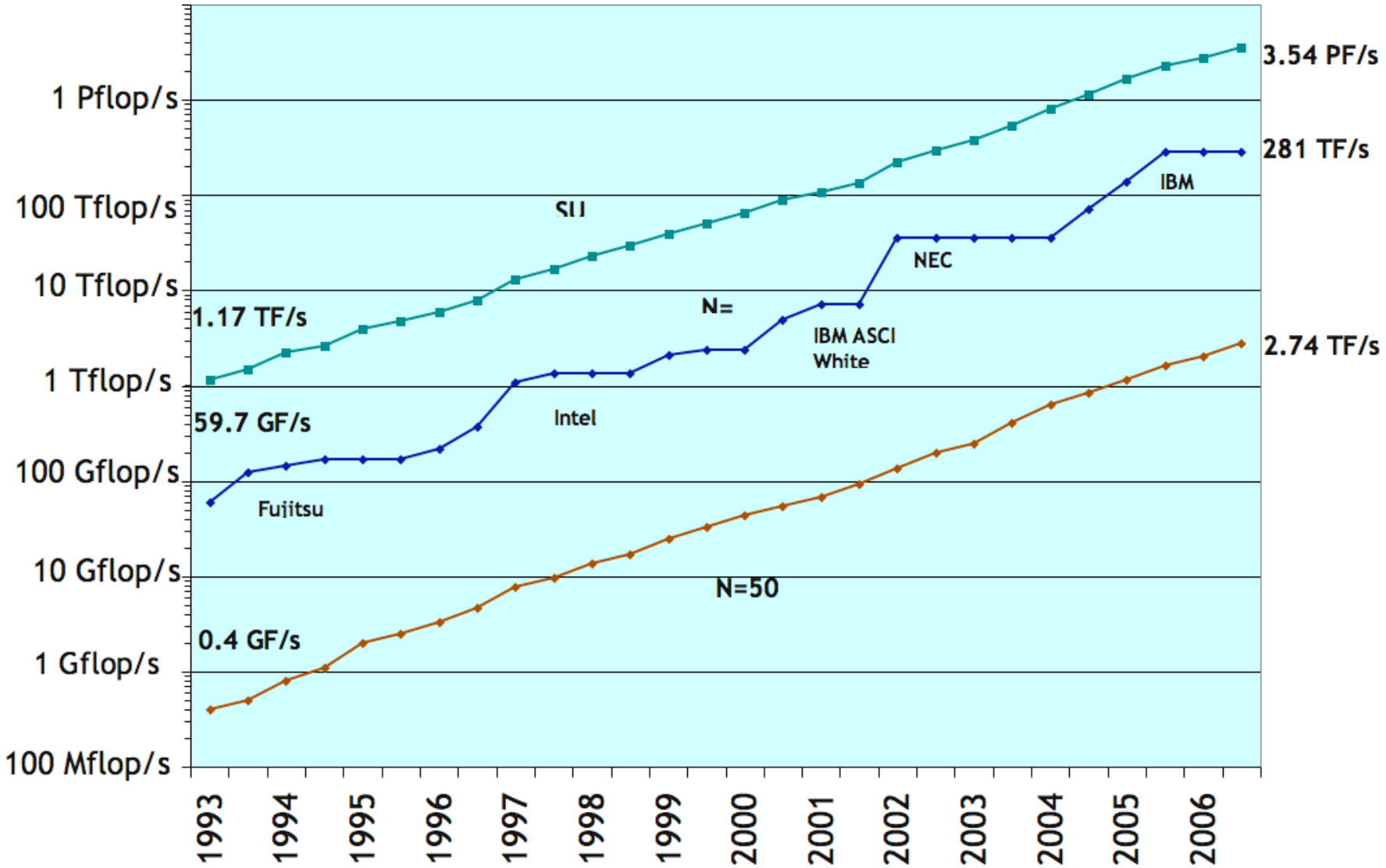
The Path from Petascale to Exascale Hardware and Applications Issues

Rick Stevens
Argonne National Laboratory
University of Chicago

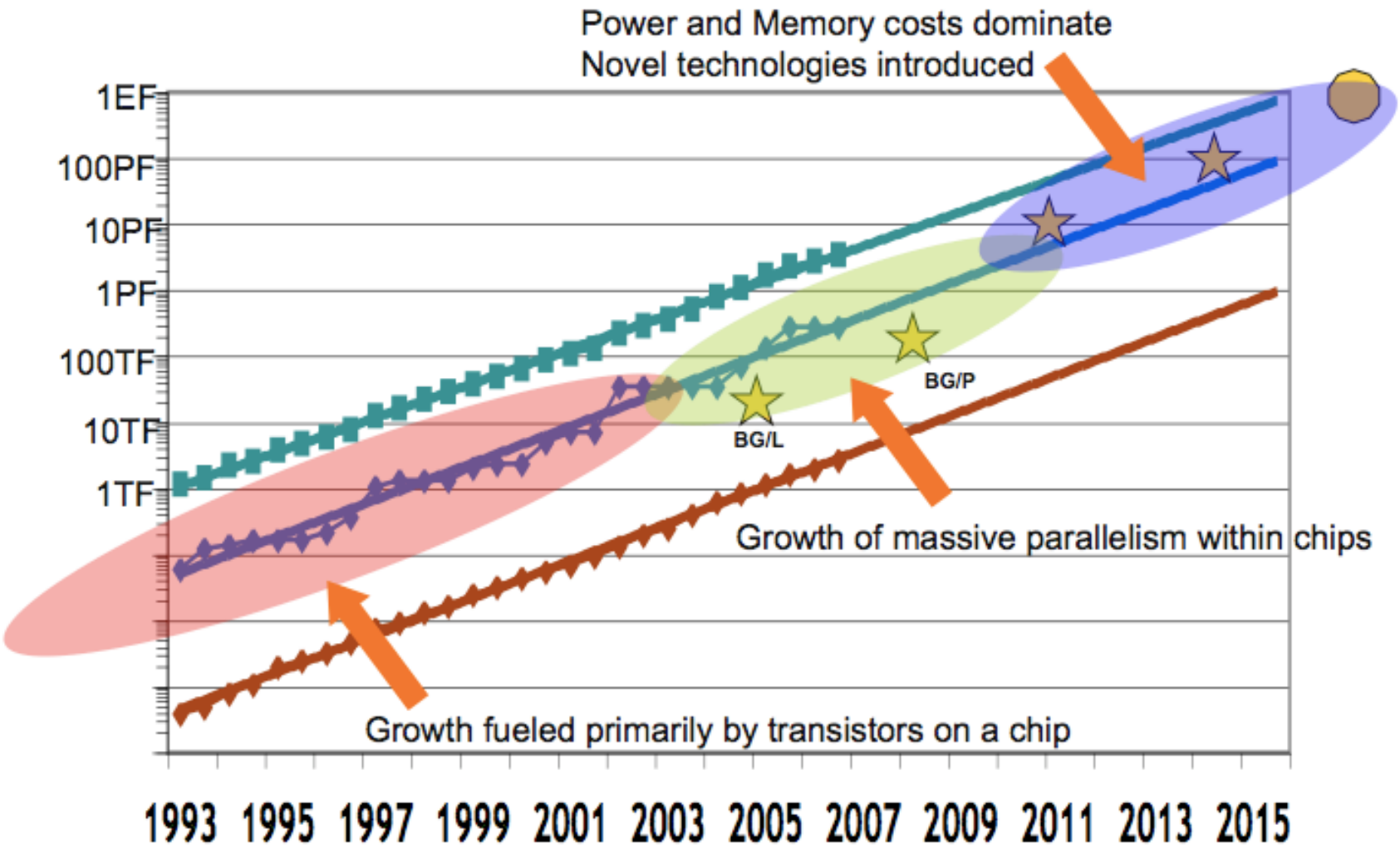
Supercomputing & Cloud Computing

- Two dominant macro architectures dominate large-scale (intentional) computing infrastructures (vs embedded & ad hoc)
- Supercomputing type Structures
  - Large-scale integrated coherent systems
  - Managed for high utilization and efficiency
- Emerging cloud type Structures
  - Large-scale loosely coupled, lightly integrated
  - Managed for availability, throughput, reliability

Top 500 Trends



Looking to Exascale



A Three Step Path to Exascale

Begin Full System Delivery (Yr)	2004	2007	2012	2015	2019
Design Parameters	**BG/L**	**BG/P**	**ONE**	**TWO**	**THREE**
Cores / Node	2	4	16	32	96
Clock Speed (GHz)	0.7	0.85	1.6	2.3	2.8
Flops / Clock / Core	4	4	8	16	16
Nodes / Rack	1024	1024	512	1024	1024
Racks / Full System Config	64	72	256	256	256
MB RAM/core	256	512	1024	1024	1024
Total Power	2.5MW	4.8MW	8MW	30MW	40MW
Flops / Node (GF)	5.6	14	205	1178	4301
Flops / Rack (TF)	5.7	14	105	1206	4404
LB Concurrency	5.E+05	1.E+06	2.E+07	1.E+08	4.E+08
Full System					
Total Cores (Millions)	0.13	0.3	2	8	25
Total RAM (TB)	33.6	151	2147	8590	25770
Total Racks	64	72	256	256	256
Peak Flops System (PF)	**0.37**	**1**	**27**	**309**	**1127**

A Three Step Path to Exascale

Begin Full System Delivery (Yr)	2004	2007	2012	2015	2019
Design Parameters	**BG/L**	**BG/P**	**ONE**	**TWO**	**THREE**
Cores / Node	2	4	16	32	96
Clock Speed (GHz)	0.7	0.85	1.6	2.3	2.8
Flops / Clock / Core	4	4	8	16	16
Nodes / Rack	1024	1024	512	1024	1024
Racks / Full System Config	64	72	256	256	256
MB RAM/core	256	512	1024	1024	1024
Total Power	2.5MW	4.8MW	8MW	30MW	40MW
Flops / Node (GF)	5.6	14	205	1178	4301
Flops / Rack (TF)	5.7	14	105	1206	4404
LB Concurrency	5.E+05	1.E+06	2.E+07	1.E+08	4.E+08
Full System					
Total Cores (Millions)	0.13	0.3	2	8	25
Total RAM (TB)	33.6	151	2147	8590	25770
Total Racks	64	72	256	256	256
Peak Flops System (PF)	**0.37**	**1**	**27**	**309**	**1127**

Top Pinch Points

- Power Consumption
  - Proc/mem, I/O, optical, memory, delivery
- Chip-to-Chip Interface Scaling (pin/wire count)
- Package-to-Package Interfaces (optics)
- Fault Tolerance (FIT rates and Fault Management)
  - Reliability of irregular logic, design practice
- Cost Pressure in Optics and Memory

Programming Models: Twenty Years and Counting

- In large-scale scientific computing today essentially all codes are message passing based (CSP and SPMD)
- Multicore is challenging the sequential part of CSP but there has not emerged a dominate model to augment message passing

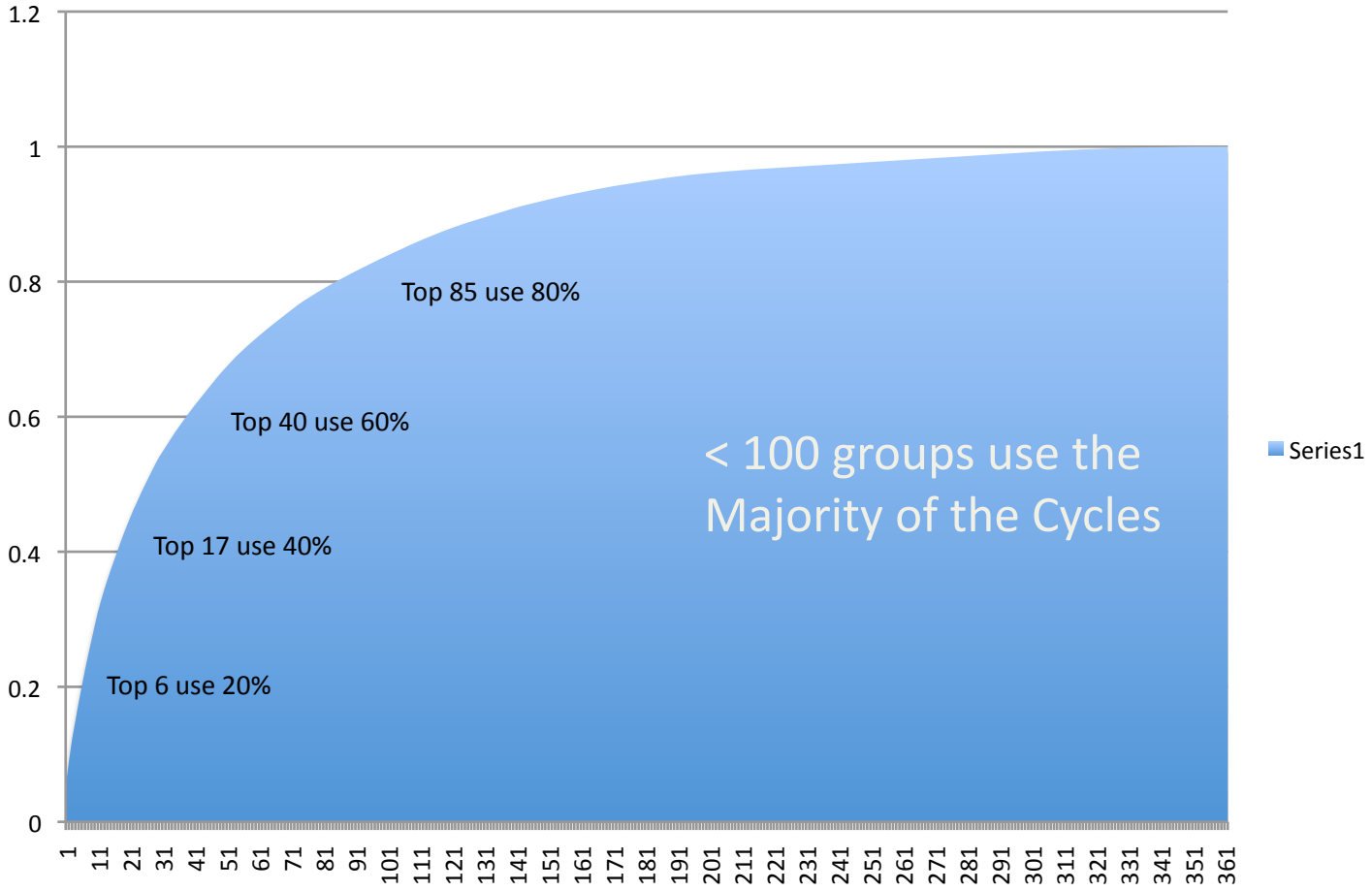
Quasi Mainstream Programming Models

- C, Fortran, C++ and MPI
- OpenMP, pthreads
- CUDA, RapidMind
- Clearspeeds Cn
- PGAS (UPC, CAF, Titanium)
- HPCS Languages (Chapel, Fortress, X10)
- HPC Research Languages and Runtime
- HLL (Parallel Matlab, Grid Mathematica, etc.)

Existing Applications of Interest

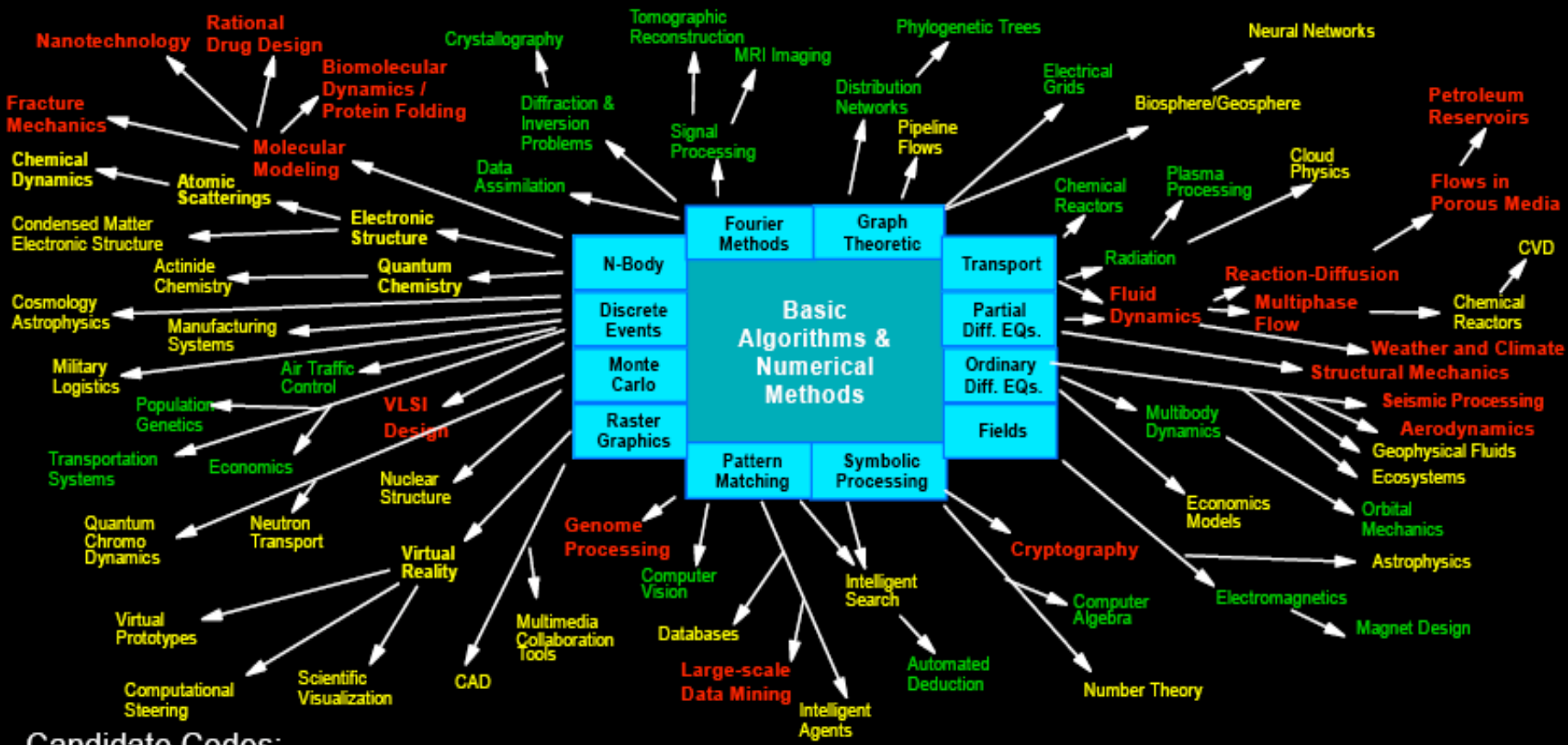
- Climate and Weather (e.g. CCM3, POP, WRF)
- Plasma Physics (e.g. GTC, GYRO, M3D)
- Combustion (e.g. S3D, NCC)
- Multi-physics CFD (e.g. NEK, SHARP)
- Lattice QCD (e.g. MILC, CPS)
- Cosmology and Relativity (e.g. ENZO, Cactus)
- Astrophysics (e.g. FLASH, CHIMERA)
- Molecular Dynamics (e.g. NAMD, AMBER)
- Electronic Structure (e.g. QBOX, LSMS, QMC)
- Evolution (e.g. mrBayes, Clustalw-MPI)

NERSC 2007 Rank Abundance



Good Better Best

Many Classes of Applications are Massively Parallel



- Candidate Codes:
  - Inherently parallel; written using MPI
  - Memory required per MPI task is less than that available on a BG/L node
  - Dominated by collective communication across all nodes
  - Locality of communications within 3D mapping
- Non-Candidate Codes:
  - Large memory footprints required on individual nodes
  - Client/server structures
  - Dominated by disk I/O

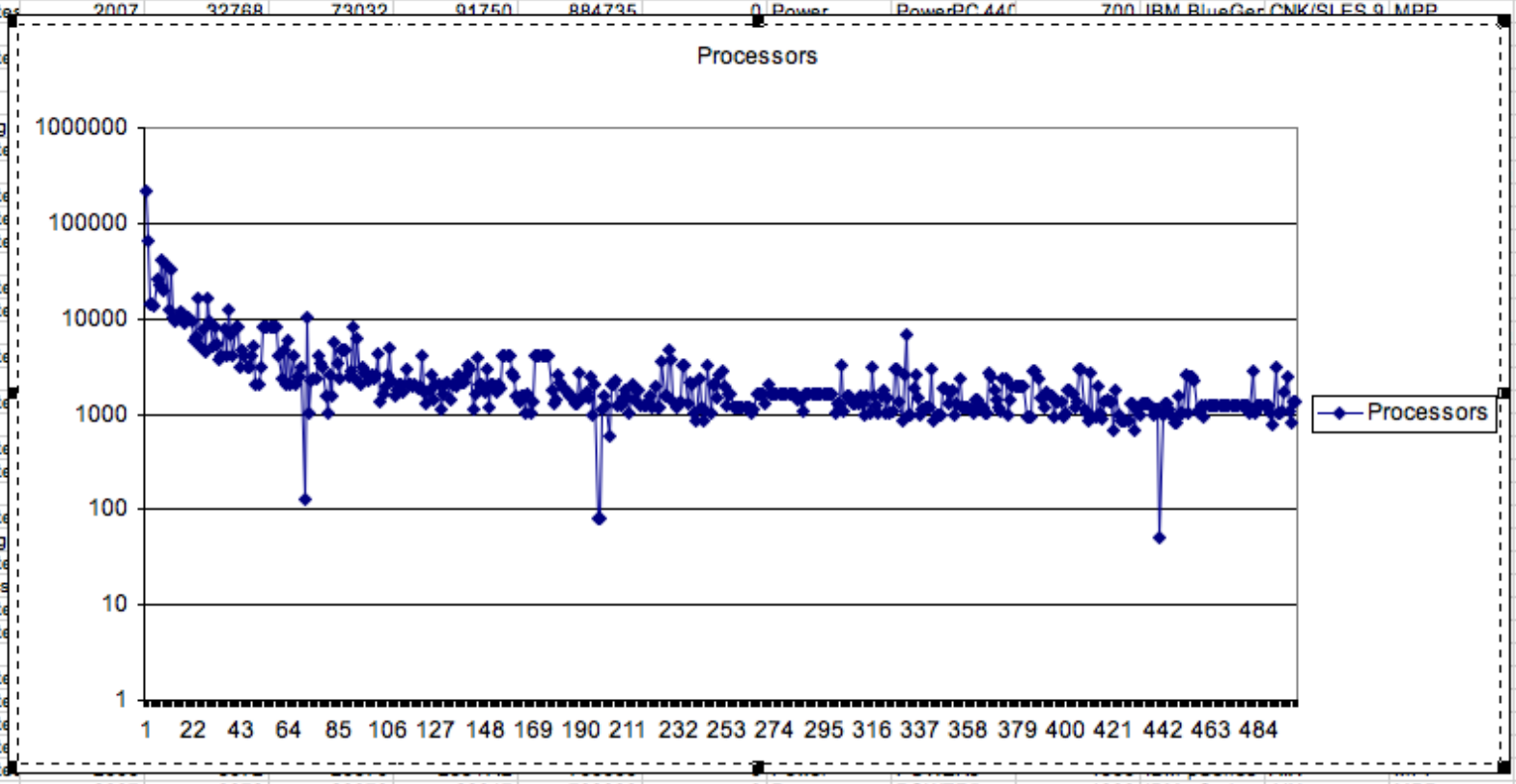
Million Way Concurrency Today

- Little's law driven need for concurrency
  - To cover latency in memory path
  - Function of aggregate memory bandwidth and clock speed
  - Independent of technology and architecture to first order
- Mainstream CPUs (e.g. x86, PPC, SPARC)
  - 8-16 cores, 4-8 hardware threads per core,
  - Total system with 10^3 – 10^5 nodes => 32K – 12M threads
  - BG/P example at 1 PF 72 x 4K = 300,000 (but each thread has to do 4 ops/clock) => 1.2M ops per clock
- GPU based cluster (e.g. 1000 Tesla 1 U nodes)
  - 3 x 128 cores x (32-96) threads per core x 1000 nodes = 12M – 36M threads

Existing Body of Parallel Software

- How many existing HPC science and engineering codes scale beyond 1000 processors?
  - My estimate is that it is less than 1000 world wide
  - Top users at NERSC, OLCF and ALCF < 200 groups
  - It appears likely that the bulk of cycles on Top500 are used in capacity mode with the exception of a sites with policies that enforce capability runs
- How quickly are new codes being generated?
  - Ab initio development
  - Migration and porting from previous generations
- There are different choices faced by large-established projects and personal explorations of new technologies

Number of Processors In the Top500



Speculations on The Shift

- Provisioning by the kilogram ← discrete units
  - I/O surface to volume effects, flexible topologies, the computer is the computer
- Reconfigurable hardware ← porting software
  - Based on programming models that are inherently parallel and scale invariant to shift the problem to emulation not discovery of concurrency
- Internally self powered ← external power sources
  - Metabolic logic? Photodriven? Beta decay? Accoustic?
- Long service lifetime (100yr+, ZeroM) ← few years + maint
  - Massively redundant computing elements embedded in structurally useful materials?
- Adiabatic logic ← dissipatory logic
  - Ambient environment, no infrastructure