ 1.0	Q2:Q4 time ≥ 2.0	Q2:Q4 time ≥ 2.0	Q2:Q4 efficiency ≥ 1.0
Metric result	(s)Q2:Q4 efficiency = 2.11 (td)Q2:Q4 time = 1.026	Q2:Q4 time = 3.2992	Q2:Q4 time = 2.6	Q2:Q4 efficiency = 31

ASCR's Benchmark Trends (FY04 - FY11)

climate research	4
condensed matter	4
fusion	5
high energy physics	3
nuclear	2
subsurface modeling	2
astrophysics	2
combustion chemistry	4
bioinformatics	1
math, data analytics	2
molecular dynamics, electronic structure	3
nuclear energy	1
Total	33

Cray	X1
	X1E
	XT3
	XT4
4-core	XT5
6-core	XT5
IBM	SP Power3
	P690
	Power5
	BG/L
SGI	Altix
HP Itanium-2	
QCDOC	
Intel / NVIDIA	w/ IB

**DOE's Advanced Scientific Computing Advisory Committee approves annual application / machine studies

Target Computing Platforms: Today, Yesterday

Hex-Core AMD Opteron (TM)	2.6e9 Hz clock	4 FP_OPs / cycle / core 128 bit registers
PEs	18,688 nodes	224,256 cpu-cores (processors)
Memory	16 GB / node 6 MB shared L3 / chip 512 KB L2 / core 64 KB D,I L1 / core	dual socket nodes 800 MHz DDR2 DIMM 25.6 GBps / node memory bw
Network	AMD HT SeaStar2+	3D torus topology 6 switch ports / SeaStar2+ chip 9.6 GBps interconnect bw / port 3.2GBps injection bw
Operating Systems	Cray Linux Environment (CLE) (xt-os2.2.41A)	SuSE Linux on service / io nodes

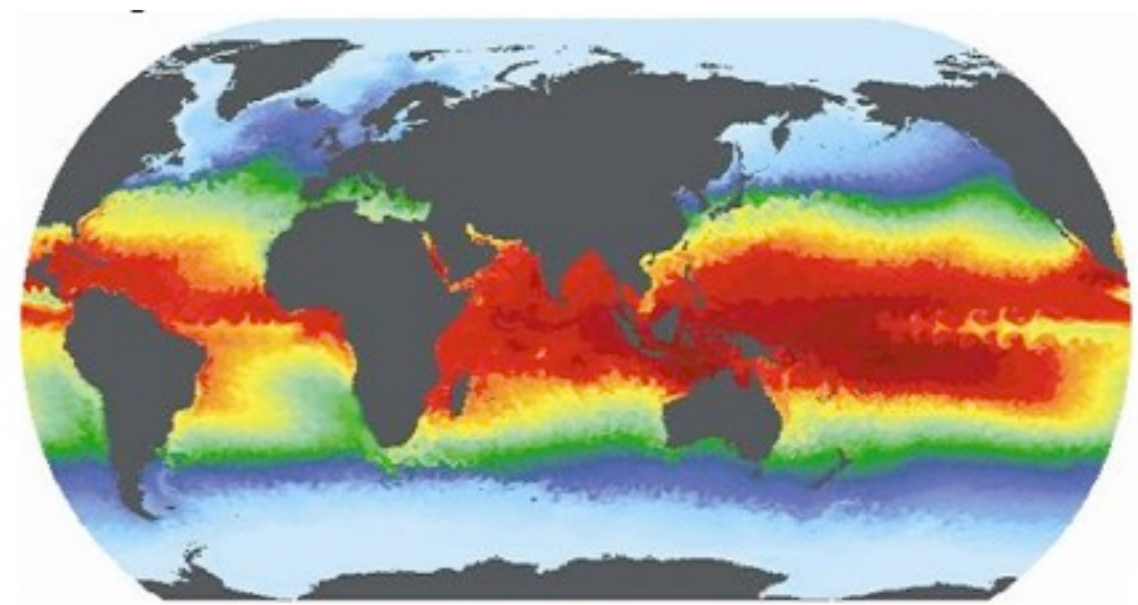
FY	Aggregated Cycles	Aggregated Memory	Aggregated FLOPs	Memory/FLOPs
2008	65.7888 THz	61.1875 TB	263.155 TF	0.2556
2009	343.8592 THz	321.057 TB	1.375 PF	0.2567
2010 / 11	583.0656 THz	321.057 TB	2.332 PF	0.1513

POP w/ Phil Jones (LANL) et al

- Parallel Ocean Program (POP) is an ocean general circulation model used for ocean and climate studies

- (to now) POP is coupled to atmosphere, land, and sea-ice models and run at a relatively coarse resolution to achieve maximum simulation throughput over centuries of simulation time

- POP is capable of resolving the mesoscale eddies that influence global ocean circulation over the course of simulated decades



- The CCSM6 collaboration is developing a fully coupled, high-resolution configuration of the CCSM using the eddy-resolving POP model coupled to a 25 km resolution atmosphere model; this model will be run for century-scale climate change simulations

- Output for the climate-coupled model will be larger and occur more frequently than it does in the ocean-only mode mode run at high resolution today

- **Throughput of more than one simulated year per CPU day is required for the fully coupled system**

POP

Benchmark Details :

-ocean-only but with coupled CCSM6 requirements in resolution and I/O
- -- 0.1 degree global grid (3600 × 2400 × 42 grid points)
- -- tracer advection via centered spatial discretization
- -- biharmonic lateral mixing for both tracers and momentum
- -- vertical mixing is performed using the k-profile parameterization (KPP)
- -- 3 simulated days at 10m time steps
- -- data dump each simulated day -- as opposed to each month at this resolution

-I/O became clear focal point
- -- observable and movie data need to be recorded
- -- observables are 8 3D fields and 19 2D fields
  - --- 11.4262104 GB / day, or about 35 GB for the benchmark
  - --- 1 observable file / day
- -- 60 movies formed each day from coordinate data
  - --- 3600 × 2400 coordinate movie data is decomposed over a virtual 60 × 80 rectangular process grid; each process has 60 × 30 block of the global data
  - --- 60 × 4 × 60 × 30 × 4800 B / day = 1.931190491 GB / day or 5.793571472 GB for the benchmark

Aside on FILEs and IO

ANSI C
- stream of BYTEs
- points to a FILE structure
- fopen,fwrite,fread,fclose

Fortran
- sequence of records
- open,write,read,close
- IOLENGTH , RECL

```
void f_copn_ ( char * ffn , int * ffd , int * len ) ;

void f_ccls_ ( int * ffd ) ;

void f_crm_ ( char * ffn , int * len ) ;

void f_cwr_ ( int * ffd , void * fbf , int * fsz , int * nobj , int * ierr ) ;

void f_crd_ ( int * ffd , void * fbf , int * fsz , int * nobj , int * ierr ) ;
```

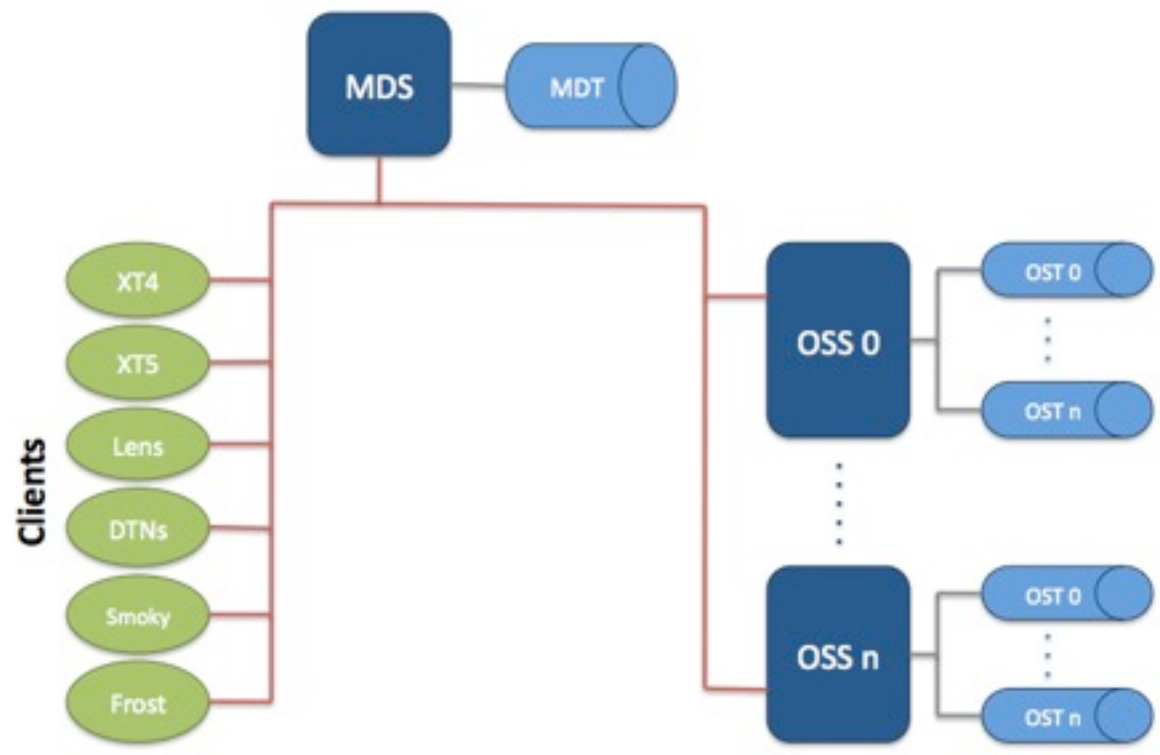
```
fn = '/tmp/work/roche/mpt-omp/ben.txt'//
CHAR(0)

call f_copn ( fn , fd , LEN( fn ) )

call f_cwr ( fd , a , 16 , ndim , ierr )

call f_ccls ( fd )

call f_copn ( fn , fd , LEN( fn ) )

call f_crd ( fd , a_bk , 16 , ndim , ierr )

call f_ccls ( fd )

call f_crm ( fn , LEN( fn ) )
```

```
typedef struct  {
        int              level;      /* fill/empty level of buffer */
        unsigned         flags;      /* File status flags          */
        char             fd;         /* File descriptor            */
        unsigned char    hold;       /* Ungetc char if no buffer   */
        int              bsize;      /* Buffer size                */
        unsigned char    *buffer;    /* Data transfer buffer       */
        unsigned char    *curp;      /* Current active pointer     */
        unsigned         istemp;     /* Temporary file indicator   */
        short            token;      /* Used for validity checking */
}       FILE;
```

```
0 -rw-r--r-- 1 roche roche 1608 2010-06-21 21:03 fortran-dat.bn
0 -rw------- 1 roche roche 1600 2010-06-21 21:03 c-data.dat
```

Aside on FILEs and IO (2)

POSIX (UNIX)
•stream of BYTES
•file descriptors
  -index into file descriptor table
  -kept in user process
  -points to entry in system in-memory inode table
•open,write,read,close, ioctl



Spider (Lustre) :

•MDS, file names and directories in the filesystem, file open, close, state mgt

•OSS, provides file service, and network request handling for set of OSTs

•OST, stores chunks of files as data objects -may be stripped across one or more OSTs
- -Spider has 672 OSTs
- -7 TB per OST
- -1 MB Default stripe size
- -4 Default OST count

Aside on FILEs and IO (3)

```
module load liblut ; -LUT

lut__open() ;

lut__close() ;

lut_putl() ;

pwrite() ;

pread() ;
```

- form modulo classes from MPI communicator over the number of I/O groups
  - for both proton and neutron communicators in nuclear case (44 for protons, 44 for neutrons)

- fit the stripe size to the largest single data item if possible
  - eg for nuclear code and 32^3 lattice, a single 4-component term is 4 * 32^3 * 16 / 2^20 = 2MB

- set the stripe pattern (I use round-robin) and number of target OSTs (I use 88 in nuc code) for target PATH / FILE
  - eg lfs setstripe /tmp/work/roche/kio -s 2m -i -1 -c 88

Performance: POSIX ~ [225,350]MBps , use of Lustre ~ [2,15]GBps

Aside on FILEs and IO (4) - POP Approach

- introduced set of parallel I/O processes within the MPI group
  - (was) gather to single process, followed by sequential write / wait phase within a loop over fields (1 PE writes, nPEs - 1 PEs wait) x nFIELDS iterations
  - (is) loop over (disjoint target) gathers to a set of designated IO PEs; after gather phase then (nIOPEs write in parallel, nPEs - nIOPEs wait) x 1 since nIOPEs > nFIELDS (8 (3D fields / day) × 42 (k-values / fields) × 1 (PE / k-value) = 336 IOPEs / day; 19 IOPEs / day for 2D fields)
- use of lut_putl() library function explicitly invoking LUSTRE file system semantics
- oracle code to search for preferred LUSTRE parameters: number of OSTs, stripe size, number of writers
- similar enhancements for 2D fields; movies require an additional index transformation which is done locally by the IO PE prior to writing (block cyclic to natural column major)

```
memcpy( ( void * ) fnbf , ( const void * ) ffn , ( size_t ) *len ) ;
for ( iniopes = 0 ; iniopes < 6 ; iniopes++ )
 for ( iscnt = 0 ; iscnt < 7 ; iscnt++ )
  for ( istrp = 0 ; istrp < 6 ; istrp++ )
   {
    sprintf( fn , "%s/lpop-io%d-sc%d-str%d" , fnbf , iniopes , iscnt , istrp ) ;
    b_t() ; /* start running internal clock */
    wr_lstr_orcl( fn , com , ndays , ndddfld , nddfld , ni , nj , nk , strp[ istrp ] , scnt[ iscnt ] , niopes[ iniopes ] , dbf , dbf_ ) ;
    rt = e_t( 0 ) ;
    if ( ip == 0 )
     printf( "case: T[ %f ] ISTRP[ %d ] SCNT[ %d ] IOPEs[ %d ]\n" , rt , strp[ istrp ] , ( int ) scnt[ iscnt ] , niopes[ iniopes ] ) ;
```

POP

Q2

4800 PEs, Q2	Time(s)	INS	FP_OP
Barotropic	220.285649	3362619394734242	10914798749862
Baroclinic	84.623336	638046552543018	123489441332158
T_avg	554.416994	10459543609613288	22070416032
Movie	98.516514	1838543581529579	15638400
TOTALs	957.842493	1.629875313842013e+16	134,426,326,136,452

Q4,e

4800 PEs, Q4	Time(s)	INS	FP_OP
Barotropic	162.845484	2493523139608176	10918903717734
Baroclinic	81.234007	611926226154622	123489442062604
T_avg	72.995206	1369947333186195	22070417409
Movie	12.397561	228560389936546	15640101
TOTALs	329.472258	4,703,957,088,885,539	134,430,431,837,848

Q4,s

9600 PEs, Q4	Time(s)	INS	FP_OP
Barotropic	143.867992	4352776136294947	11696471278395
Baroclinic	47.994133	755616085382567	133265275114487
T_avg	84.648207	3180959264572214	24868719153
Movie	13.812455	505002308418671	31278501
TOTALs	290.322787	8,794,353,794,668,399	144,986,646,390,536

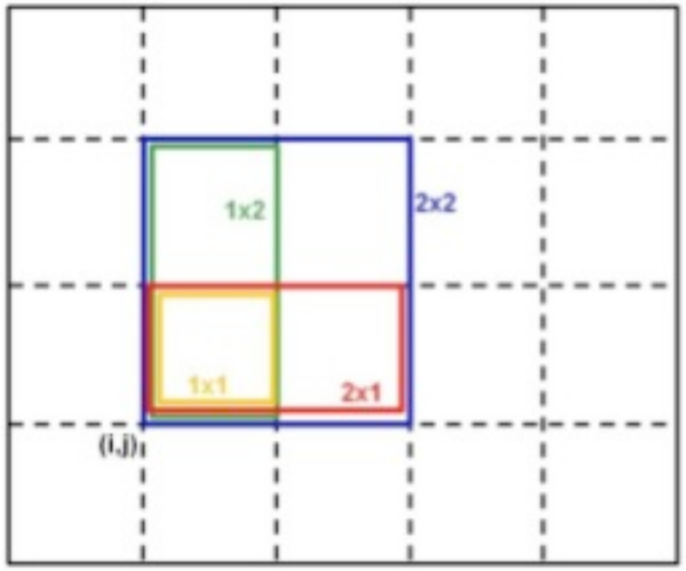
Efficiency:

```
PES     : 1
TIME    : 0.343973315454068 (329472258 / 957842493)
INS     : 0.288608401448646 (4703957088885539 / 1.629875313842013e+16)
FP_OP   : 1.000030542390869 ( 134430431837848 /    134426326136452  )
```

Strong Scaling:

```
PES     : 2
TIME    : 0.3031007593855 (290.322787 / 957.842493)
INS     : 0.539572181993355 (8794353794668399 / 1.629875313842013e+16)
FP_OP   : 1.078558423469556 (144986646390536 / 134426326136452)
```

LS3DF w/ Lin-Wang Wang (LBL) et al



LS3DF is a modern DFT solver for normal systems

• based on a divide-and-conquer charge density patching algorithm that cancels out the artificial boundary effects due to subdivision

• The fragment division is based on a real space grid, which is provided by the user. The grid cell corresponds to the smallest fragment size: the larger the fragment size, the more accurate the results. For good accuracy, the smallest fragment in a typical computation corresponds to roughly eight atom cells.

• ab initio ~ the total energy, the dipole moment, the band alignment, and the atomic positions

• linear since Coulomb is treated classically and local interactions are approximated

• resulting LS3DF total energy differs from the direct whole-system DFT calculations by only a few meV per atom

LS3DF

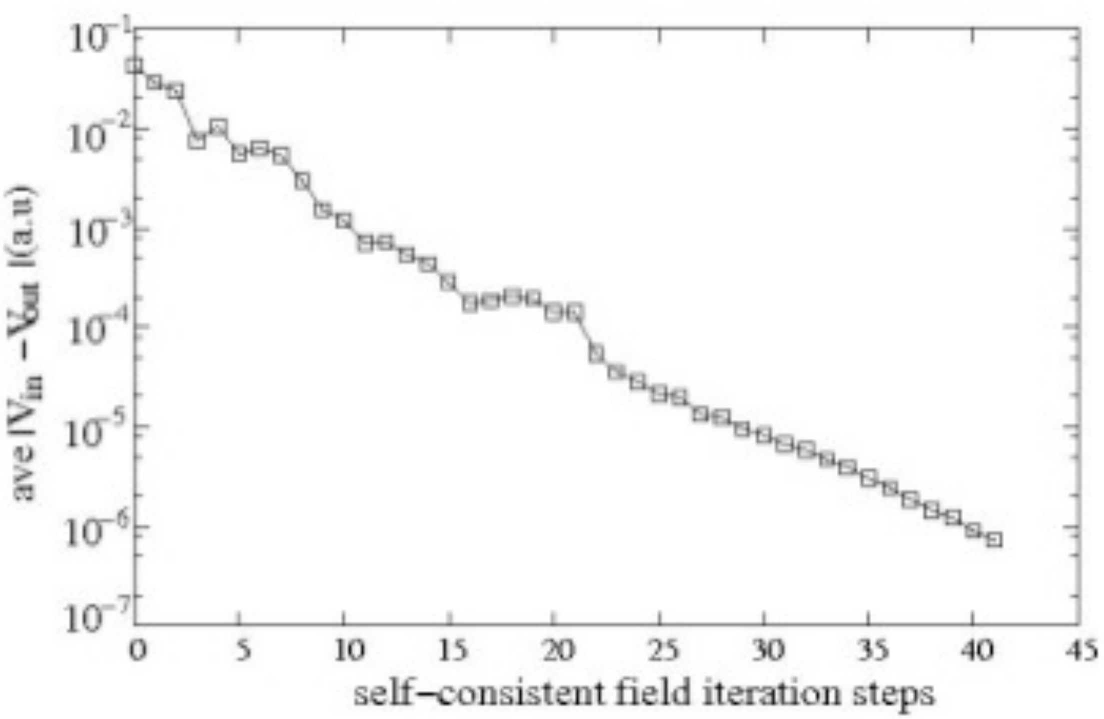
<u>Benchmark Problem</u>

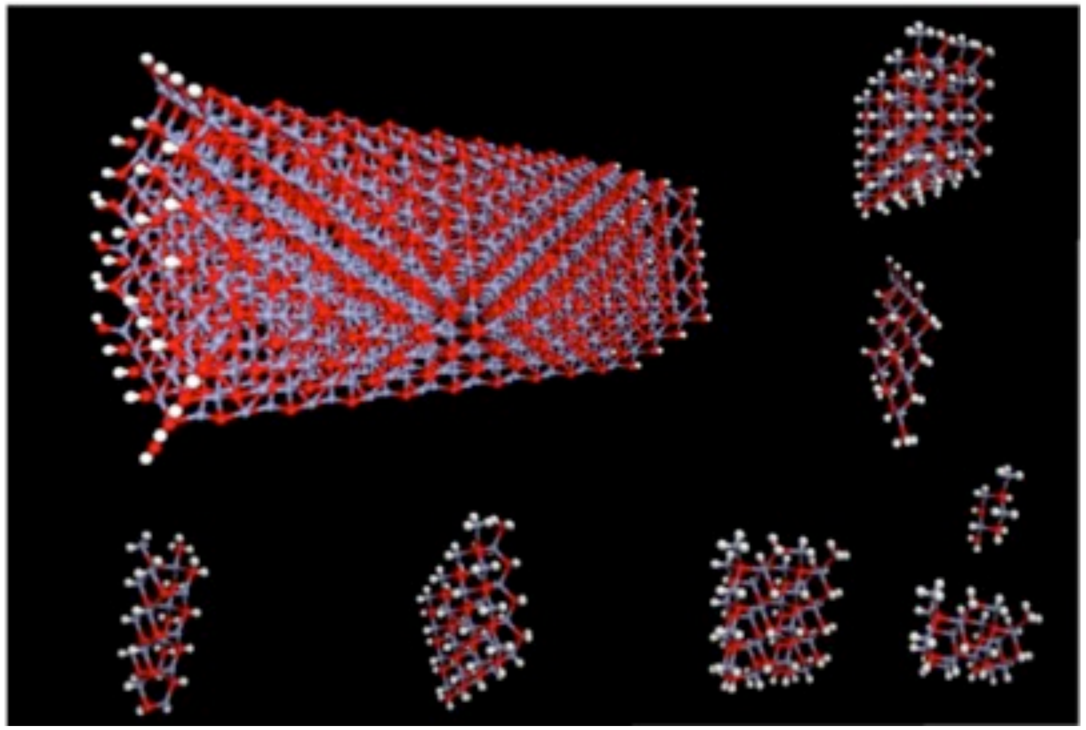
• compute the total charge density and potential and study the total dipole moment and internal electric field of a ZnO nanorod

• 2776 atom system, 24220 valence electrons; Zn *d*-electron is included in the valence electrons

• H passivates the bottom (O-terminated) and OH group is used to passivate the top (Zn-terminated) dipole surfaces

•20 initial iterations (fragment charge density), and 40 global self-consistent field (SCF) iterations





LS3DF

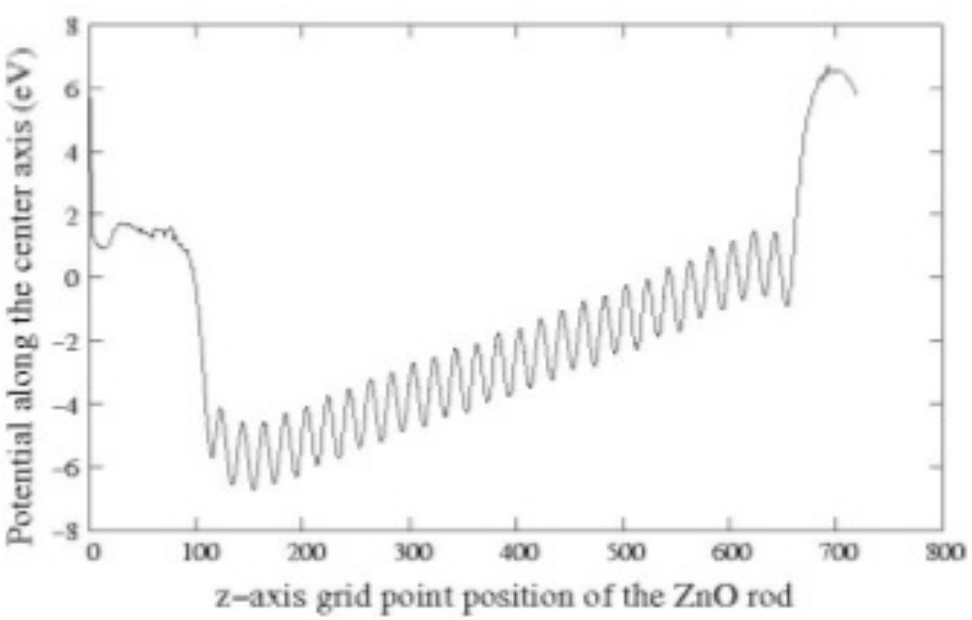
Enhancements

• introduced a wave function band index parallelization within the PEtot_F subroutine

• implemented a new algorithm: the direct inversion of the iteration space (DIIS) method, in addition to the conjugated gradient (CG) method, in the PEtot_F subroutine to converge the wave functions

• developed a better formula to estimate the computational time of each fragment, which allows a better static assignment of fragments into fragment groups thus improving the load balance between different fragment groups

Strong Scaling: PE(Q4)/PE(Q2) = 86400 / 43200 = 2 ,T(Q4)/T(Q2) = 5328s /13932s = .38



•large dipole moment and internal potential is found

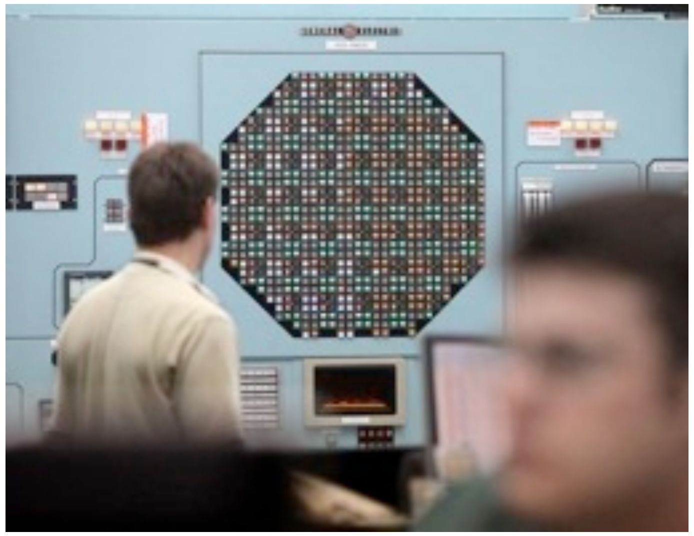
•tilting of the internal potential from one size of the rod to the other is about 6 Volts, which is larger than the ZnO band gap (3.3 eV). If such a large tilting occurs in a physical system, the occupied valence electron at one side will flow to the conduction band state at the other side - a self- compensation effect.

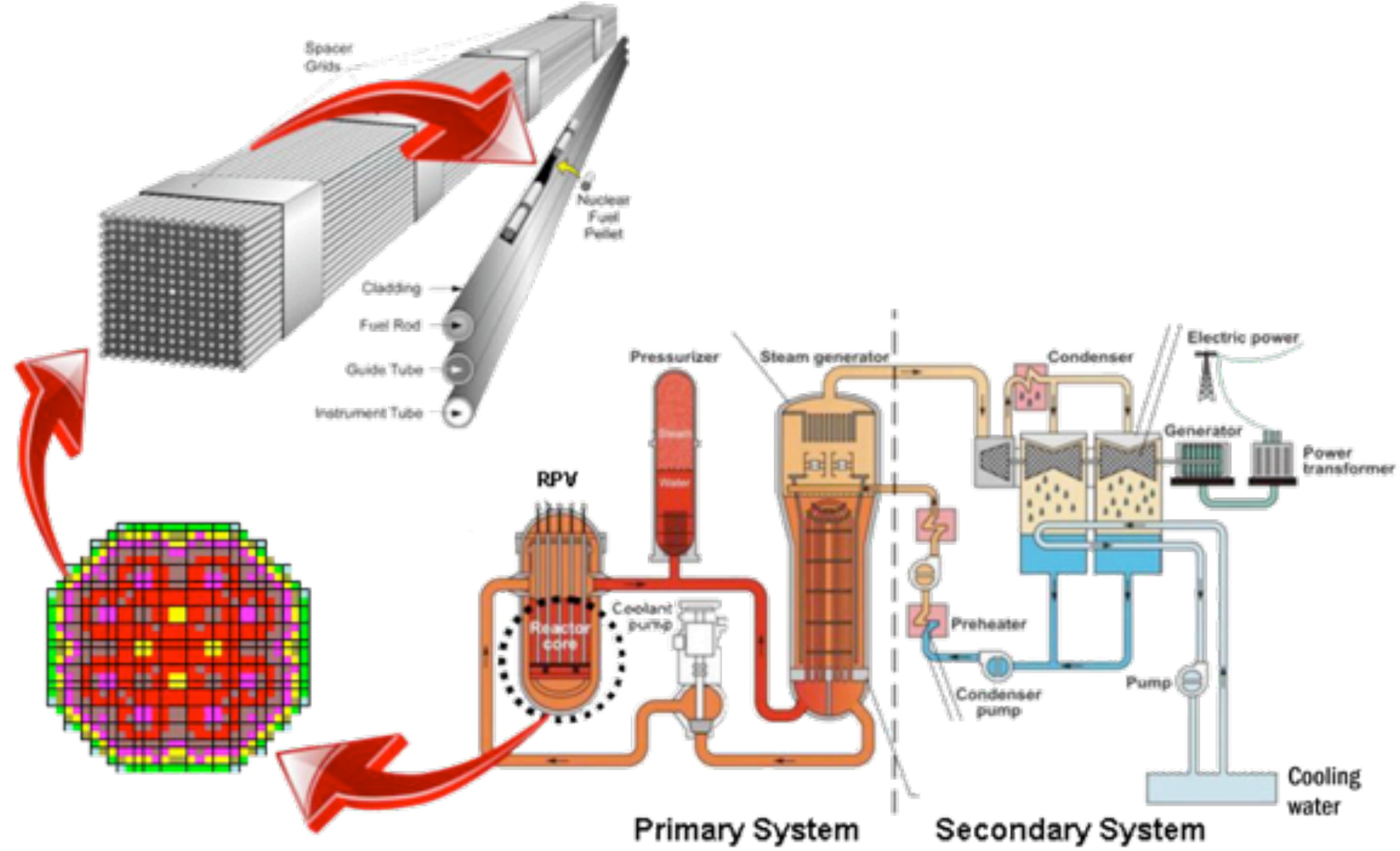
• in LS3DF method the large tilting is possible because we occupy each local fragment with a fixed number of electrons. This prevents electrons from flowing from one side to another while still allowing the dipole moment to exist.

•The ability to prevent charge compensation in the LS3DF method provides a means to study the total dipole moment effect without the additional complication of the charge flow, which depends on other factors like the surface electronic states.

Denovo

w/ Tom Evans (ORNL) et al





nuclear reactor analysis

- accurate characterization of the neutron distribution in the reactor in order to determine power, safety, and fuel and component performance

linear Boltzmann transport equation is used to model the neutron transport

- solves the time-independent linear Boltzmann equations using the discrete ordinates (SN) method. It also features a Monte Carlo module that can be used to solve the multigroup equations on the S spatial grid with continuous angular treatment.

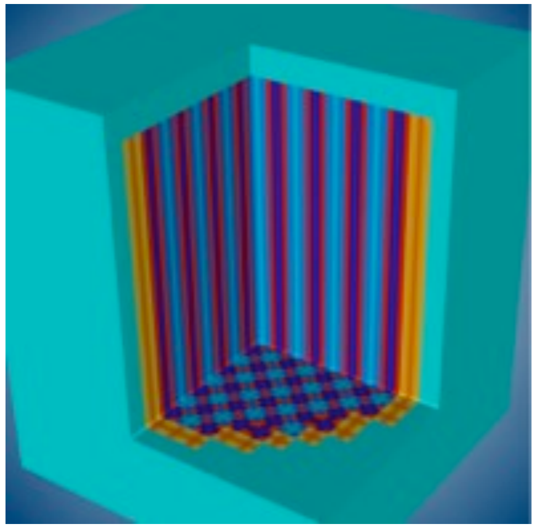
- solves for the *k*–eigenvalue and the scalar flux throughout the core

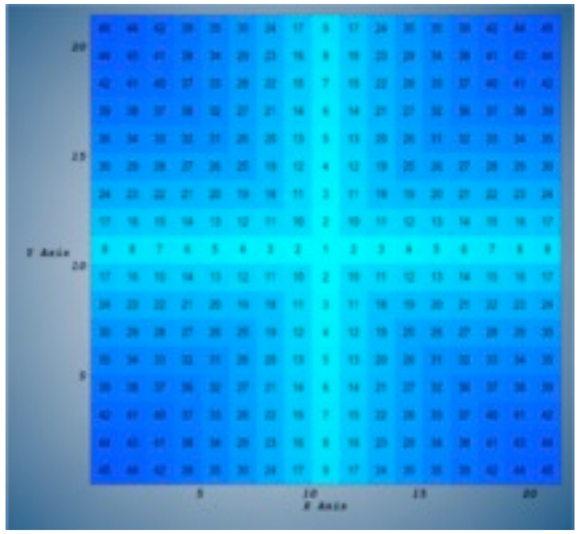
the pin power distribution, fission source, and groupwise power distributions can be subsequently analyzed

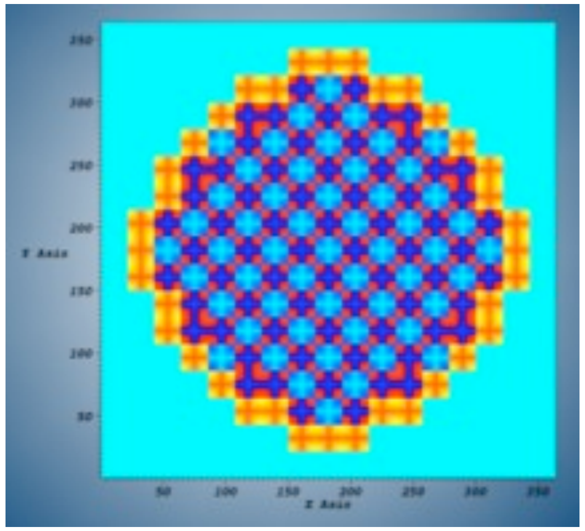
Solving pin-homogenized, whole-core problems with transport, as opposed to diffusion or other low-order approximations, is the first step towards fully predictive reactor core modeling and simulation

Denovo

- a full-core pressurized water reactor (PWR)
  - core height of 4m
  - core contains 289 (17×17) total assemblies, 3.6m height
    - 157 fuel, 132 reflector
  - three different fuel enrichments ranging from 1.5% to 3.25% (LEU, MEU, HEU) in the assemblies
  - each fuel assembly has 17×17 fuel pins
  - 45 pin-cells per assembly with 3 enrichment levels := 135 total materials
  - LEU (light blue), MEU (red/blue), and HEU (yellow/orange)



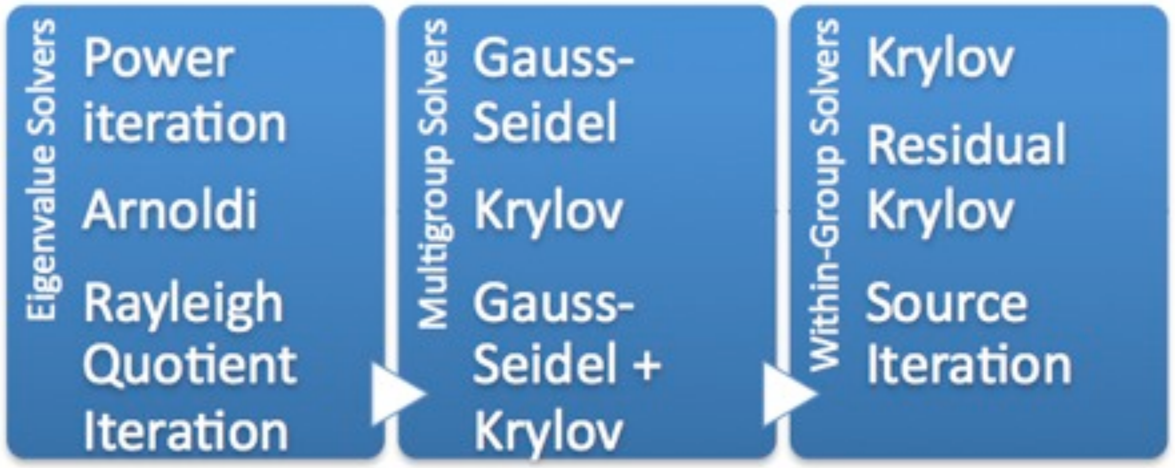




Denovo

Enhancements

a new set of advanced solvers was developed in Denovo enabling a multilevel decomposition over energy provides the necessary parallelism to scale to O(100K) cores



multi-group solvers

-energy is decomposed in sets, space-angle is decomposed in blocks

-can be used in inner iteration of eigensolver

within-group solvers

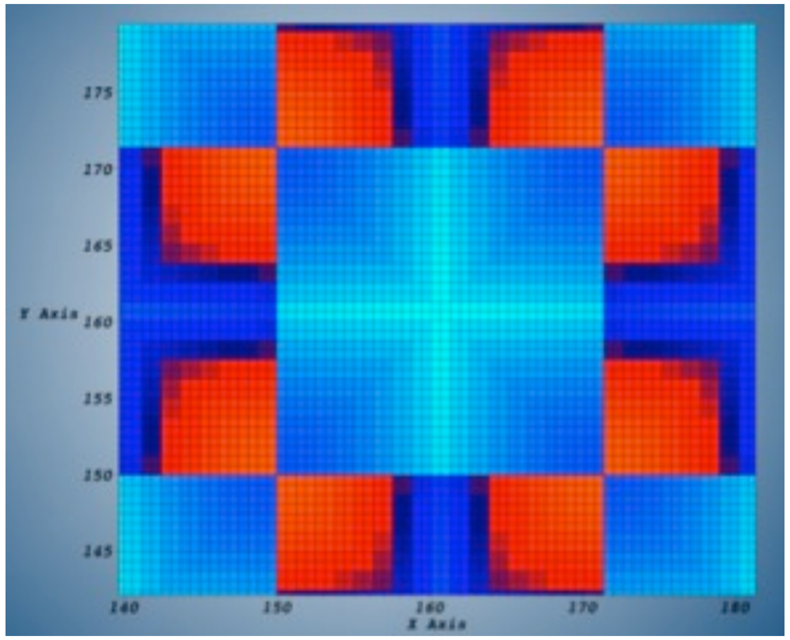
-parallelized over space w/ no coupling between energy groups so operate only within a set, not across sets.

eigensolvers

parallelization is determined by the choice of multigroup solver since some eigenvalue solvers can solve both energy-dependent and energy-independent eigenvectors

Best Case: Used the Arnoldi eigenvalue solver with a Krylov multigroup solver partitioned over 2 sets. The mesh decomposition was 102×100 with 10 z-blocks.

Denovo



Q2 , Q4 results
The power distribution in a full EDF PWR900 model core is computed. Solves for the k-eigenvalue and scalar flux throughout the core using a k_{eff} tolerance of 0.001 and an eigenvector tolerance of 0.10.

Other
2×2 spatial mesh array per pin cell
- 578 mesh cells in the x and y directions (0.63 cm width)
- 700 cells in the axial (z) direction (0.60 cm width)
- total ~ 233,858,800 cells (578x578x700)

solves a discretized Boltzmann equation consisting of one scalar unknown per cell -168 angular directions per scalar unknown

Q2
2 energy groups (fast and thermal)
DoFs := 7.86e10
PEs := 17,424
Time := 187.68 min (11,260.8 s)

Q4
44 energy groups
DoFs := 1.73e12
PEs := 112,200
Time := 1201.8s

Weak Scaling

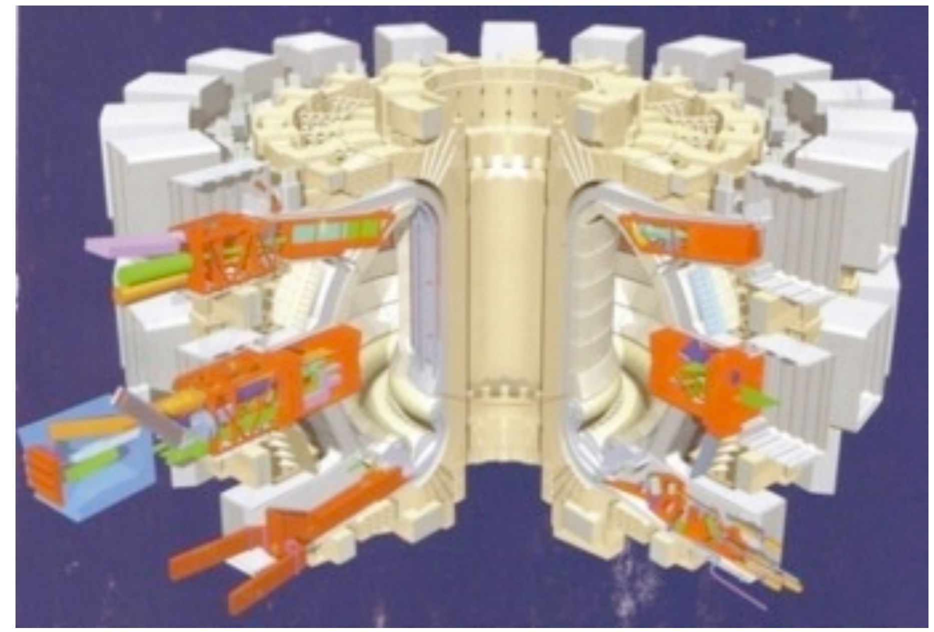
EGs := 22
PEs := 6.439
Time := .1067

~10X ideal hyper-weak scaling!

XGC1: 5D Gyrokinetic Full-Function Particle-in-Cell Model for Whole Plasma Dynamics in Experimentally Realistic Magnetic Fusion Devices

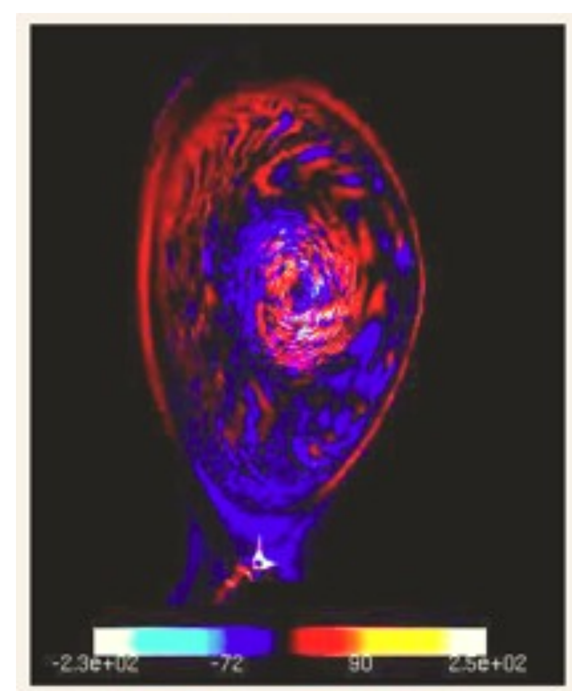
Model

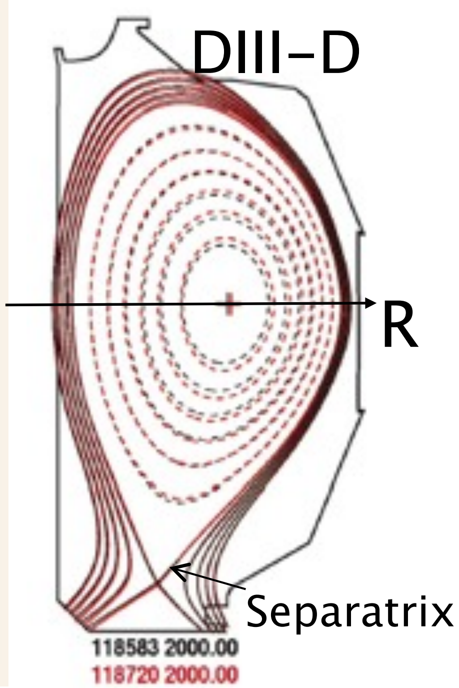
- Gyrokinetic “full-f” PIC model of magnetic fusion plasmas, with inclusion of magnetic separatrix, magnetic X-point, conducting material wall, & momentum/energy conserving Coulomb collisions
- Full-f description allows turbulence and background plasma to interact self-consistently and background plasma to evolve to a self-organized state
- **Focus:** understand and predict plasma transport and profile in the “edge pedestal” around separatrix



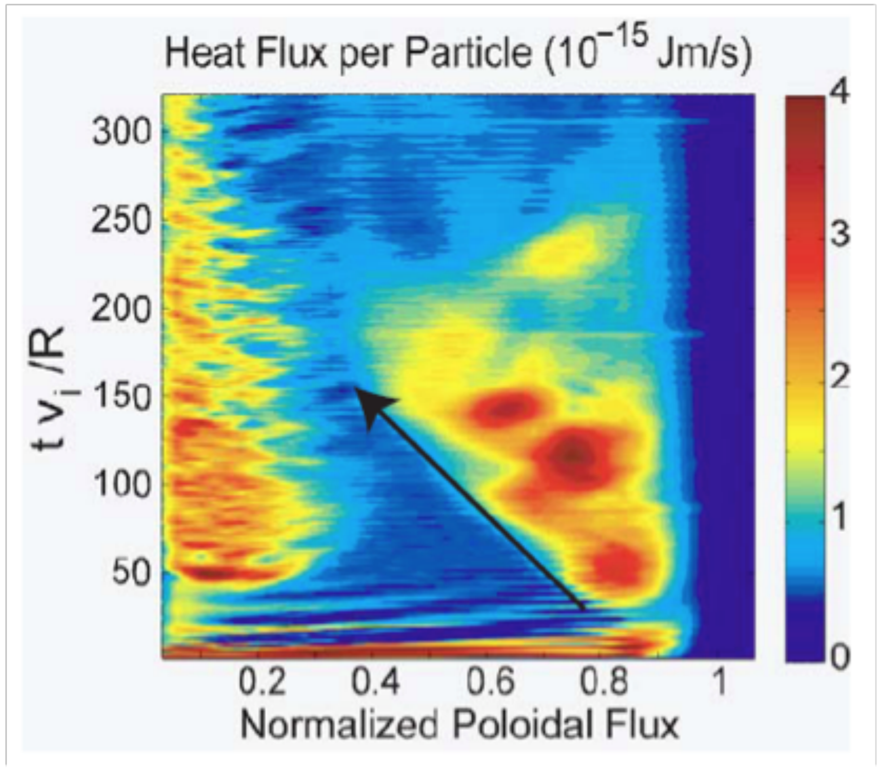
Algorithm & implementation

- Fixed unstructured grid following equilibrium magnetic field lines with embedded discrete marker particles representing ions, electrons, and neutral particles
- Marker particles time-advanced with Lagrangian equation of motion (either 4^{th} order PC or 2^{nd} order RK)
- Marker particle charges accumulated on grid, followed by gyrokinetic Poisson solve for electrostatic field
- PETSc for Poisson solve, ADIOS for I/O, Kepler for workflow, Dashboard for monitoring/steering

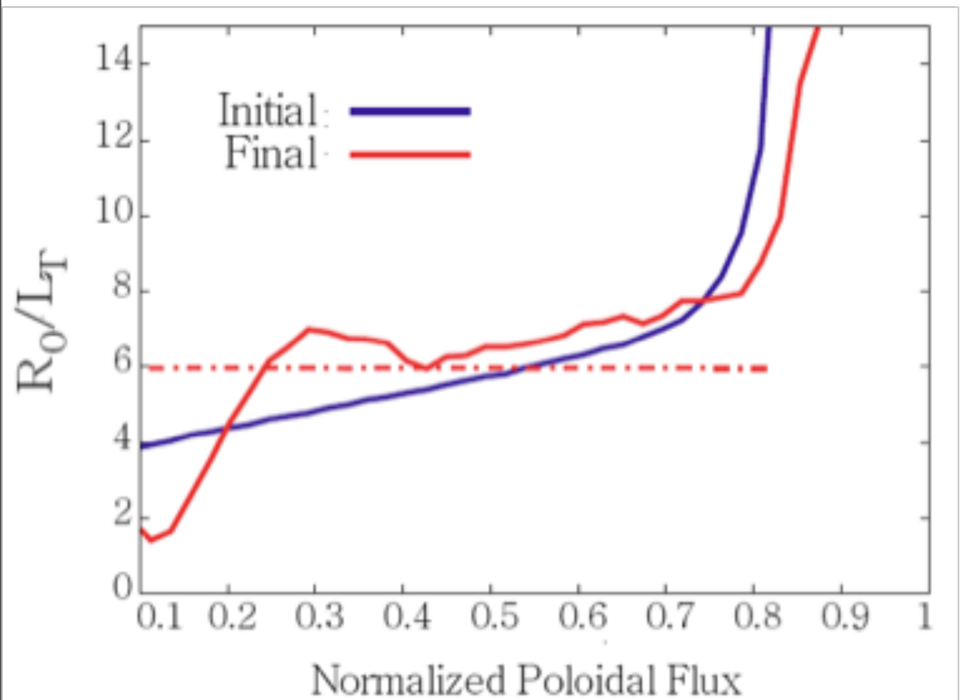




- High-confinement mode ("H-mode") and operation appears to be required for adequate yield ratios (Q>10) in magnetic toroidal fusion plasmas
  - At high enough core heating, plasma can bifurcate from low density/T state @ edge to very high just inside of magnetic separatrix; core temperature then continues to rise without the high T plasma contacting the wall (the "edge pedestal")
  - Core ion T increases in proportion to the edge pedestal T, with its radial slope being "stiff" and independent of the core heating power, entering into the "H-mode" of operation



- Many aspects of the H-mode remain poorly understand over the last 25 years
  - Why does the edge pedestal form this shape? Why is strong core heating necessary? Why is there an instantaneous central T_i and turbulence improvement after H-mode bifurcates? Why is the radial T_i profile stiff?



- First attempt to study the nonlocal H-mode coupling physics between the edge and core turbulence in a realistic DIII-D tokamak geometry
  - **Initial stage:** turbulence intensity propagation from edge to core, as a result of nonlocal interaction between edge and core. Initial turbulence intensity is strong and bursty. Plasma conditions not yet close to experimental state. **(Q2)**
  - **Final stage:** plasma in self-organized quasi steady-state, allowing probing of unexplained experimental H-mode phenomena **(Q4)**

XGC1: Performance Enhancements

- Solving gyrokinetic Poisson equation requires interpolating charges to grid points
- Solutions have to be interpolated back to particle positions to time evolve according to eqns of motion

  - B field is evaluated employing spatial splines at each spatial position
    - ✳ Precompute and store spline coefficients -search instead of recompute
    - ✳ Used common partial results in the computation of derivatives significantly decreasing the number of required floating operations per time step
  - Improve MPI communication in Poisson solution
  - Improve MPI communication in the reassignment of particles to processes
  - OMP parallelism was implemented allowing the use of 1/4 as many MPI processes

Measurements In Nested Loop Constructs

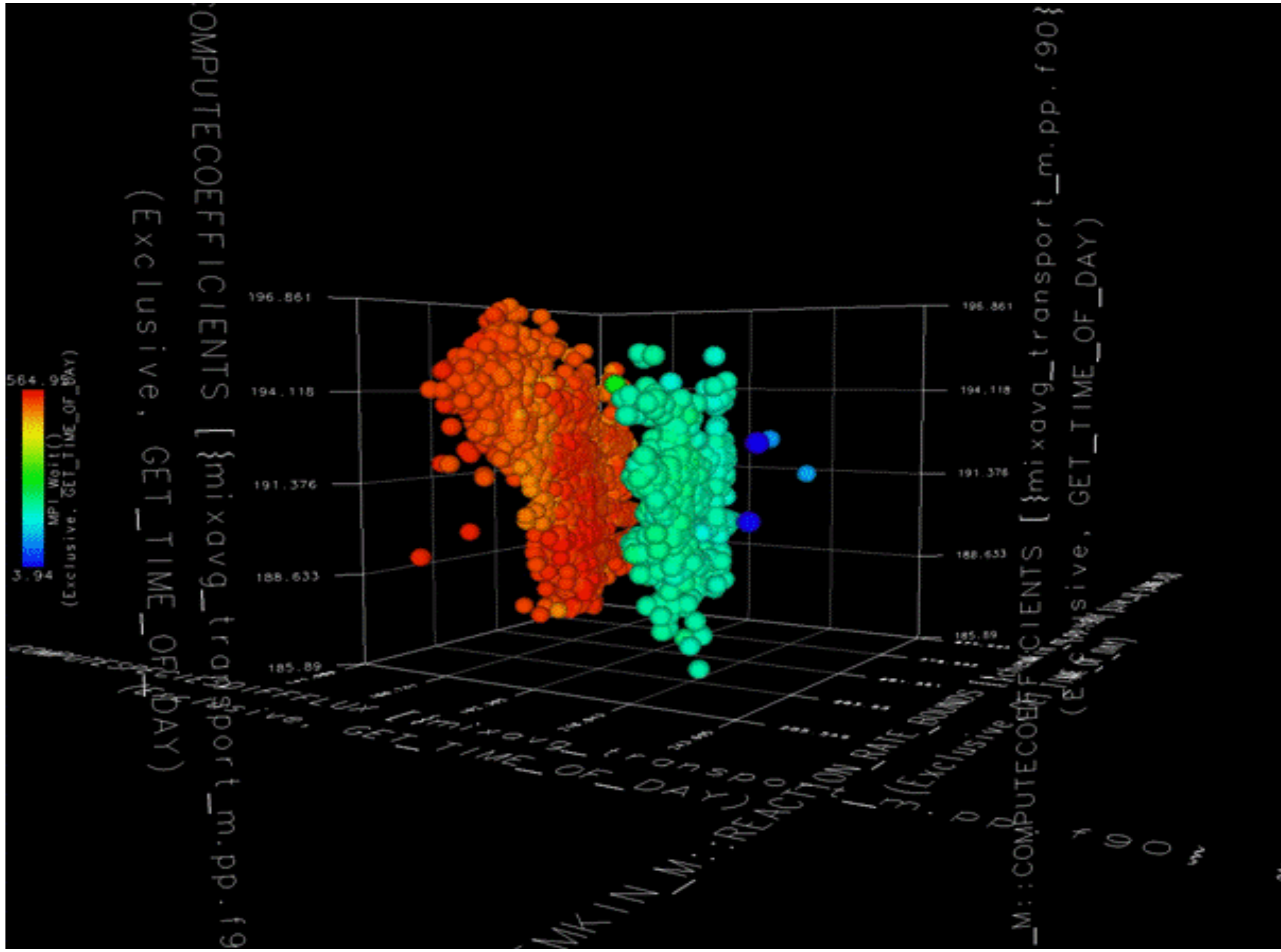
3 Loop Iterations
2 Computing Phases (different zgemm versions/instances -since we know what should happen)
10 PEs

```
roche@jaguarpf-login1:/tmp/work/roche/joule-q4> time aprun -n 10 ./xfusr-krp
 m l n
32 32 32
 m2 l2 n2
128 128 128
 nits
3

THY P1( FP_OPS ) = PEs * nits * (8.m.n.l + 13.m.n)            ==     8263680
THY P2( FP_OPS ) = PEs * nits * (8.mm.nn.ll + 13.mm.nn)  ==   509706240

                    time          ins           fp           dm
P-1:                2201      33936960      8294400        16114
P-2:               67371    2099823391    510197760       616193

Application 2670781 resources: utime 0, stime 0

real    0m19.724s
user    0m0.148s
sys     0m0.076s
roche@jaguarpf-login1:/tmp/work/roche/joule-q4>
```

Machine Events Are Useful But Cannot Tell Whole Story

This problem completed execution successfully from the application software perspective.

There is a clear problem in the performance.



1. Form group G1 from MPI_COMM_WORLD
2. Form group G2 := outliers (feature extraction)
3. Form group G3 = G1 \ G2 and COMM3 (work group and communicator)

We have to be smart and aware too

Blocking

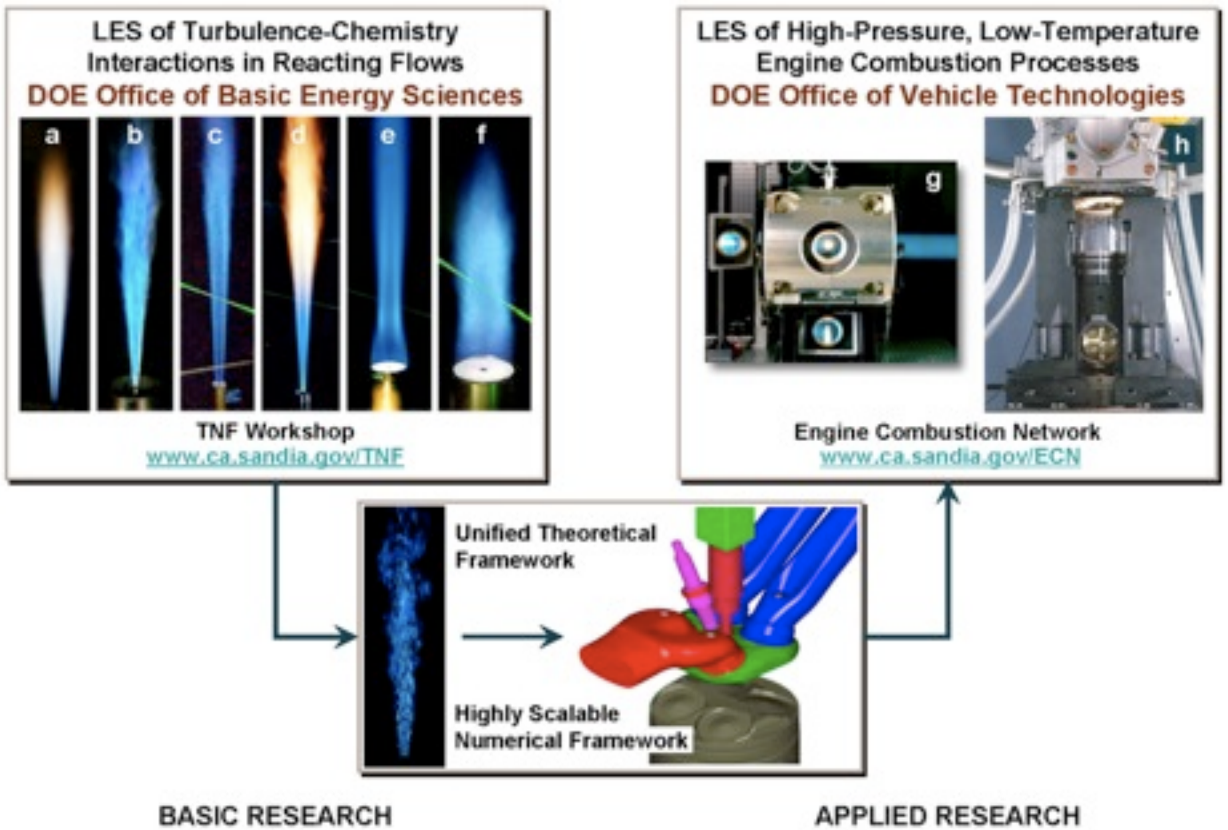
```
if ( ip % 2 )
  { /* BLOCKING */
  MPI_Send( sbf , n , MPI_DOUBLE , ngh[ 0 ] , itag , MPI_COMM_WORLD ) ; /* send to left */
  MPI_Recv( rbf , n , MPI_DOUBLE , ngh[ 1 ] , itag , MPI_COMM_WORLD , &mpi_st ) ; /* receive from right */
  MPI_Send( sbf + n , n , MPI_DOUBLE , ngh[ 1 ] , itag , MPI_COMM_WORLD ) ; /* send to right */
  MPI_Recv( rbf + n , n , MPI_DOUBLE , ngh[ 0 ] , itag , MPI_COMM_WORLD , &mpi_st ) ; /* receive from left */
  }
else
  {
  MPI_Recv( rbf , n , MPI_DOUBLE , ngh[ 1 ] , itag , MPI_COMM_WORLD , &mpi_st ) ; /* receive from right */
  MPI_Send( sbf , n , MPI_DOUBLE , ngh[ 0 ] , itag , MPI_COMM_WORLD ) ; /* send to left */
  MPI_Recv( rbf + n , n , MPI_DOUBLE , ngh[ 0 ] , itag , MPI_COMM_WORLD , &mpi_st ) ; /* receive from left */
  MPI_Send( sbf + n , n , MPI_DOUBLE , ngh[ 1 ] , itag , MPI_COMM_WORLD ) ; /* send to right */
  }
```

Non-Blocking

```
  { /* ASYNCHRONOUS  */

    MPI_Isend( sbf , n , MPI_DOUBLE , ngh[ 0 ] , itag , MPI_COMM_WORLD , r ) ; /* send to the left */
    MPI_Isend( sbf + n , n , MPI_DOUBLE , ngh[ 1 ] , itag , MPI_COMM_WORLD , r + 1 ) ; /* send to the right */
    MPI_Irecv( rbf , n , MPI_DOUBLE, ngh[ 1 ] , itag , MPI_COMM_WORLD , r + 2 ) ; /* receive from the right */
    MPI_Irecv( rbf + n , n , MPI_DOUBLE , ngh[ 0 ] , itag , MPI_COMM_WORLD , r + 3 ) ; /* receive from the left */
    MPI_Waitall( 4 , r , _st ) ;
  }
```

nn exchanges > 2X performance gain, same results!

RAPTOR: Large Eddy Simulation of turbulent, chemically reacting, multiphase flows w/ Joe Oeffelein (SNL) et al



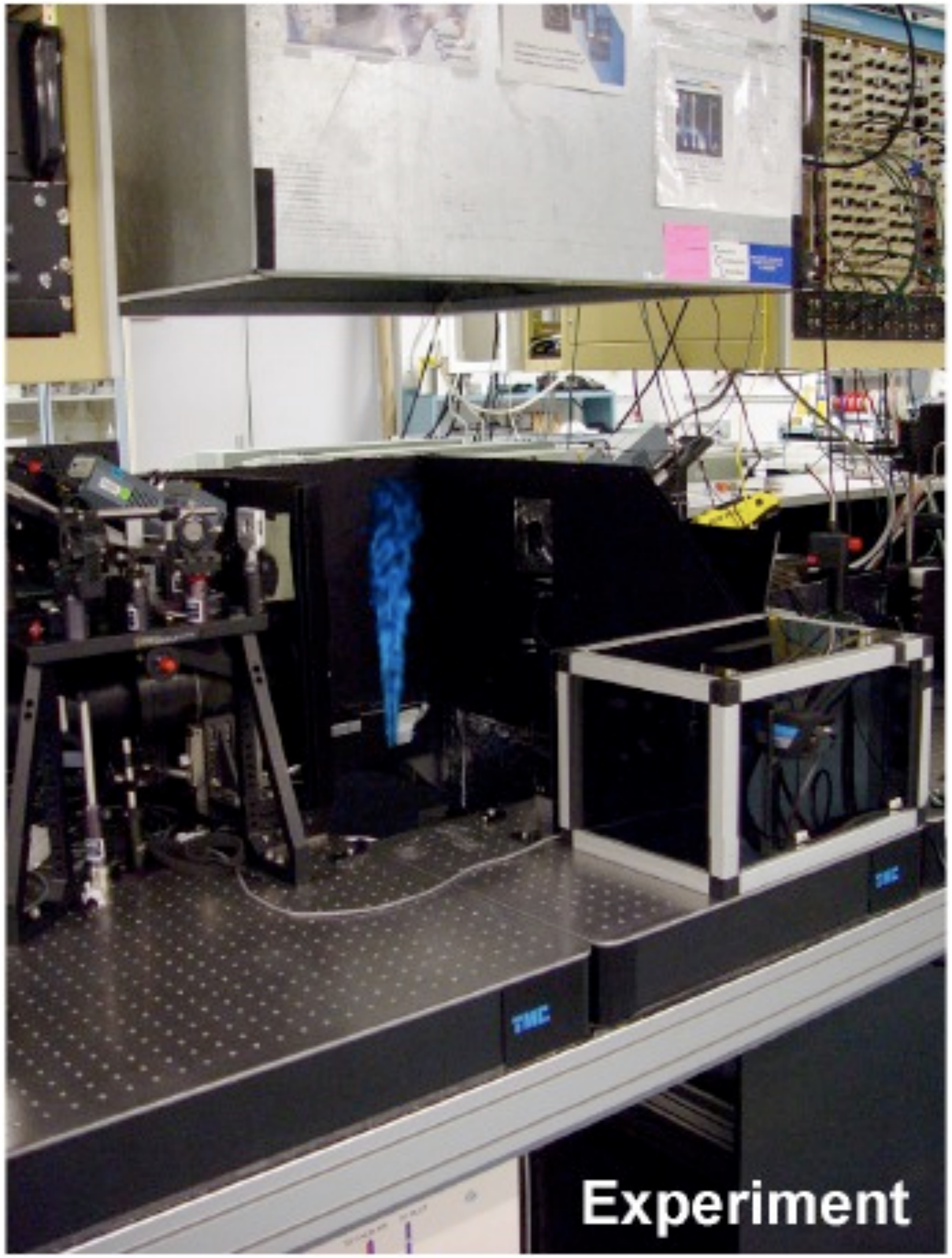
- Fully coupled conservation equations of mass, momentum, total-energy, and species for a chemically reacting flow system (gas or liquid) in complex geometries
  - ✳ Detailed chemistry, thermodynamics, & transport processes at the molecular level and uses detailed chemical mechanisms
  - ✳ Generalized subgrid-scale model framework
  - ✳ Spray combustion processes and multiphase flows using a Lagrangian-Eulerian formulation
- Temporal integration scheme employs an all Mach number formulation using dual-time stepping with generalized preconditioning
  - ✳ Fourth-order accurate in time and provides a fully implicit solution using a fully explicit (highly-scalable) multistage scheme in pseudo-time
- Non-dissipative spatial scheme that is discretely conservative, with staggered, finite-volume differencing stencils
  - ✳ Formulated in generalized curvilinear coordinates with a general R-refinement adaptive mesh (AMR) capability.

Software Implementation

- Distributed multi-block domain decomposition with a generalized connectivity scheme
- Parallelism implemented via MPI and the Single-Program–Multiple-Data model
- Generalized hexahedral cells
- Fully modular, self-contained, and written in ANSI standard Fortran 90
- Extensively validated over last 16 years

- How can simulation "bridge the gap" between basic research and conditions of interest in typical applications?
  - Focus: application of LES models to low-temperature, high-pressure IC-engines
  - Establish high-fidelity computational benchmarks that match geometry and operating conditions of key target experiments using a single unified theoretical-numerical framework
  - Establish a scientific foundation for advanced model development
- Understanding and applying Reynolds number (Re) scaling in combustion modeling is crucial for simulation is to affect engine design
  - Focus on flames studied in the Reacting Flow Research Program at SNL – in particular passive scalar mixing – in a baseline flame (DLR-A experiment) configuration
  - Challenge: most data at Re~10^4 or less; IC engines typically run at Re~10^5 or greater
- Can reliable Re scaling relationships for turbulent flame dynamics and scaling mixing processes be devised appropriately?
  - Pushes mesh resolution up hence a weak scale driver
- Perform a series of weak scaling studies to demonstrate effects of increasing Re (starting from 15.2K) on scalar mixing dynamics
  - These benchmarks provide a direct one-to-one correspondence between measured and modeled results at conditions unattainable using DNS - simulations represent the fully coupled dynamic behavior of a reacting flow with detailed chemistry and realistic levels of turbulence.

RAPTOR Benchmark Motivation



1. study the effects of LES grid resolution on scalar-mixing processes

2. understand the relationship between the grid spacing and the measured turbulence length scales from a companion set of experimental data (DLR-A, shown here)

3. study the effects of increasing jet Reynolds number on the dynamics of turbulent scalar-mixing

DLR-A Flame: $Re_d = 15{,}200$

Fuel: 22.1% CH_4, 33.2% H_2, 44.7% N_2

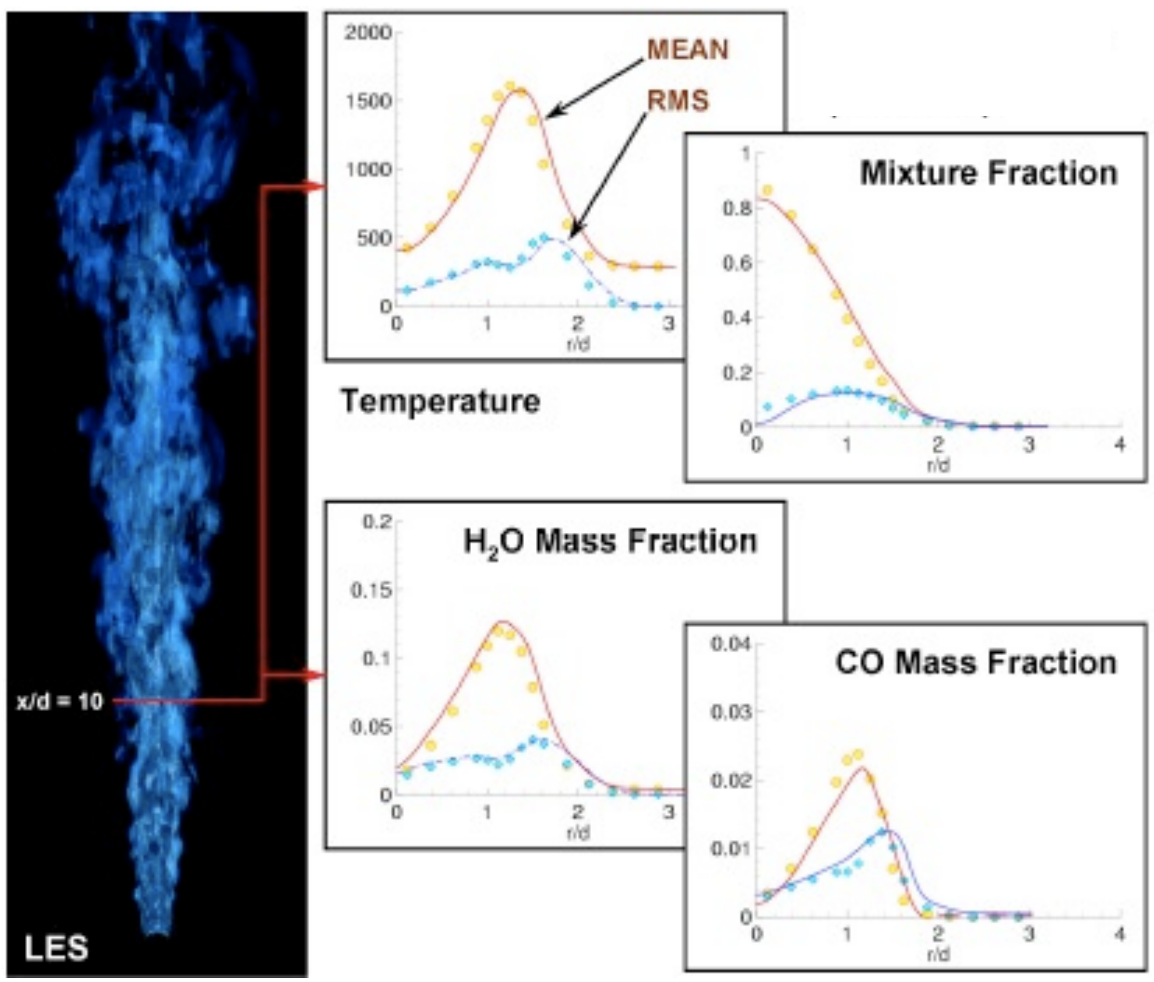
Coflow: 99.2% Air, 0.8% H_2O

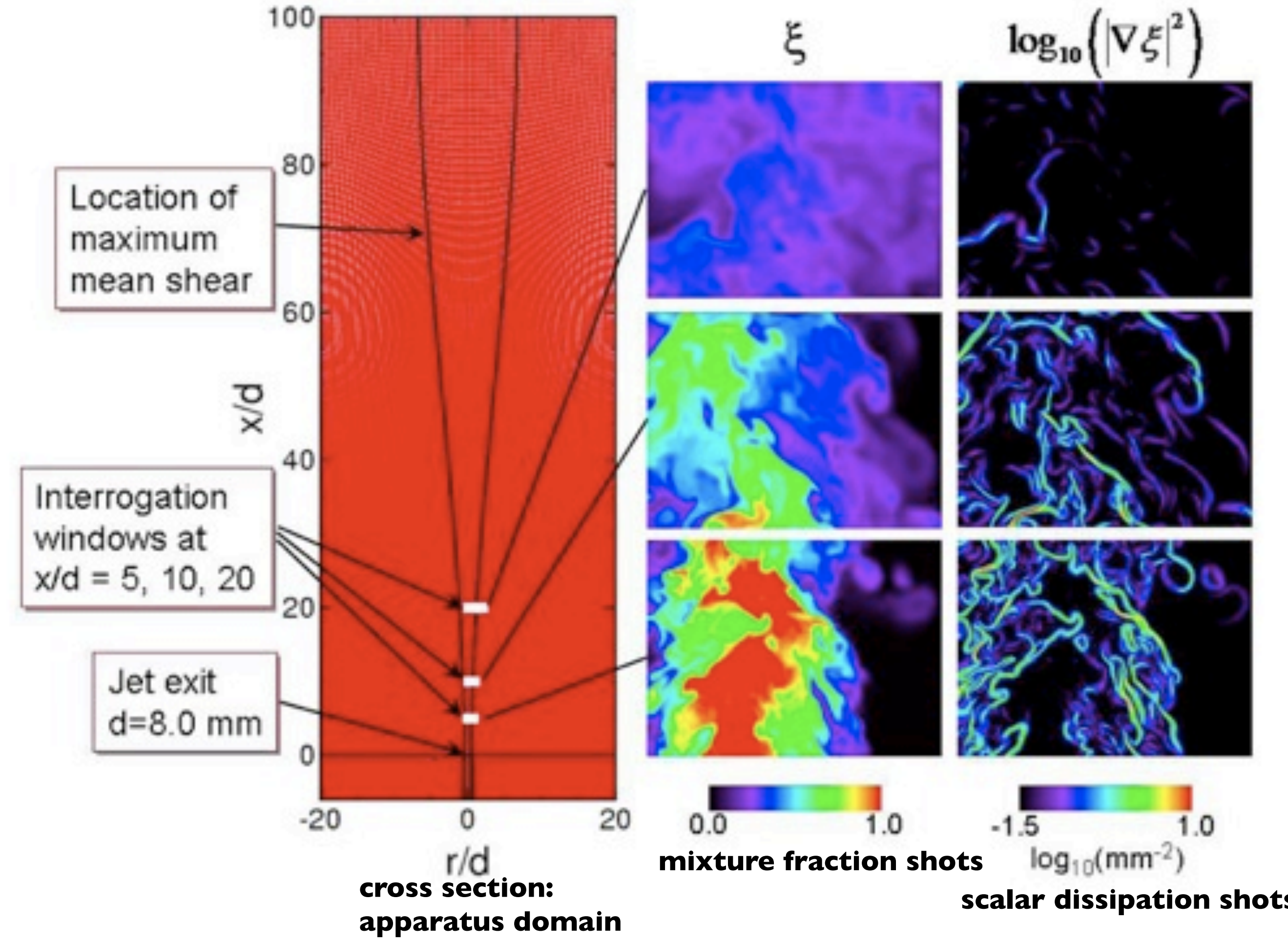
Detailed Chemistry and Transport: 12-Step Mechanism (J.-Y. Chen, UC Berkeley)

RAPTOR Benchmark Configuration

Grid Number	Total Cells	Δt (Re_d = 15,200)
1	1,285,632	1.00 µs
2	10,285,056	0.50 µs
3	82,280,448	0.25 µs

50 physical time steps per grid





Domain: entire burner geometry (inside the jet nozzle and the outer co-flow) + downstream space around burner
Inner nozzle diameter : 8.0 mm
Outer nozzle : surface is tapered to a sharp edge at the burner exit
Specifics: 110 inner jet diameters in the axial direction (88cm) x 40 jet diameters in the radial direction (32 cm)

RAPTOR: Performance Enhancements

- **Halo exchanges are nearest neighbor only**
  - ✴ Initial configuration: send/receive calls in pairs corresponding to each neighbor
  - ✴ Fix:
    - prepost all receives as the first operation in the routine (if buffer available)
    - post the sends as soon as the data is available
    - postpone the waits on send operations until the end of the routine. Non-blocking sends and receives are used throughout
    - Interleave computation to give more breathing room for communication
- **Removal of several unnecessary MPI barriers**
- **Convergence of the dual time integrator**
  - ✴global MPI_allreduce for computing the error norm each iteration
    - use the fact that the number of pseudo-time iterations for convergence does not vary much between consecutive time-steps
    - assign a static variable X to the last pseudo-time step in which convergence was achieved in the previous physical time-step and wait X -1 pseudo-timesteps before computing expensive convergence check

China Grabs Supercomputing Leadership Spot in Latest Ranking of World's Top 500 Supercomputers

Thu, 2010-11-11 22:42

MANNHEIM, Germany; BERKELEY, Calif.; and KNOXVILLE, Tenn.—The 36th edition of the closely watched TOP500 list of the world's most powerful supercomputers confirms the rumored takeover of the top spot by the Chinese Tianhe-1A system at the National Supercomputer Center in Tianjin, achieving a performance level of 2.57 petaflop/s (quadrillions of calculations per second).

News of the Chinese system's performance emerged in late October. As a result, the former number one system — the Cray XT5 "Jaguar" system at the U.S. Department of Energy's (DOE) Oak Ridge Leadership Computing Facility in Tennessee — is now ranked in second place. Jaguar achieved 1.75 petaflop/s running Linpack, the TOP500 benchmark application.

Third place is now held by a Chinese system called Nebulae, which was also knocked down one spot from the June 2010 TOP500 list with the appearance of Tianhe-1A. Located at the National Supercomputing Centre in Shenzhen, Nebulae performed at 1.27 petaflop/s.

Tsubame 2.0 at the Tokyo Institute of Technology is number four; having achieved a performance of 1.19 petaflop/s. Tsubame is the only Japanese machine in the TOP10.

At number five is Hopper, a Cray XE6 system at DOE's National Energy Research Scientific Computing (NERSC) Center in California. Hopper just broke the petaflop/s barrier with 1.05 petaflop/s, making it the second most powerful system in the U.S. and only the third U.S. machine to achive petaflop/s performance.

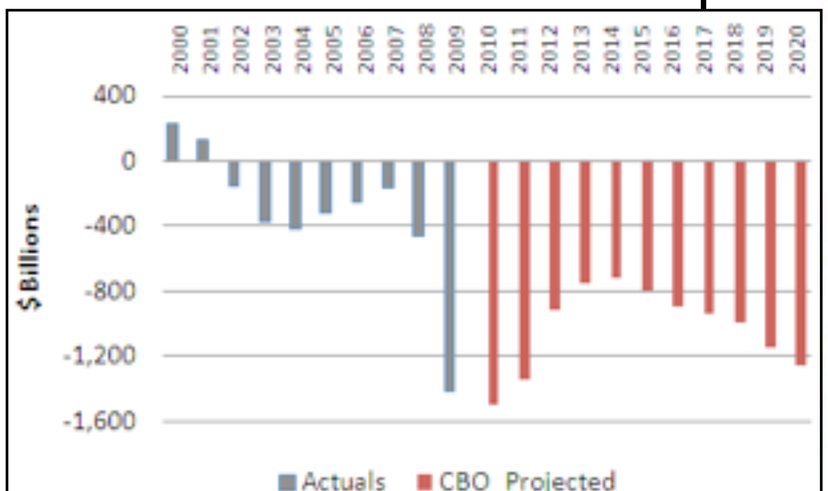
President Obama's FY12 Budget Proposal

• $126M to DOE for next-generation supercomputing
($91 million in SC and $36 million in NNSA)

• Federal budget explicitly mentions "exascale"

• Development of exascale system estimated in 2018-2020 time frame, ***contingent on development of software systems that can utilize ~100 million cores***

$$$$



Department of Energy
(In millions of dollars)

	Actual 2010	Estimate 2011	Estimate 2012
Spending			
Discretionary Budget Authority:			
National Defense:			
National Nuclear Security Administration	9,881		11,783
Cancellation of unobligated balances	—		–70
Other Defense Activities	847		859
Energy Resources	4,445		5,697
Science	4,964		5,416
Environmental Management	6,459		6,130
Corporate Management	256		171
Power Marketing Administrations	150		86
Offsetting receipts	–508		–525
Total, Discretionary budget authority	26,494	28,353	29,547

DOE SC (units of $1K)

	FY 2010 Current Approp.	FY 2011 President's Request	FY 2011 Full Year CR	FY 2012 President's Request	FY 2012 vs. FY 2010	
Advanced Scientific Computing Research	383,199	426,000	394,000	465,600	+82,401	+21.5%
Basic Energy Sciences	1,598,968	1,835,000	1,636,500	1,985,000	+386,032	+24.1%
Biological and Environmental Research	588,031	626,900	604,182	717,900	+129,869	+22.1%
Fusion Energy Sciences	417,650	380,000	426,000	399,700	-17,950	-4.3%
High Energy Physics	790,811	829,000	810,483	797,200	+6,389	+0.8%
Nuclear Physics	522,460	562,000	535,000	605,300	+82,840	+15.9%
Workforce Development for Teachers and Scientists	20,678	35,600	20,678	35,600	+14,922	+72.2%
Science Laboratories Infrastructure	127,600	126,000	127,600	111,800	-15,800	-12.4%
Safeguards and Security	83,000	86,500	83,000	83,900	+900	+1.1%
Science Program Direction	189,377	214,437	189,377	216,863	+27,486	+14.5%
Subtotal, Office of Science	**4,721,774**	**5,121,437**	**4,826,820**	**5,418,863**	**+697,089**	+14.8%
Small Business Innovation Research/ Technology Transfer (SBIR/STTR) (SC portion)	107,352	...	...	...	-107,352	-100.0%
Congressionally-directed projects	74,737	...	...	...	-74,737	-100.0%
Undistributed	...	...	76,890	...	...	...
Use of prior year balances	-153	...	...	-2,749	-2,596	-1,696.7%
Subtotal, Office of Science	**4,903,710**	**5,121,437**	**4,903,710**	**5,416,114**	**+512,404**	+10.4%
SBIR/STTR (transfer from other DOE programs)	60,177	...	...	...	-60,177	-100.0%
Total, Office of Science	**4,963,887**	**5,121,437**	**4,903,710**	**5,416,114**	**+452,227**	+9.1%

NNSA
+19.25%

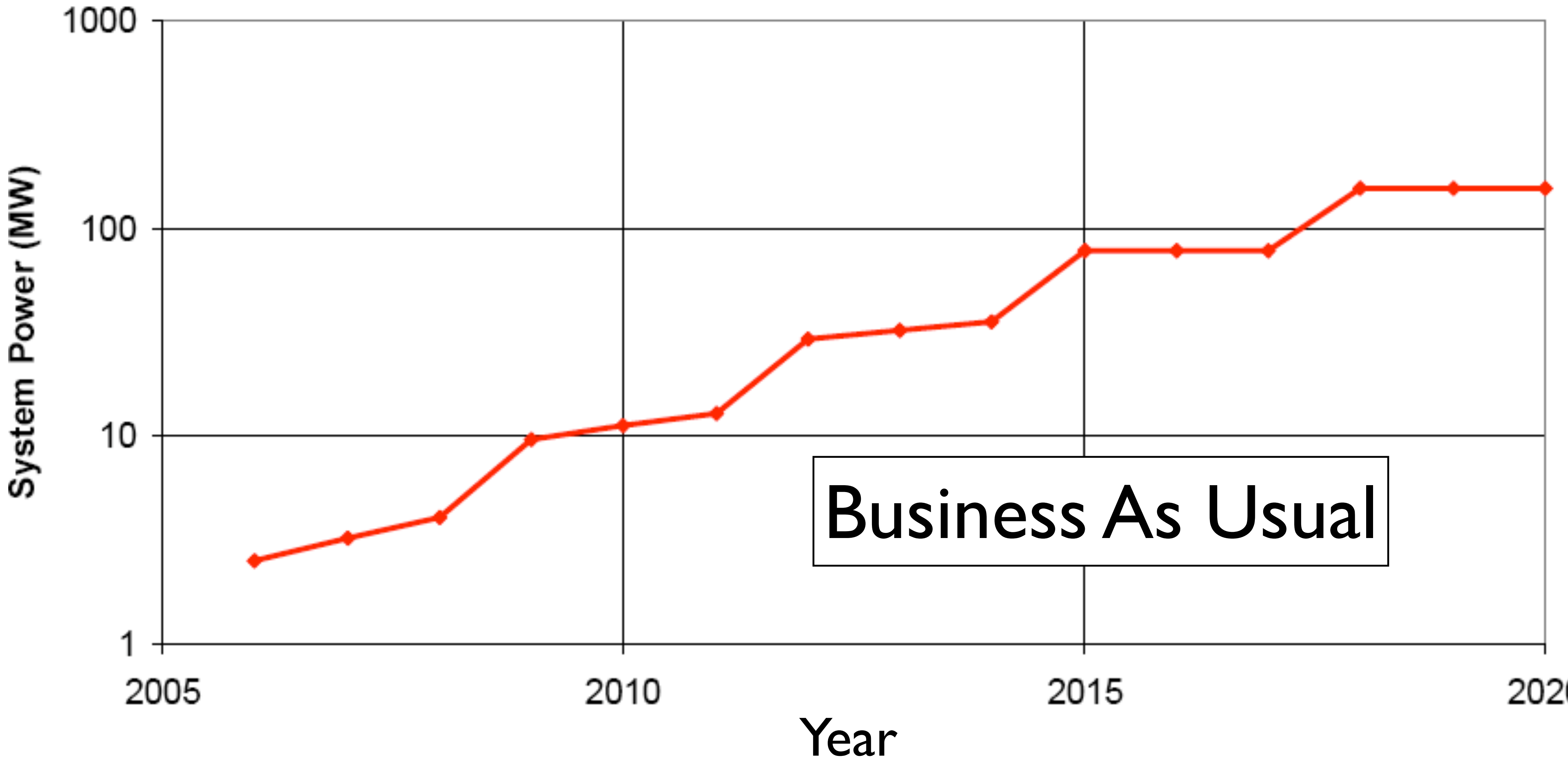
ASCR

- At $1M per MW, energy costs are substantial
- 1 Pf in 2010 ~ 3 MW
- 1 Ef in 2018 at 200 MW with "usual" scaling

NNSA

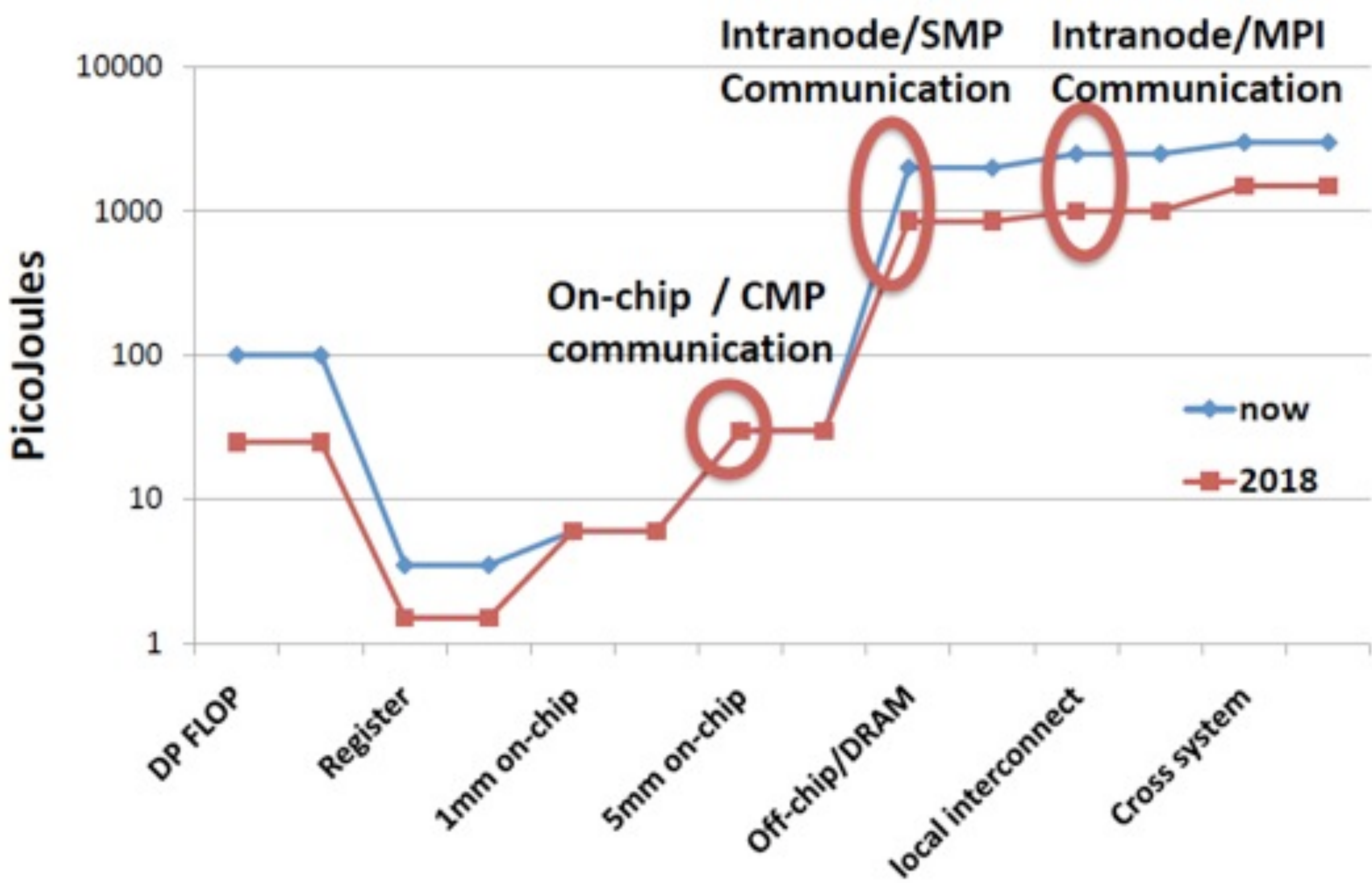
- Power constraints using current technology are unaffordable
- 20 Pf Sequoia requires ~ 10MW to operate
- 1Ef requires ~500MW with current technologies

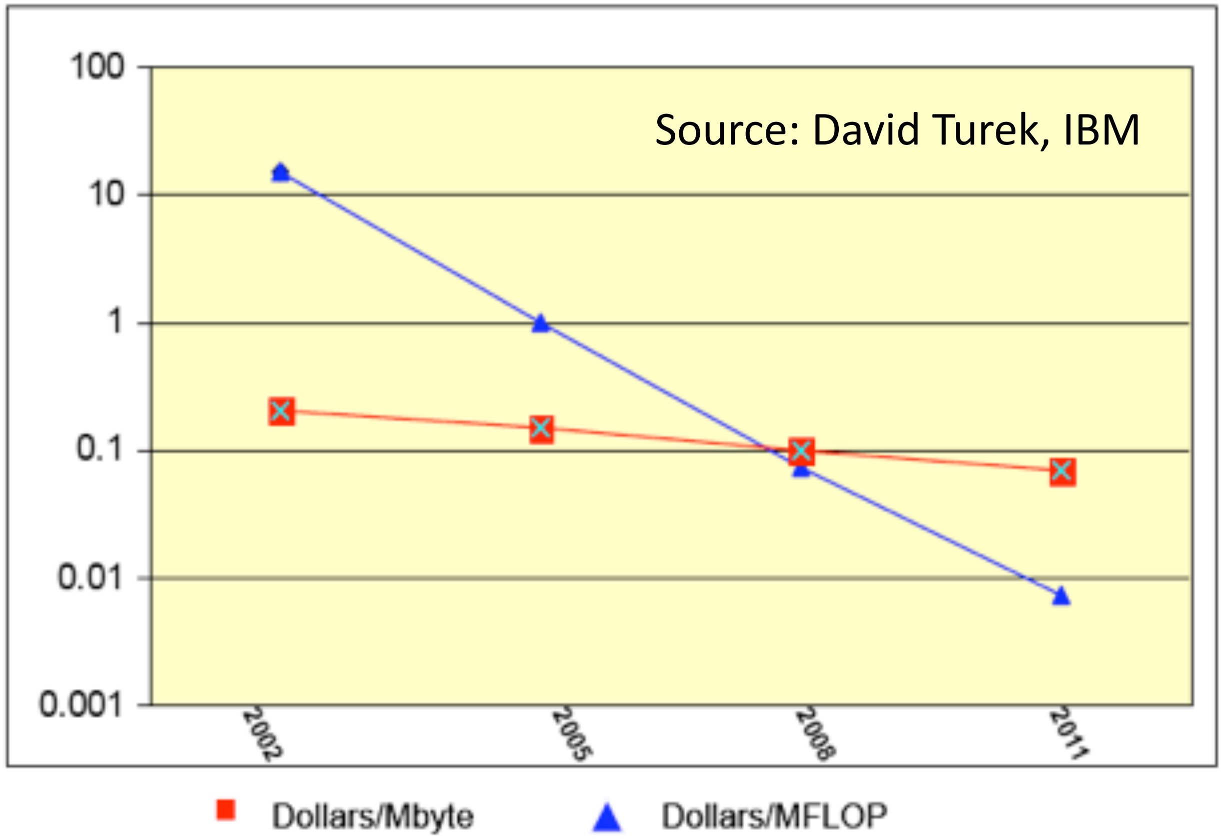
1 Exaflop in 2018 at 20 MW is target!



Exascale Table -guess work?

	2010	2018	Factor Change
System peak	2 Pf/s	1 Ef/s	500
Power	6 MW	20 MW	3
System Memory	0.3 PB	10 PB	33
Node Performance	0.125 Gf/s	10 Tf/s	80
Node Memory BW	25 GB/s	400 GB/s	16
Node Concurrency	12 cpus	1,000 cpus	83
Interconnect BW	1.5 GB/s	50 GB/s	33
System Size (nodes)	20 K nodes	1 M nodes	50
Total Concurrency	225 K	1 B	4,444
Storage	15 PB	300 PB	20
Input/Output bandwidth	0.2 TB/s	20 TB/s	100

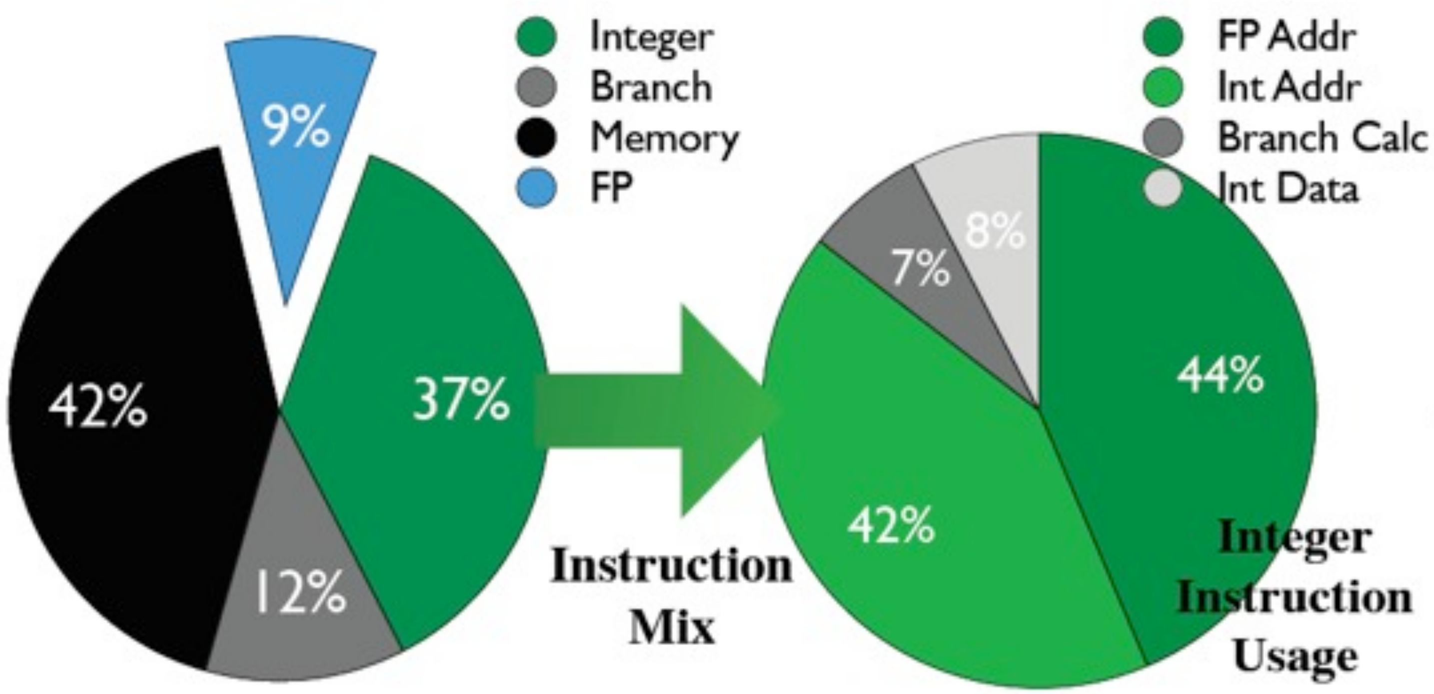


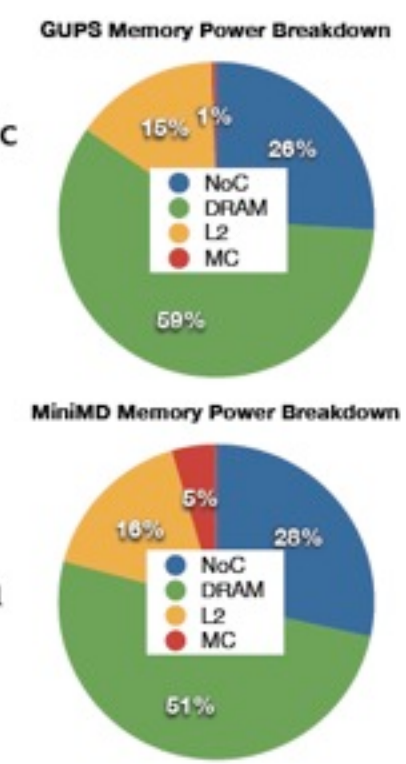


The cost to sense, collect, generate and calculate data is declining much faster than the cost to access, manage and store it

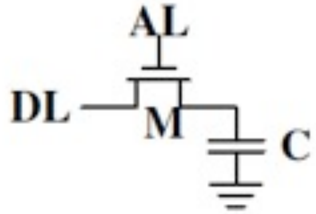
Data movement (DRAM) dominates:

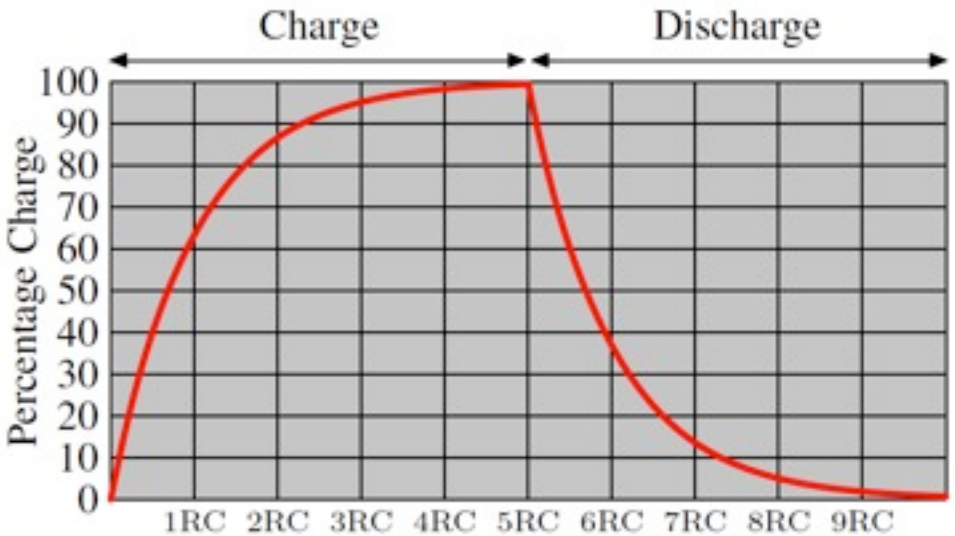
- energy costs
- application performance

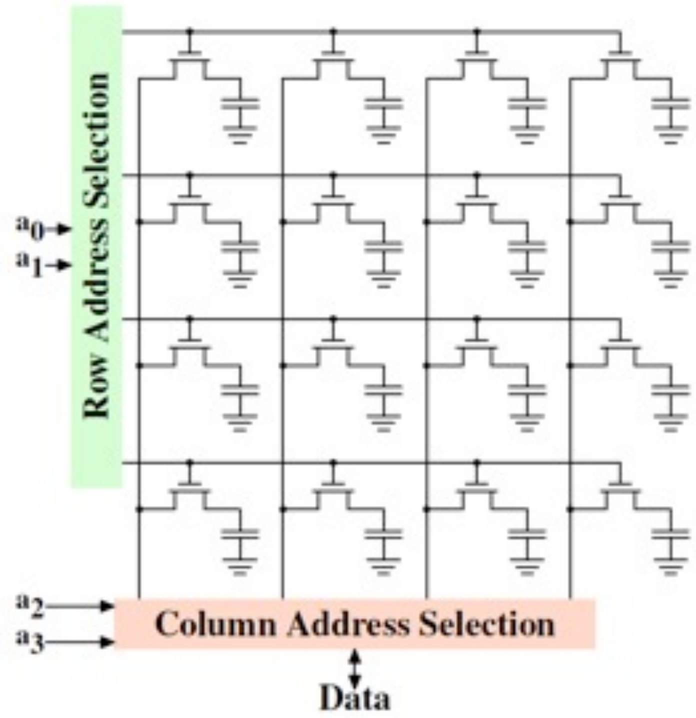


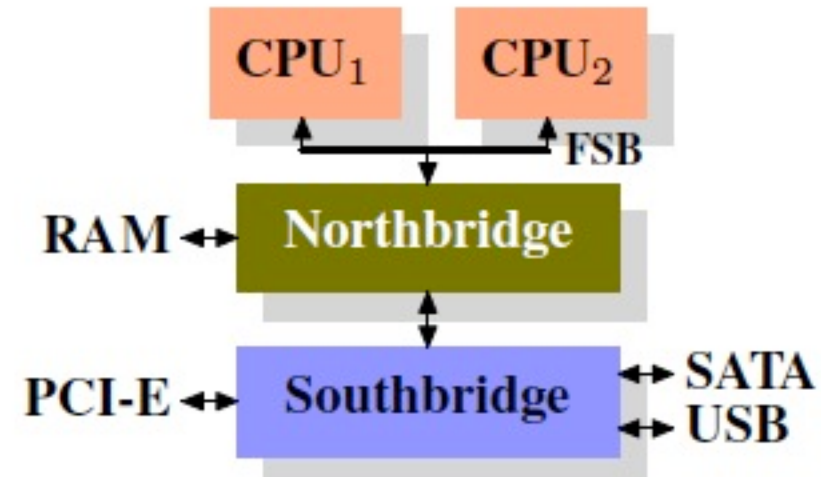


Power = Capacity X Voltage^2 X Frequency



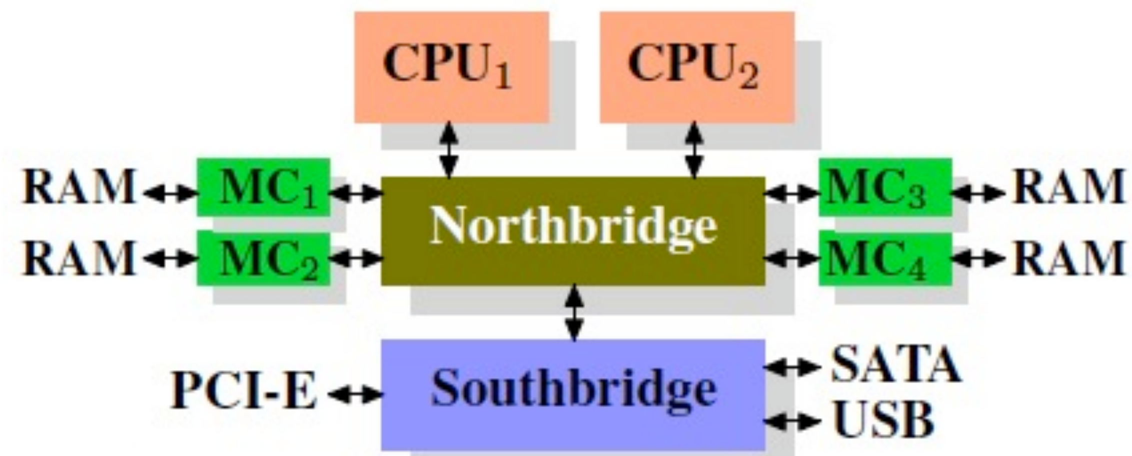






Today's Memories ...

- 10^9 cells
- cell capacitance < femto-farad
- resistance O(tera-ohms)



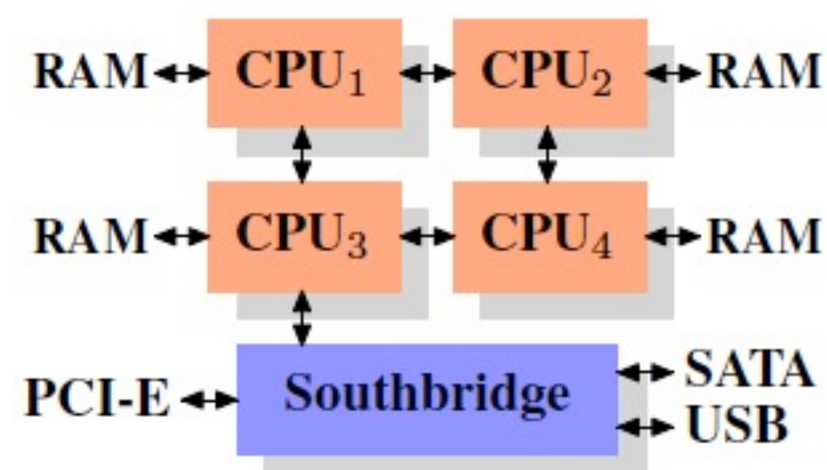
external memory control

Refresh Cycles ~ 64ms

- leakage
- reading drains the charge (read + recharge)

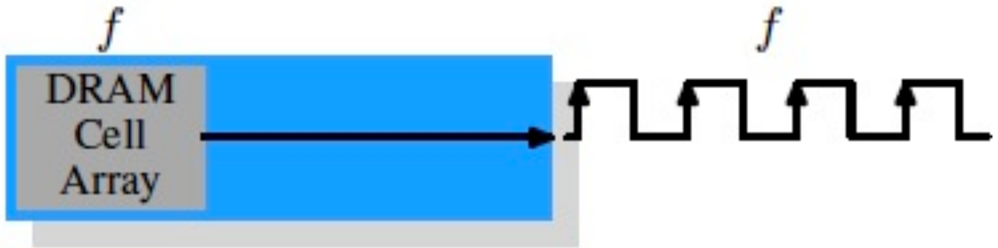
Faster memory::

- lower voltage --> decreases stability,
- increase frequency --> $$$ as arrays get large (i.e. more addressable memory) and voltage is increased to assure stability

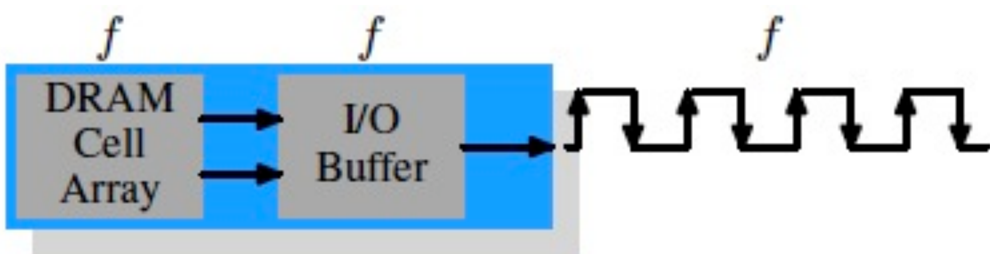


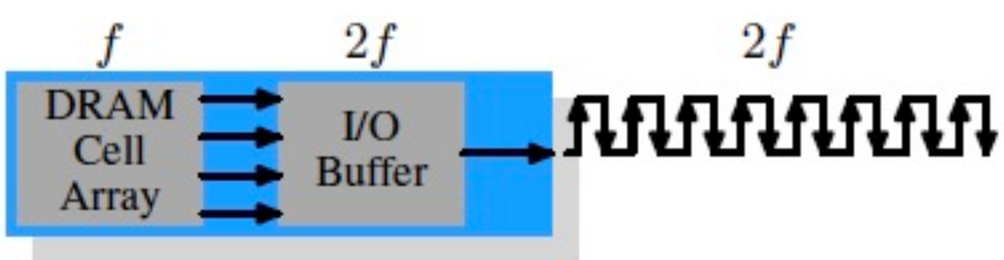
cpu integrated memory control

SDR (PC100) ~
DRAM cell array 100MHz
data transfer rate 100Mbps



DDR (PC1600) ~ moves 2X the data / clock (leading , falling)
add "I/O" buffer (2 bits / data line) adjacent to DRAM cell array
pull two adjacent column cells per access over 2 line data bus
100 MHz X 64 bit / data bus X 2 data bus lines = 1600 MBps





DDR2 (PC6400) ~ moves 4X the data / clock
double the bus frequency --> 2X bandwidth
double "I/O" buffer speed to match the bus
4 bits / clock on 4 line data bus
200MHz array; 400MHz bus; 800MHz FSB (effective freq)
200 MHz X 64 bit / data bus X 4 data bus lines = 6400 MBps
240 PIN addressing @ 1.8V

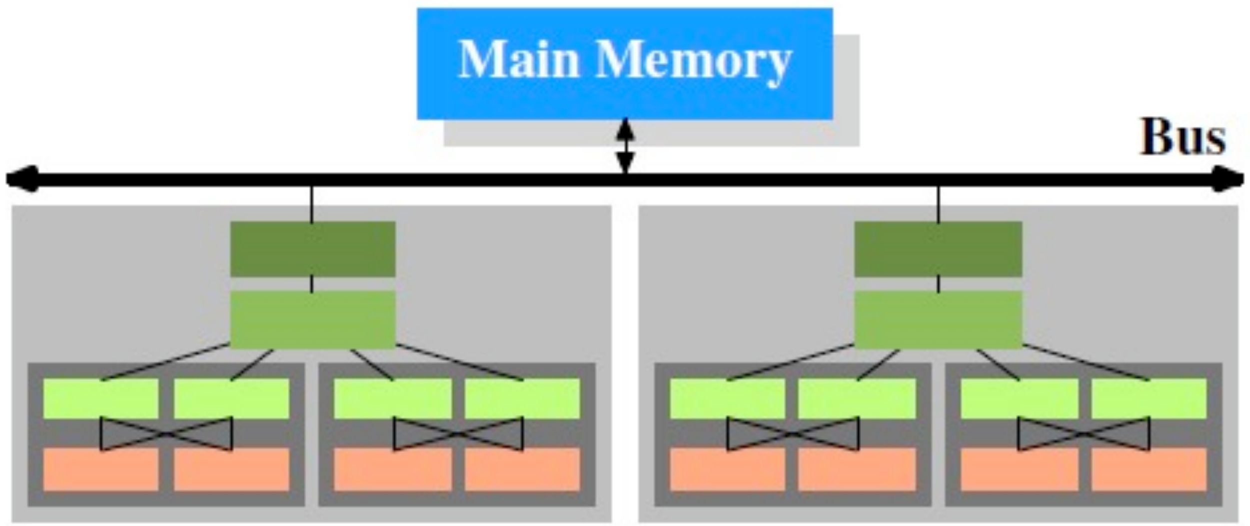
***each stall cycle on the memory bus is > 11 cpu cycles even in the best systems**

Cache

temporal locality
when a referenced resource is referenced again sometime in the near future

spatial locality
the chance of referencing a resource is higher if a resource near it was just referenced

To Where	Cycles
Register	≤ 1
L1d	~ 3
L2	~ 14
Main Memory	~ 240



set-associative dereferencing (the larger the set and CL, the fewer the misses):
tag and data in sets -a set maps to the address of the cache line, a small number of values is cached for the same set value ; the tags for all such sets are compared in parallel
ie 8 sets for L1 and 24 associativity levels for L2 are common;
for 4MB/64B and 8 way set-associativity then 8192 sets (requires 13bit address tag) ; to find if the address is in cache only 8 tags have to be compared!

Coherency:

write-through, if cache line is written to, the processor also writes to main memory (at all times cache and memory are in synche)

write-back, cache line is marked dirty, write back is delayed to when cache line is being evicted

>1 processor core is active (say in SMP) -all processors still have to see the same memory content; have to exchange CL when needed -includes the MC

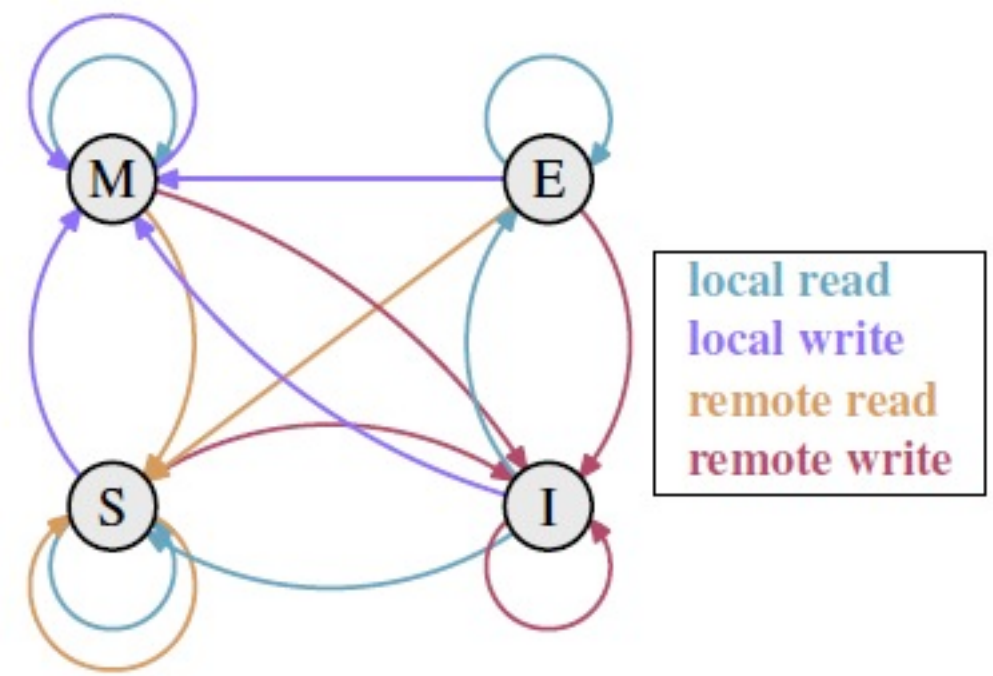
write-combining (ie on graphics cards)

*M*odified, local processor has only copy of data and modifies it

*E*xclusive, CL is not modified and not in another processor's (core) cache

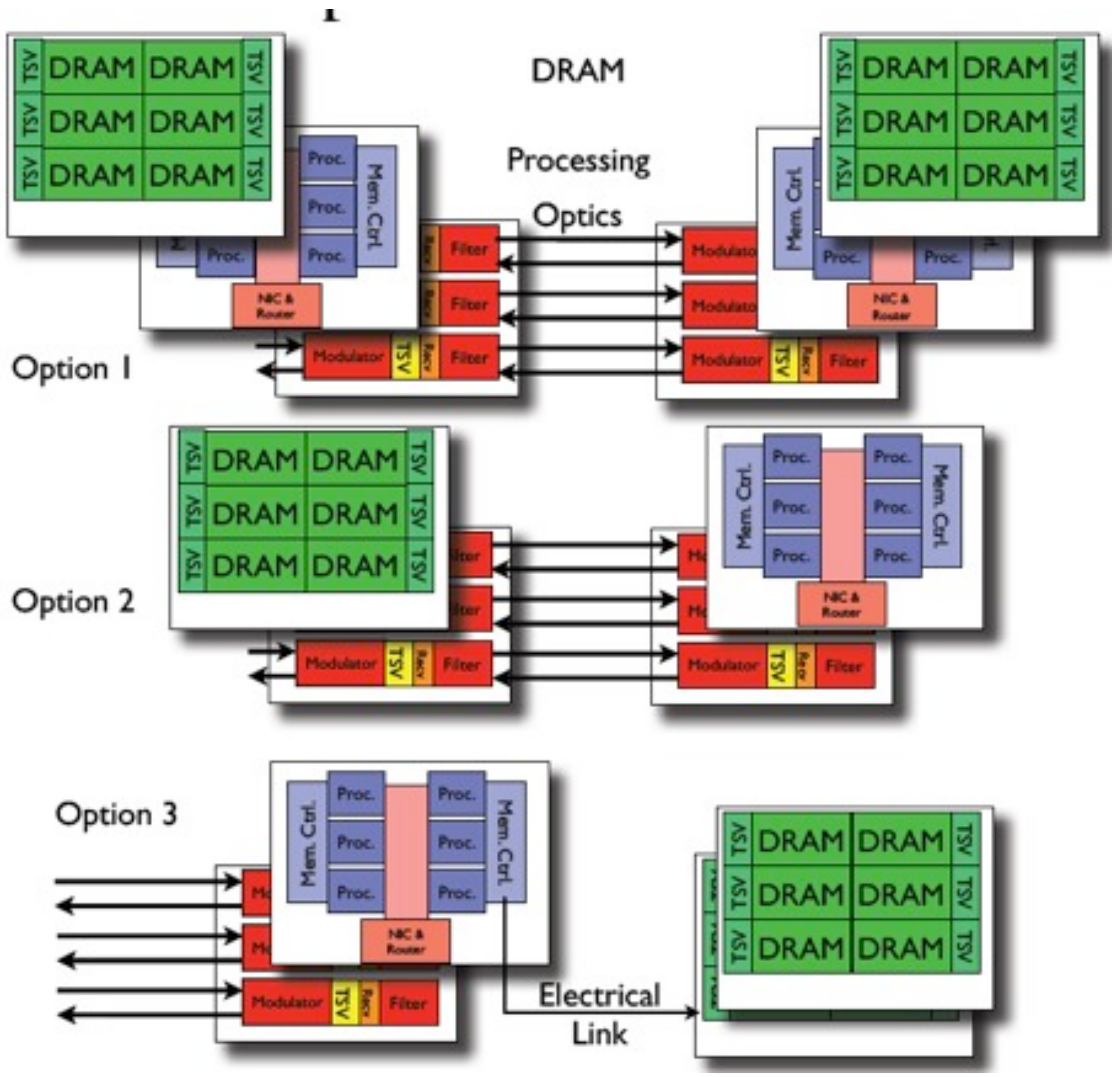
*S*hared, CL not modified -might be in cache somewhere

*I*nvalid, CL is invalid -not used



*other processor's activities are snooped on the address bus

Data Movement Dominates
Advanced Memory Technology to Address the Real Exascale Power Problem (SNL -lead)



study architectures which combine stacked memory, processing, and photonic interconnect.

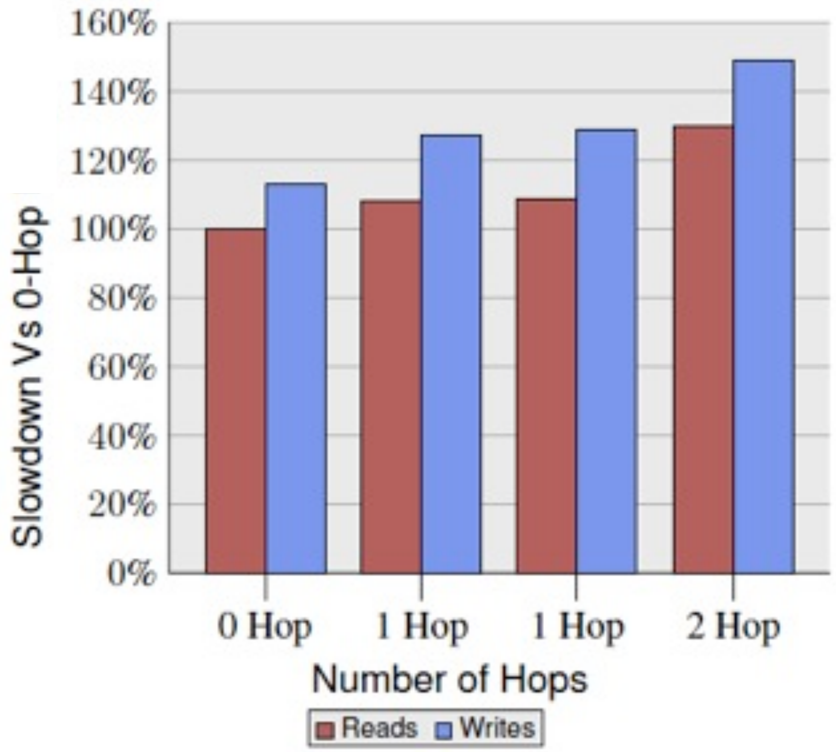
PhoenixSim optical interconnect simulator;
the DRAMsim advanced memory simulator;
Structural Simulation Toolkit (SST),
which will provide processor and I/O models as well as a parallel simulation and power analysis infrastructure.

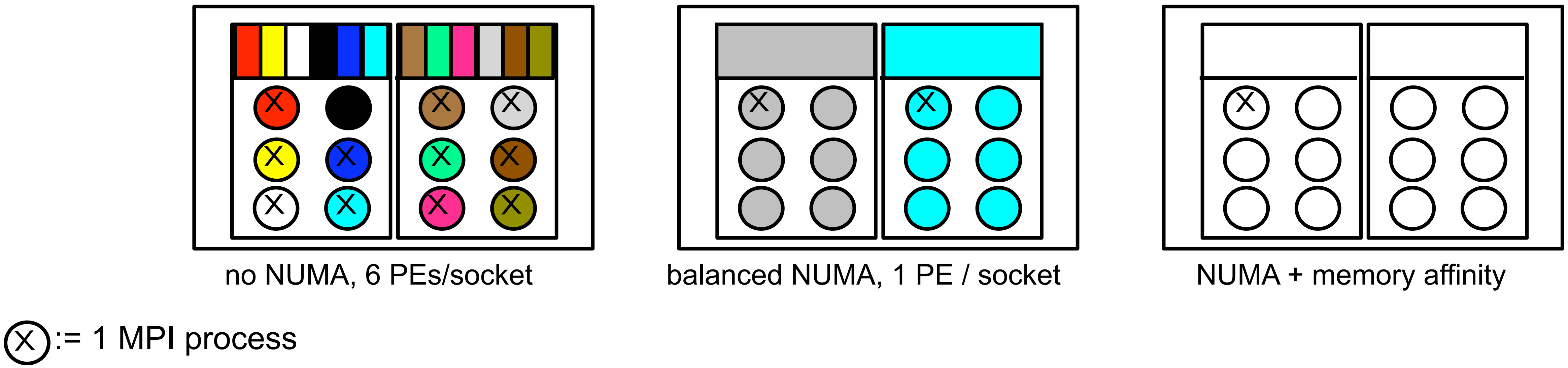
NUMA Node of XT5 --> Multi-core Hybrid Programming Model

- **MPI processes** spawn lightweight processes
- **OpenMP threads**, #include <omp.h> , omp_set_num_threads();
- **POSIX threads**, #include <pthread.h> , pthread_create();

-lsize=12	MPI	LWP	DRAM
aprun -n <1-12>	1 - 12	1	1.33 * 2^30
aprun -n 2 -sn 2 -S 1 -d 6	2	1 - 6	8 * 2^30
aprun -n 1 -N 1 -d 12	1	1 - 12	16 * 2^30

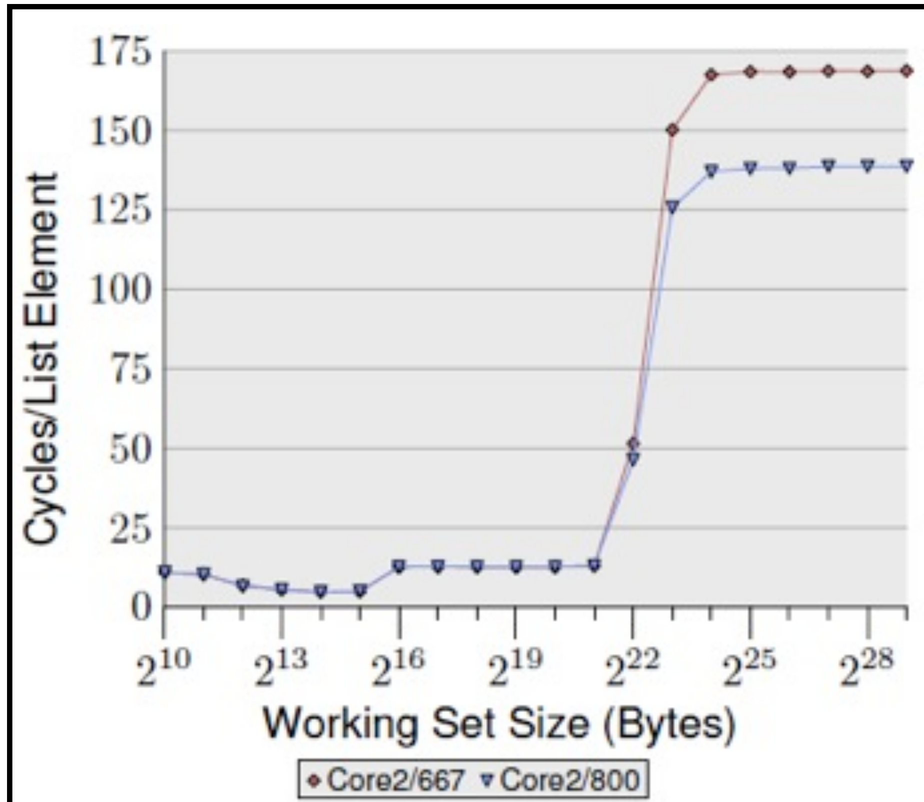
<-S> * <-d> cannot exceed the maximum number of CPUs per NUMA node





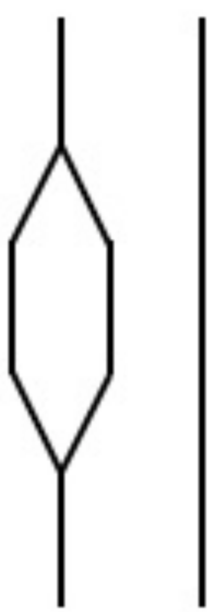
MC / NUMA / SMP

- **threaded**, concurrency, atomicity, bandwidth
  - cache contention
  - memory bandwidth
  - scheduling

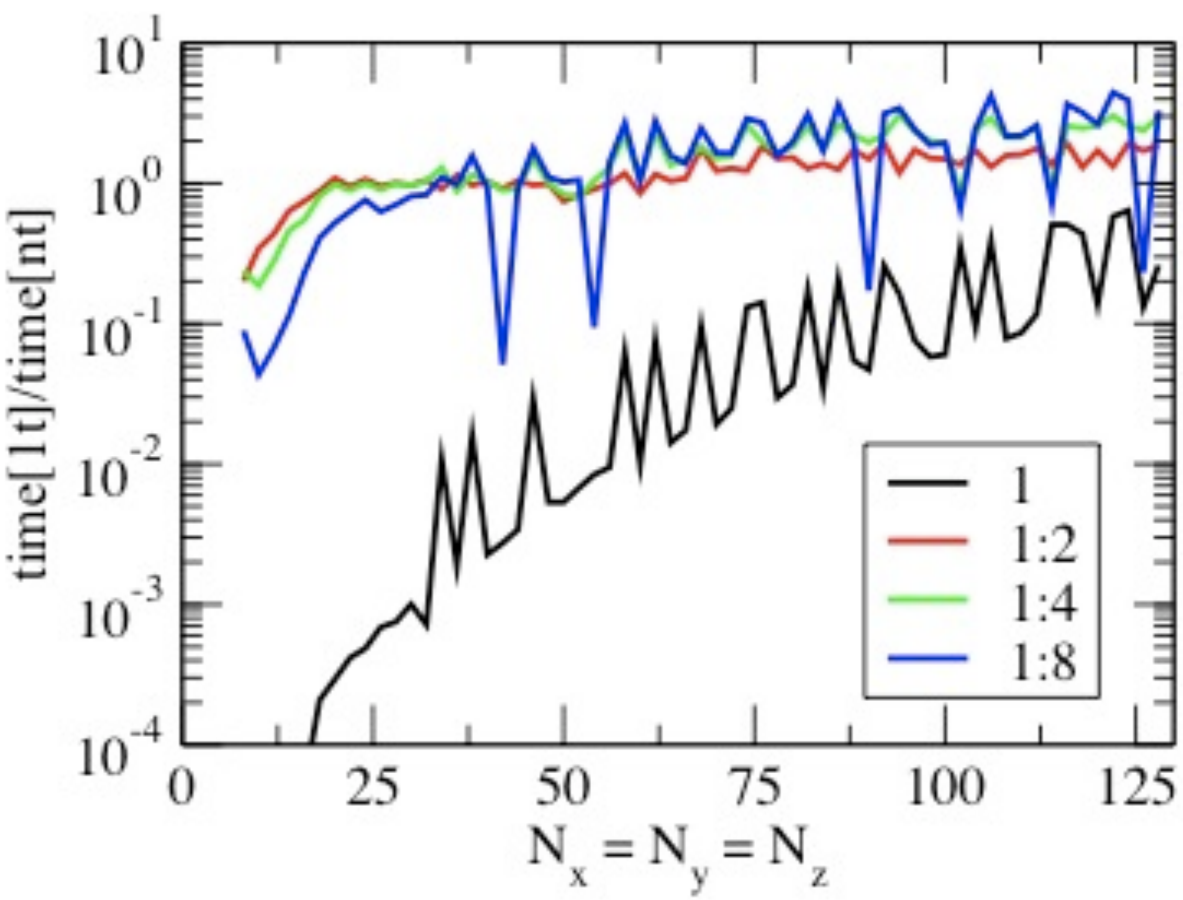


•it helps to make the FSB faster with increasing core count

Fork / Join Overhead

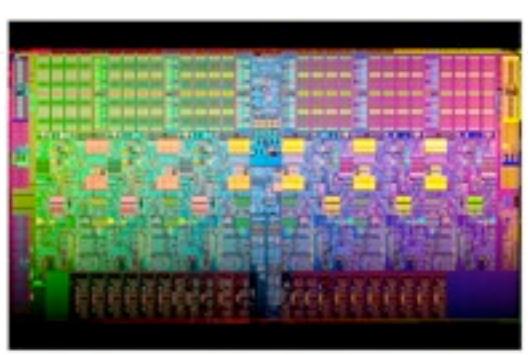


NT	Cycles	L2DCM
1	1959379	69
2	2020818	81
4	2289393	122
6	2366367	146
8	2499159	239



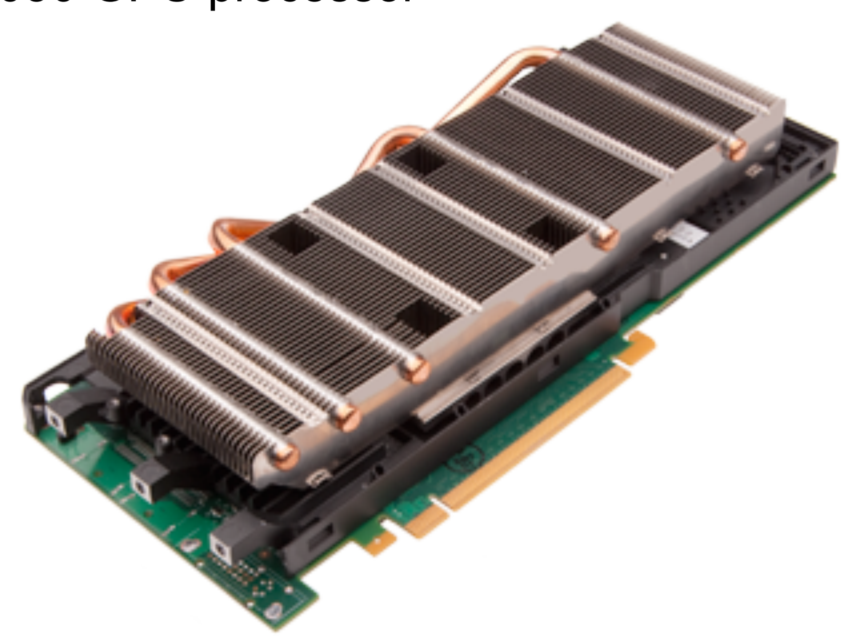
Tianhe-1 Chinese National University of Defense Technology (NUDT) Changsha, Hunan

112 compute cabinets
12 storage cabinets
6 communications cabinets
8 I/O cabinets
Storage := 2PB , Lustre
Addressable memory := 262 TB
86,016 cpu- compute cores
112 cc X 4 frames / cc X 8 blades / frame X 2 nodes / blade X 2 Hex-Core Intel Xeon / node
112 cc X 4 frames / cc X 8 blades / frame X 2 nodes / blade X 1 Nvidia M2050 GPU



Intel Xeon X5670 - 6 core
2.93 GHz clock (11.72GFlps)

Nvidia M2050 GPU processor



1/20th the power consumption and 1/10th the cost

1 Tesla GPUs
Double Precision floating point performance (peak) : 515 Gflops
Single Precision floating point performance (peak) : 1.03 Tflops
Total Dedicated Memory : 3GB GDDR5
Memory Speed : 1.55 GHz
Memory Interface : 384-bit
Memory Bandwidth : 148 GB/sec
System Interface : PCIe x16 Gen2
Software Development Tools : CUDA C/C++/Fortran, OpenCL, DirectCompute Toolkits; NVIDIA Parallel Nsight™ for Visual Studio

Some Buses (Bandwidth) *

PCI := 132 MB/s
AGP 8X := 2,100 MB/s
PCI Express 1x := 250 [500]* MB/s
PCI Express 2x := 500 [1000]* MB/s
PCI Express 4x := 1000 [2000]* MB/s
PCI Express 8x := 2000 [4000]* MB/s
PCI Express 16x := 4000 [8000]* MB/s
PCI Express 32x := 8000 [16000]* MB/s
USB 2.0 (Max Possible) := 60 MB/s
IDE (ATA100) := 100 MB/s
IDE (ATA133) := 133 MB/s
SATA := 150 MB/s
SATA II := 300 MB/s
Gigabit Ethernet := 125 MB/s
IEEE1394B [Firewire 800] := ~100 MB/s

* 2X for both lanes

programming tomorrow? hybrid - heterogeneous pm

Developer Drivers for Linux (260.19.21)
32-bit
64-bit
CUDA Toolkit

- C/C++ compiler
- cuda-gdb debugger
- Visual Profiler
- GPU-accelerated BLAS library
- GPU-accelerated FFT library
- GPU-accelerated Sparse Matrix library
- GPU-accelerated RNG library
- Additional tools and documentation

Linux Getting Started Guide
Release Notes
Release Notes Errata
CUDA C Programming Guide
CUDA C Best Practices Guide
OpenCL Programming Guide
OpenCL Best Practices Guide
OpenCL Implementation Notes
CUDA Reference Manual (pdf)
CUDA Reference Manual (chm)
API Reference
PTX ISA 2.2
CUDA-GDB User Manual
Visual Profiler User Guide
Visual Profiler Release Notes
Fermi Compatibility Guide
Fermi Tuning Guide
CUBLAS User Guide
CUFFT User Guide
CUSPARSE User Guide
CURAND User Guide
CUDA Developer Guide for Optimus Platforms
License

already supported ...
Fedora 13
RedHat Enterprise Linux 5.5
Ubuntu Linux 10.04
RedHat Enterprise Linux 4.8
OpenSUSE 11.2
SUSE Linux Enterprise Desktop 11 SP1

NVIDIA Performance Primitives (NPP) library

CULA: GPU-accelerated LAPACK libraries

CUDA Fortran from PGI

GPU Computing SDK code samples

NVIDIA OpenCL Extensions

Compiler_Options
D3D9 Sharing
D3D10 Sharing
D3D11 Sharing
Device Attribute Query
Pragma Unroll

Programmers will have to ...
Re-Invent Hit-or-Miss Strategies of Today

- **non-temporal writes,** ie don't cache the data writes since it won't be used again soon (i.e. n-tuple initialization)
  - avoids reading cache line before write, avoids wasteful occupation of cache line and time for write (*memset()*); does not evict useful data
  - *sfence()* compiler set barriers
- **loop unrolling** , transposing matrices
- **vectorization**
  - 2,4,8 elements computed at the same time (SIMD) w/ multi-media extensions to ISA
- **reordering** elements so that elements that are used together are stored together -pack CL gaps w/ usable data (i.e. try to access structure elements in the order they are defined in the structure)
- **stack alignment,** as the compiler generates code it actively aligns the stack inserting gaps where needed ... is not necessarily optimal -if statically defined arrays, there are tools that can improve the alignment; separating n-tuples may increase code complexity but improve performance
- **function inlining**, may enable compiler or hand -tuned instruction pipeline optimization (ie dead code elimination or value range propagation) ; especially true if a function is called only once
- **prefetching**, hardware, tries to predict cache misses -with 4K page sizes this is a hard problem and costly penalty if not well predicted; software (*void _mm_prefetch(void *p, enum _mm_hint h)* --*_MM_HINT_NTA* -when data is evicted from L1d -don't write it to higher levels)

Vancouver: Designing a Next-Generation Software Infrastructure for Productive Heterogeneous Exascale Computing -ORNL lead

Programming constructs to span the range from the fine-grain parallelism supported by heterogeneous computing devices, to the large-scale parallelism required for the Exascale.

... *OpenCL can address the former, and the Message Passing Interface (MPI) can address the latter, but there are few languages or software tools that address both levels simultaneously*

fox.xstack.org

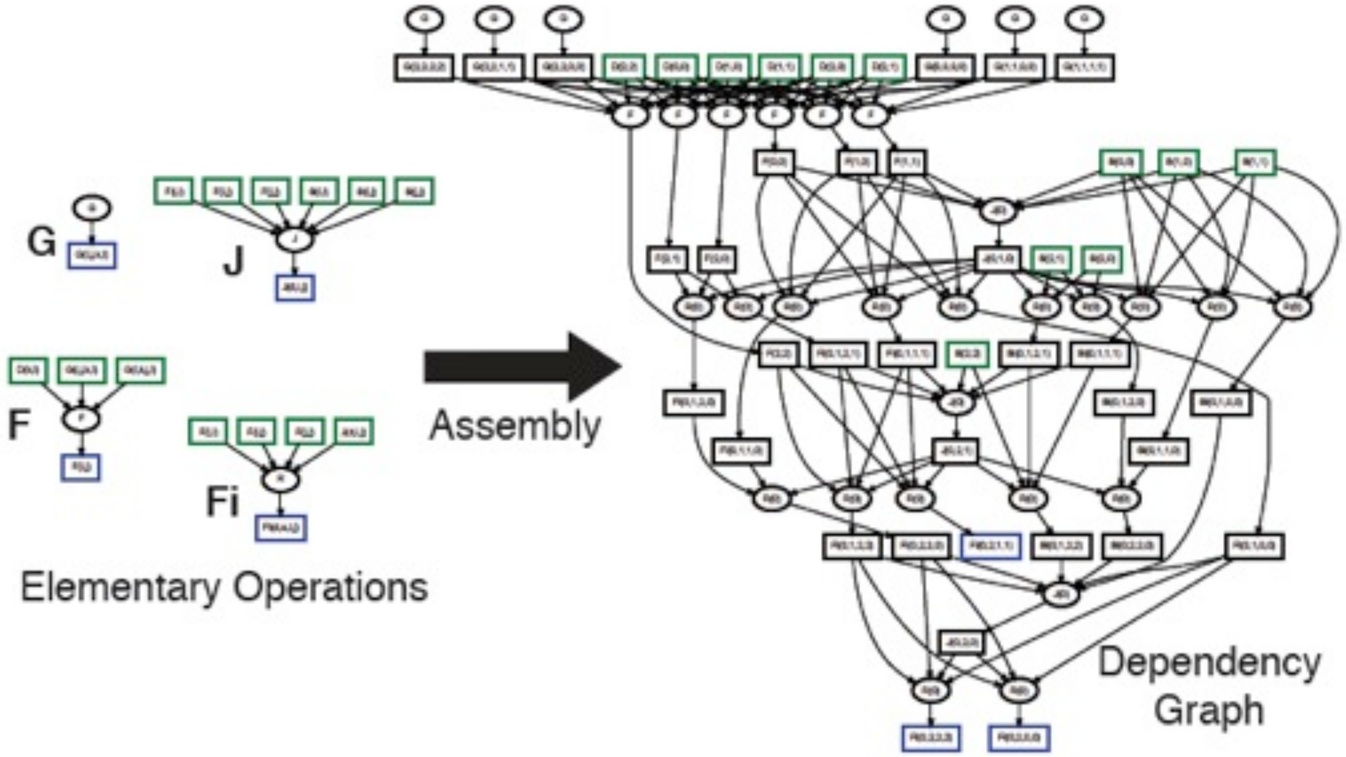
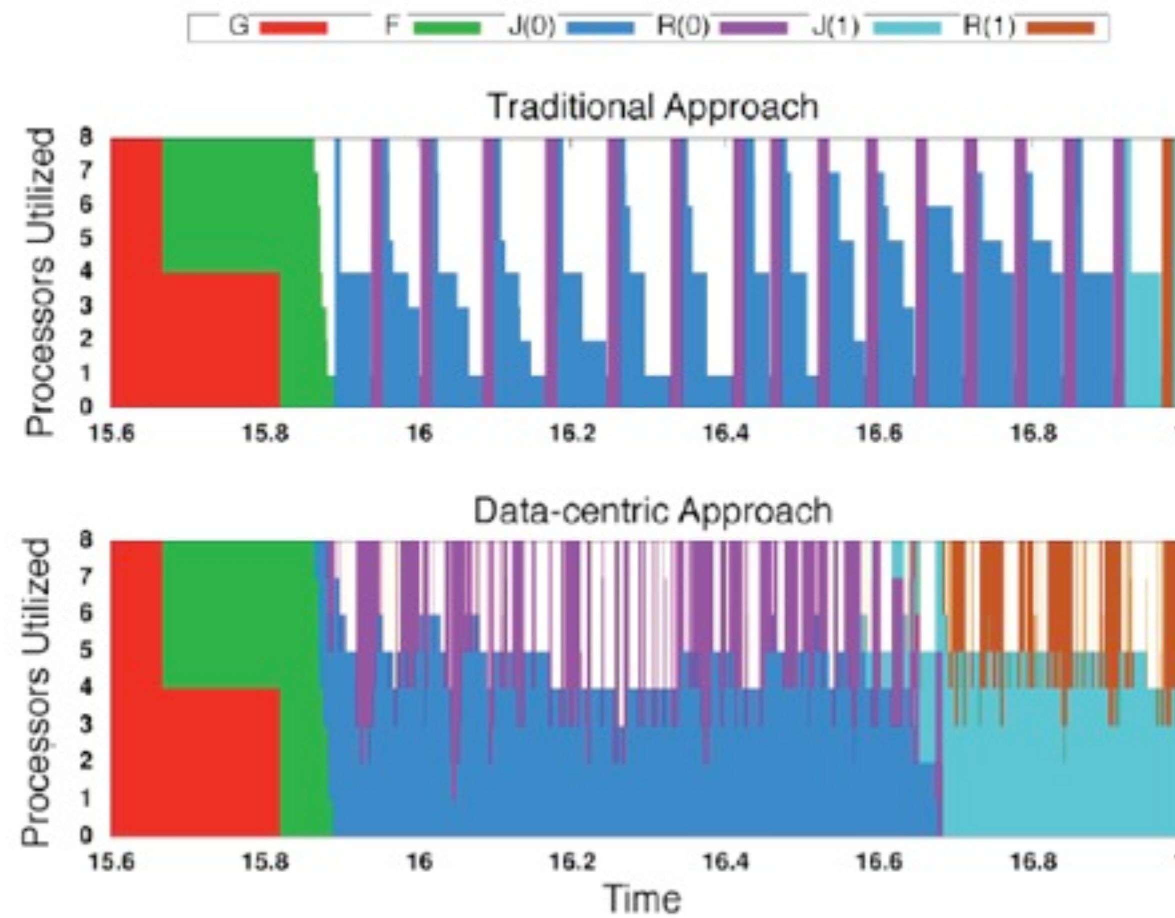
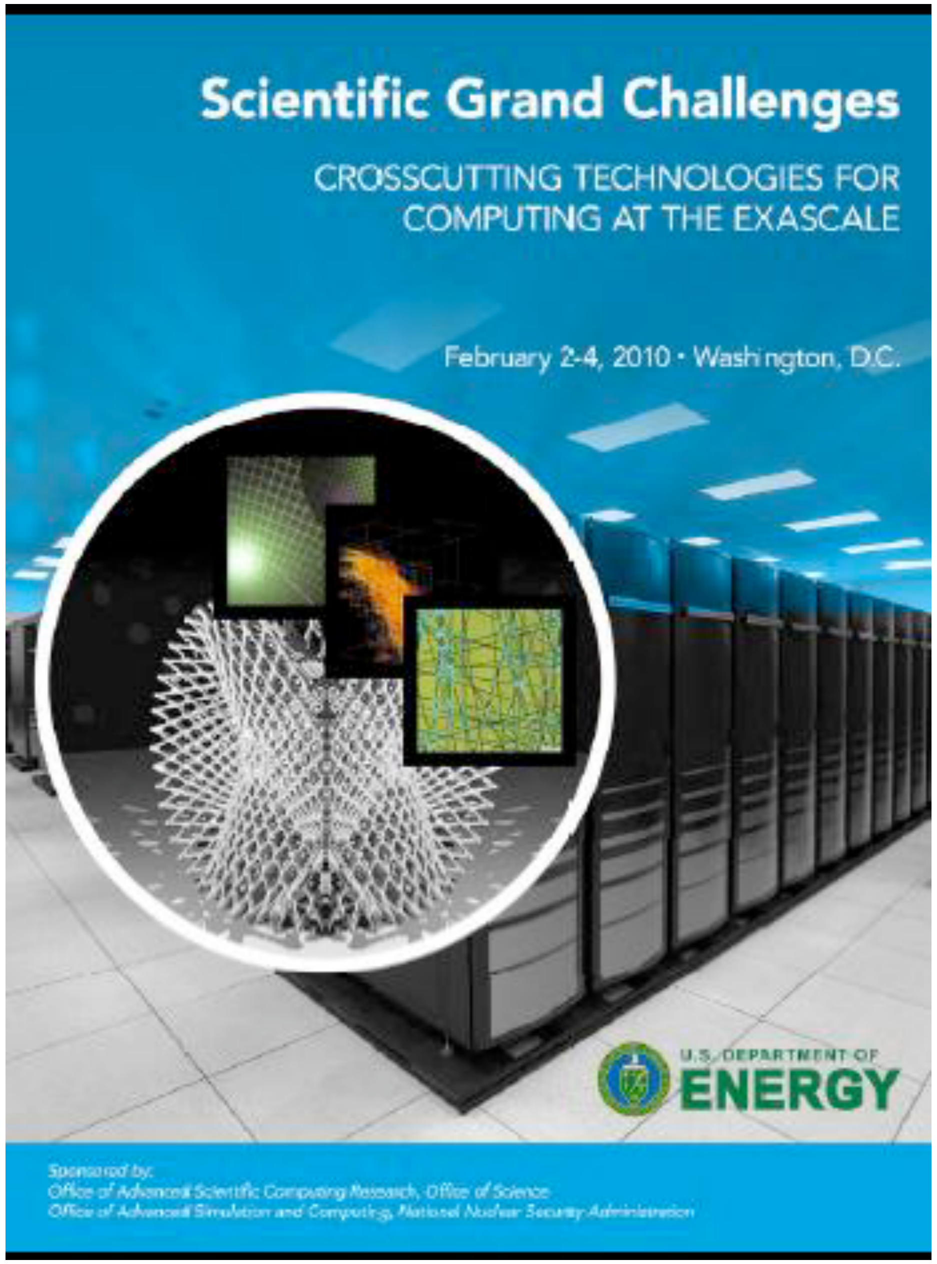


Figure 1. Dependency graph for a portion of the Hartree-Fock procedure.



- Utilization is higher,
- cycle times for an iteration are shorter
- work on the next cycle can begin well before the current cycle completes

one-sided active messages with sub-10-microsecond latency. The send overhead is just 1.165 microseconds



Algorithms:

Recast of applied math algorithms; data analysis; mini-apps; simulations of emerging architectures; etc.

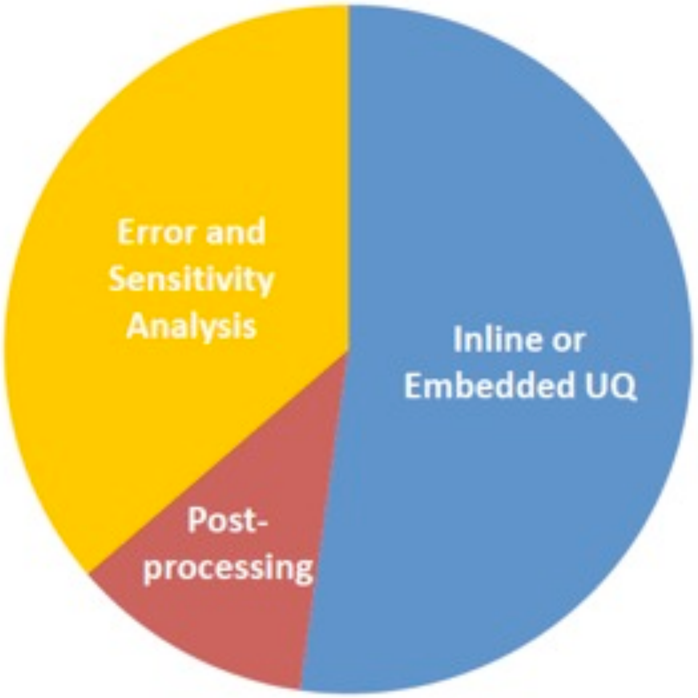
Programming Models

MPI+X; APIs for dynamic resource & power management; scalable I/O; PM support for memory mgt, latency, fault tolerance & resilience; etc.

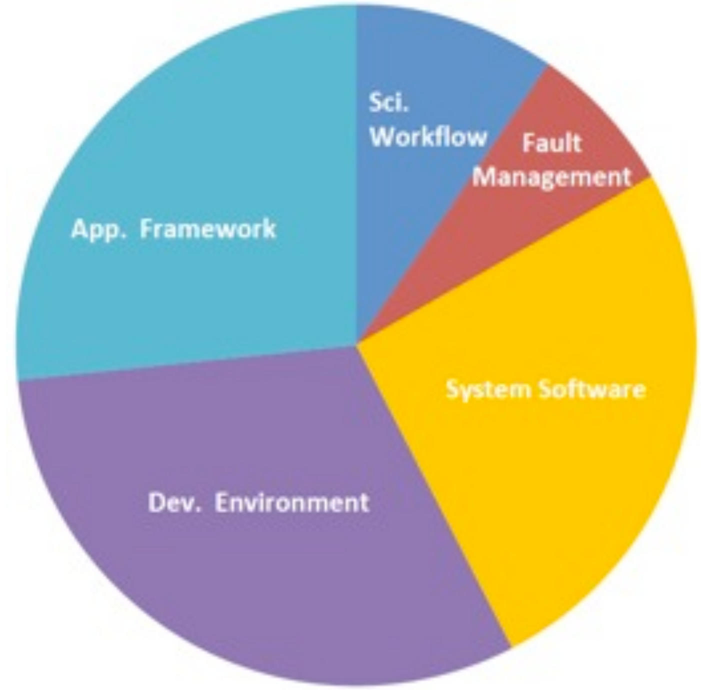
System Software

Node-level parallelism, dynamic resource allocation, memory access, perf measurement & analysis tools, fault management, exascale I/O, etc

ASCR Exascale Funding Trends

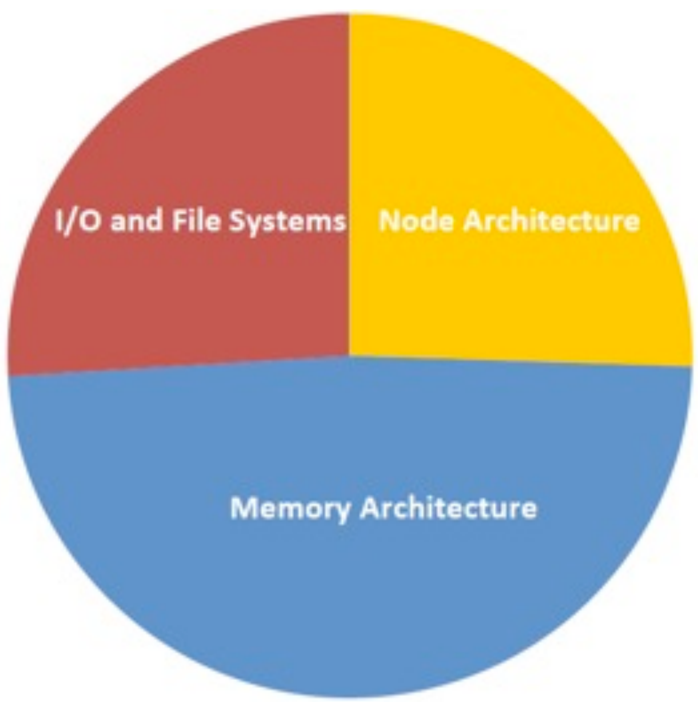


Uncertainty Quantification
6 funded at $3M/yr

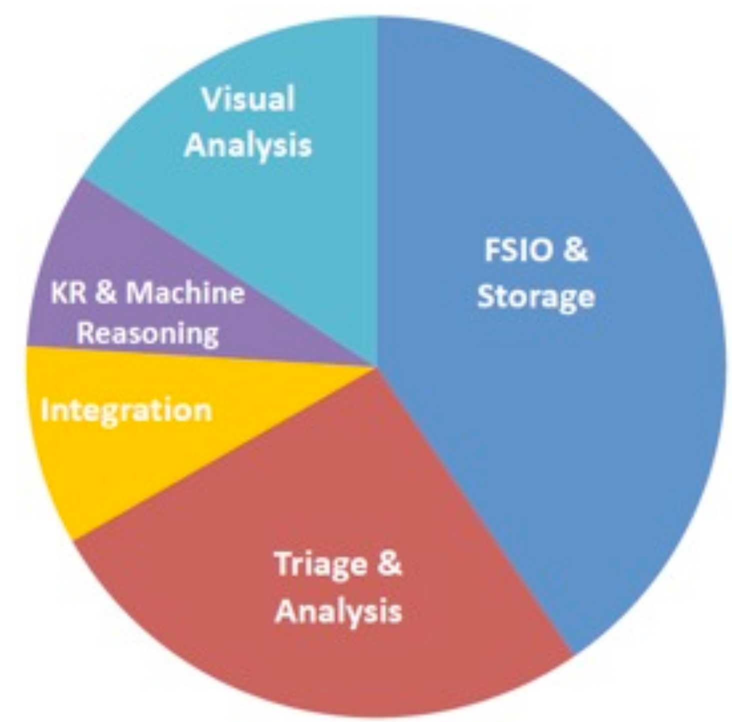


X-Stack
11 funded at $8.5M/yr

Advanced Architectures
6 funded at $5M/yr



Scientific Data Management and Analysis at Extreme Scale
10 (11?) projects funded at $5M/yr



ASCR Exascale Funding Trends: *Co-Design Applications

Center for Exascale Simulation of Advanced Reactors (CESAR)
Rosner (ANL)

FLASH High Energy Density Physics Exascale Codesign Center
Lamb (ANL)

The CERF Center: Co-design for Exascale Research in Fusion
Koniges (LBNL)

Exascale Co-Design Center for Materials in Extreme Environments: Engineering-Scale Predictions
Germann (LANL)

Chemistry Exascale Co-Design Center
Harrison (ORNL)

Combustion Exascale Co-Design Center
Chen (SNL)

Exascale Performance Research for Earth System Simulation (EXPRESS)
Jones (LANL)

* as of March

ASC Working Groups (+ ASCR Participation)

- Applications
- Visualization and Data Analysis
- Solvers, Algorithms, and Libraries
- Programming Models
- System Software
- Tools
- I/O, Networking, and Storage
- Hardware Architecture

From Thuc Hoang for Application Scientists

•What are your computational needs for the next decade?

•Are you currently adapting your codes to changes in architectures?

•Are you exploring alternate programming models to MPI-everywhere? If so, do you have any preliminary conclusions?

•Do you have requirements that you would propose for a new programming model?

•Who do you believe is responsible for creating alternative programming models? Do you see this as a CS community activity, an applications activity, or something in between?

•Do you see a direct path forward for your application code on exascalearchitectures? What changes do you believe will be required of your application code and what is your estimate of resources required?

•Is the right talent available and are you able to recruit the right talent to work on your application that will take it to exascale?

•How much flexibility do you have to adapt your methods and algorithms to changes in technology? Will you need to fundamentallyrethink the approach or can you proceed incrementally? If fundamental re-thinking is required, do you have the staff required to do so?

•Do you believe you have the programmatic flexibility to explore co-design space for the future of your application? Why or why not?

•What tools do you need to provide feedback to the architectures community?

•What programmatic workload do you anticipate in the future? Is the ASC balance between capability and capacity platforms still correct? Is defining computing in this manner still useful?

Performance is Limited by ...

1) **System power** -primary constraint
2) **Memory** bandwidth and capacity are not keeping pace
3) **Concurrency** 1000X increase in-node
4) **Processor** open question
5) **Programming model** compilers will not hide this
6) **Algorithms** need to minimize data movement, not flops
7) **I/O bandwidth** unlikely to keep pace with machine speed
8) **Reliability and resiliency** will be critical at this scale
9) **Bisection bandwidth** limited by cost and energy

Bottom Line Challenges of Exascale Computing

Power efficiency,
Reliability,
Programmability