T.J. Watson Research Center

# Using The Active Storage Fabrics Model To Address Petascale Storage Challenges

Blake G. Fitch
Aleksandr Rayshubskiy
T.J. Chris Ward
Michael C. Pitman
Robert S. Germain

4th Petascale Data Storage Workshop
Sunday November 15, 2009
Portland Convention Center

# BG/ASF starts with a standard enterprise data center…

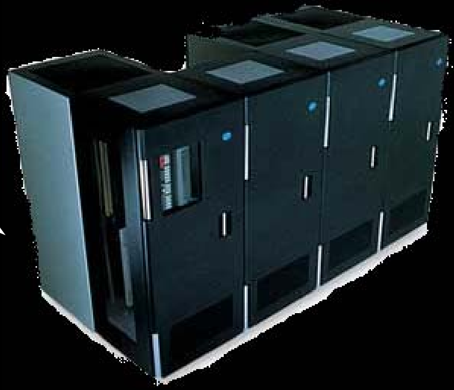
**IBM Enterprise Server**

**Application Solution Driver**

**IBM System Storage DS8000**

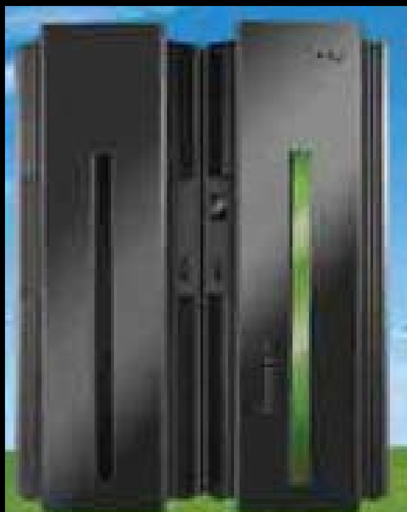
**Persistent Storage**

**IBM 3494 Tape Library**



# …and adds parallel processing in the storage hierarchy.

**IBM Enterprise Server**



Application
Solution
Driver

100s GB/s

**Blue Gene/ASF Option**

Managed as
scalable
Solid State Storage
with
Embedded Process
Modules

**IBM System Storage DS8000**

Persistent
Storage

**IBM 3494 Tape Library**

# Blue Gene Family History (BG/L & BG/P)

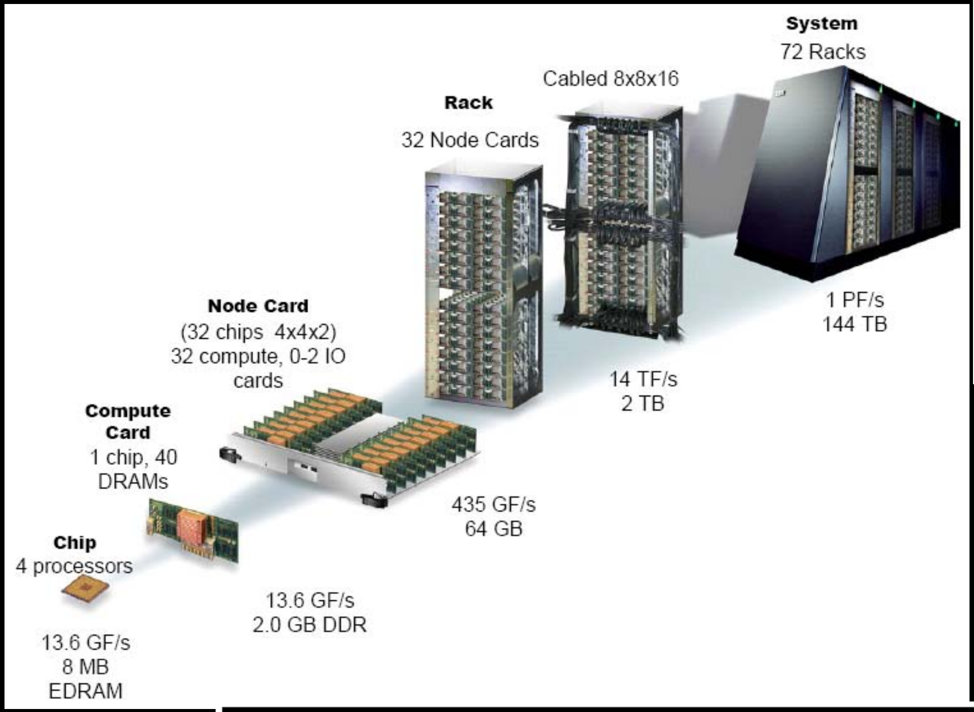
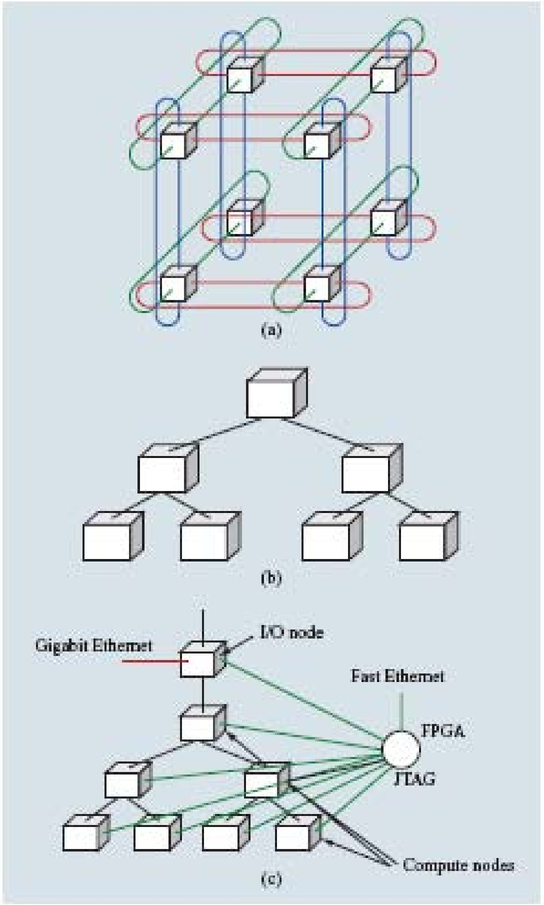



*Table 1-1 Comparison of Blue Gene/L and Blue Gene/P packaging*

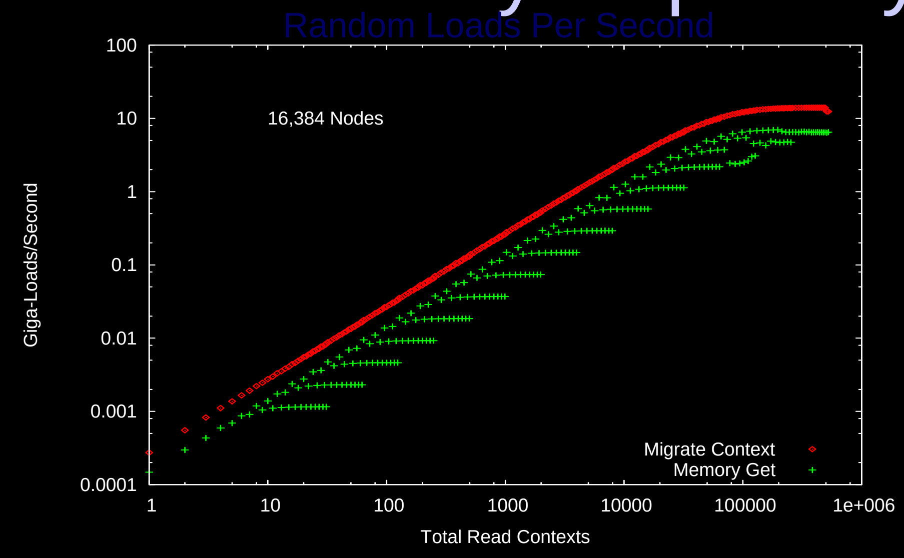
| | Blue Gene/L | | Blue Gene/P | |
|---|---|---|---|---|
| | Quantity per component | To obtain processing speed[a] | Quantity per component | To obtain processing speed[b] |
| Chip | 2 processors | 2.8 GF/s<br>5.6 GF/s | 4 processors | 13.6 GF/s |
| Compute card | 2 chips | 5.6 GF/s<br>11.2 GF/s | 1 chip | 13.6 GF/s |
| Node card | 32 chips; 16 per midplane | 90 GF/s<br>180 GF/s | 32 chips; 16 per midplane | 435 GF/s |
| Rack | 32 node cards | 2.8 TF/s<br>5.6 TF/s | 32 node cards | 14 TF/s |
| System | 64 racks | 180 TF/s<br>360 TF/s | 72 racks | 1 PF/s |

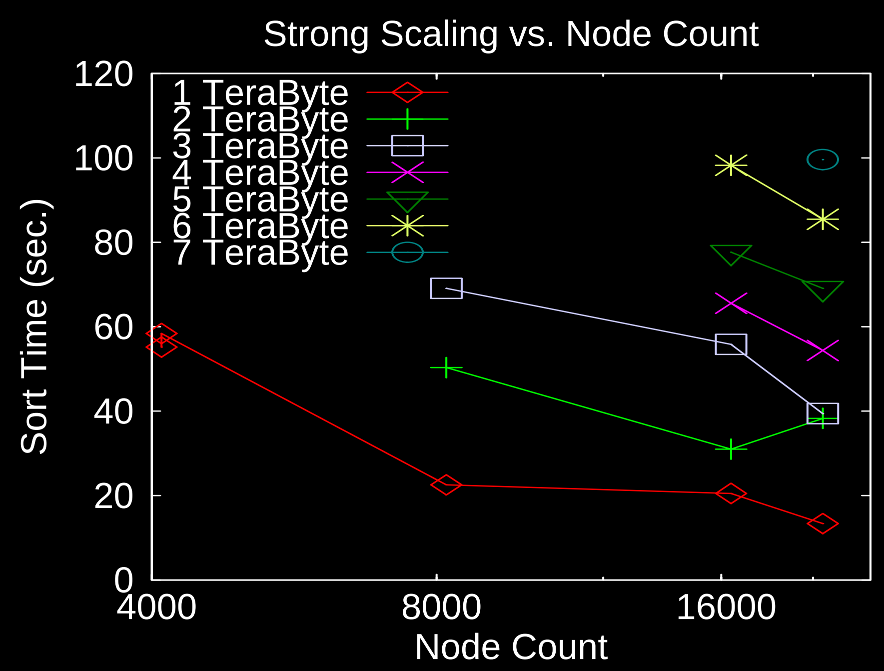
| Feature | Blue Gene/L | Blue Gene/P |
|---|---|---|
| **Network topologies** | | |
| **Torus network** | | |
| Bandwidth | 2.1 GB/s | 5.1 GB/s |
| Hardware latency (nearest neighbor) | 200 ns (32B packet) and 1.6 µs (256B packet) | 100 ns (32B packet) and 800 ns (256B packet) |
| **Global collective network** | | |
| Bandwidth | 700 MB/s | 1.7 GB/s |
| Hardware Latency (round trip worst case) | 5.0 µs | 3.0 µs |
| **Full system (72 rack comparison)** | | |
| Peak performance | 410 TFlop/s | ~1 PFlop/s |
| Power | 1.7 MW | ~2.3 MW |



# Blue Gene Hardware Processor-In-Memory Capability

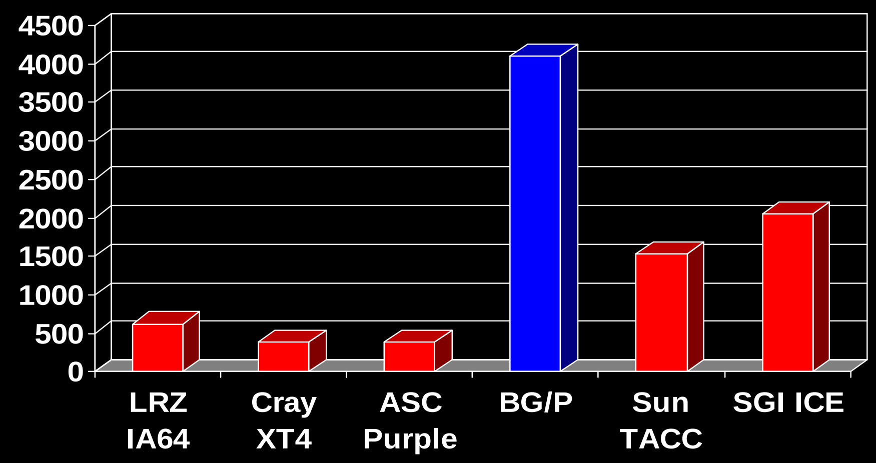
- Blue Gene/L micro benchmarks show order of magnitude improvement over largest SMP and clusters
- Scalable random pointer chasing **through terabytes of memory**
  - BG/L 1.0 GigaLoads / Rack
    - 14 GigaLoads/s for 16 racks
  - P690 0.3 Gigaloads
- Scalable "Indy" sort of terabytes in memory
  - BG/L 1.0 TB sort in 10 secs
  - BG/L 7.5 TB sort in 100 secs
  - Linux cluster, 80 nodes, 2530 disks 435 seconds (disk-to-disk)
- Streams benchmark
  - BG/L 2.5TB / Sec / Rack
  - P5 595 0.2TB / Sec / Rack



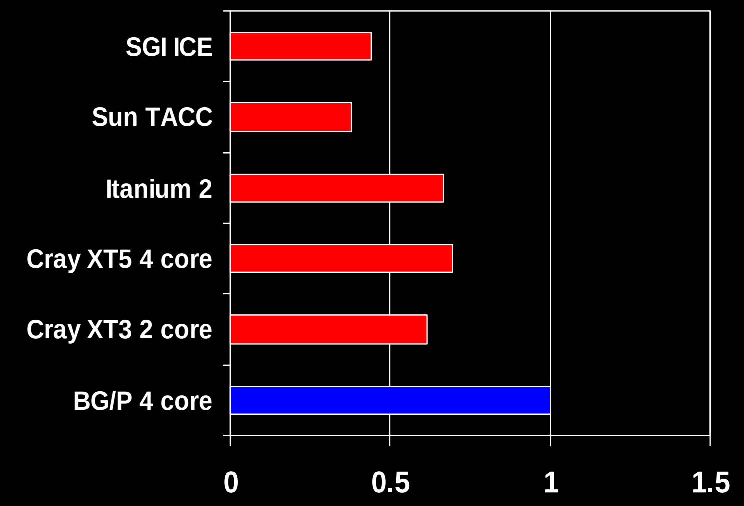



# Blue Gene System-On-A-Chip Advantage

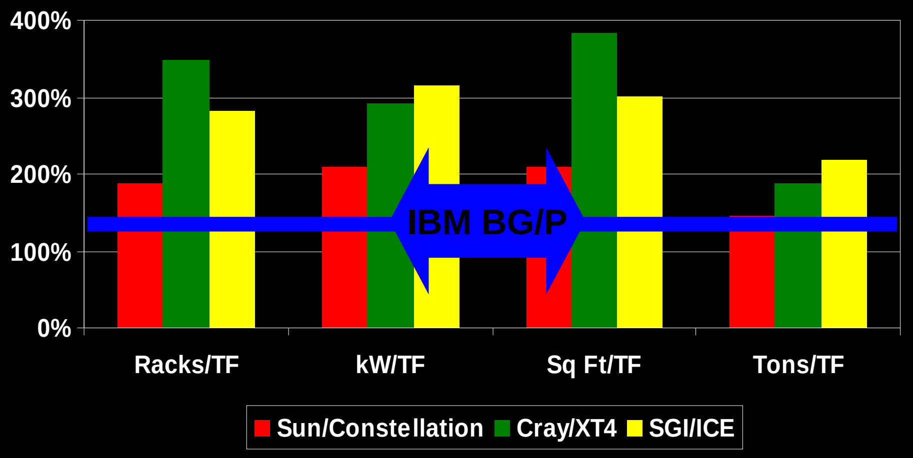
## Main Memory Capacity per Rack



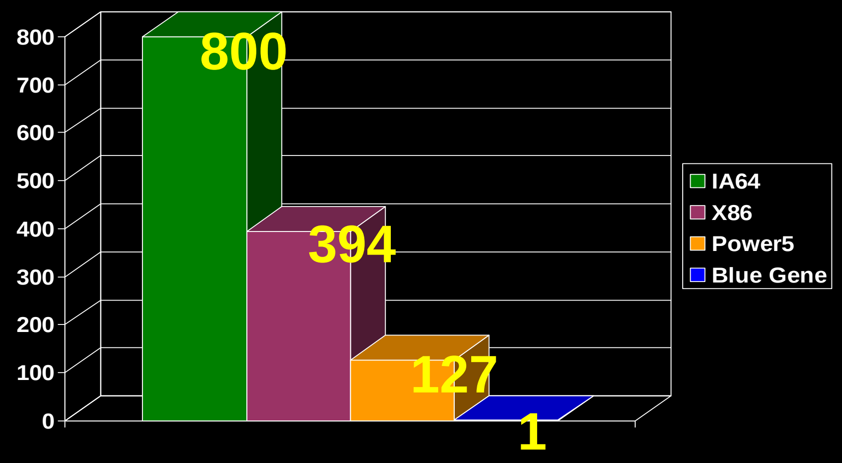

## Peak Memory Bandwidth per node (byte/flop)



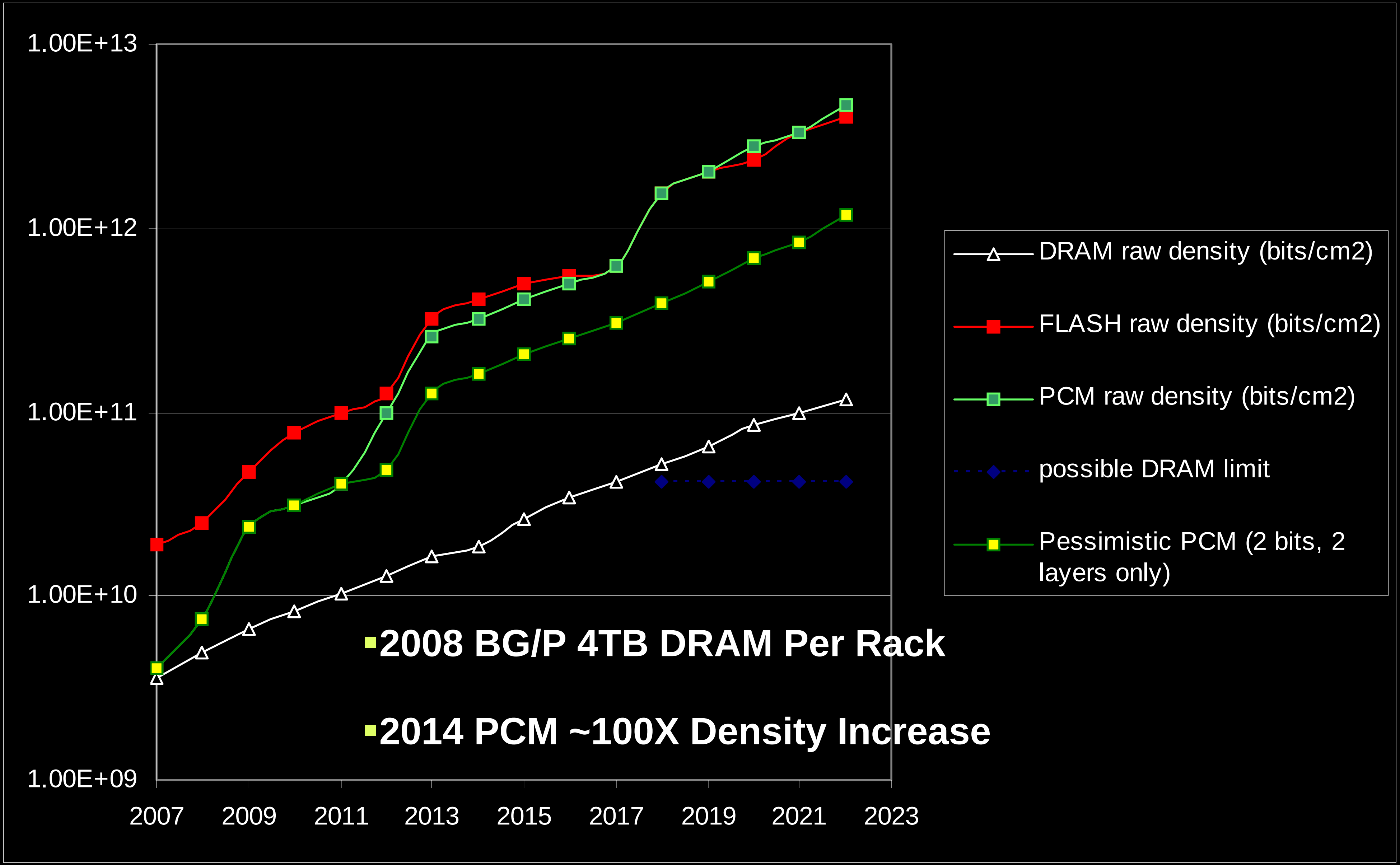

## Relative power, space and cooling efficiencies




## Failures per Month per @ 100 TF



# Phase Change Memory (PCM) Density Timeline




*ITRS 2008 data*

# Scalable, System-on-a-Chip, Storage Class Memory Platform

- **Hypothetical SCM BG/ASF system:**
  - 1024 BG nodes, each with 1TB of SCM → 1 Petabyte Capacity
  - SCM bandwidth to be balanced with network all-to-all capability
    - bounded by bisectional bandwidth
  - Forms a persistent active storage fabric

- **BG/ASF allows hosts to transparently exploit large storage class memories**
  - Offload of bottleneck operations avoids pulling data through the main processor complex
  - resiliency model reduces need to transfer data outside of fabric

- **Current Blue Gene systems already make available sufficient memory for SCM systems research**

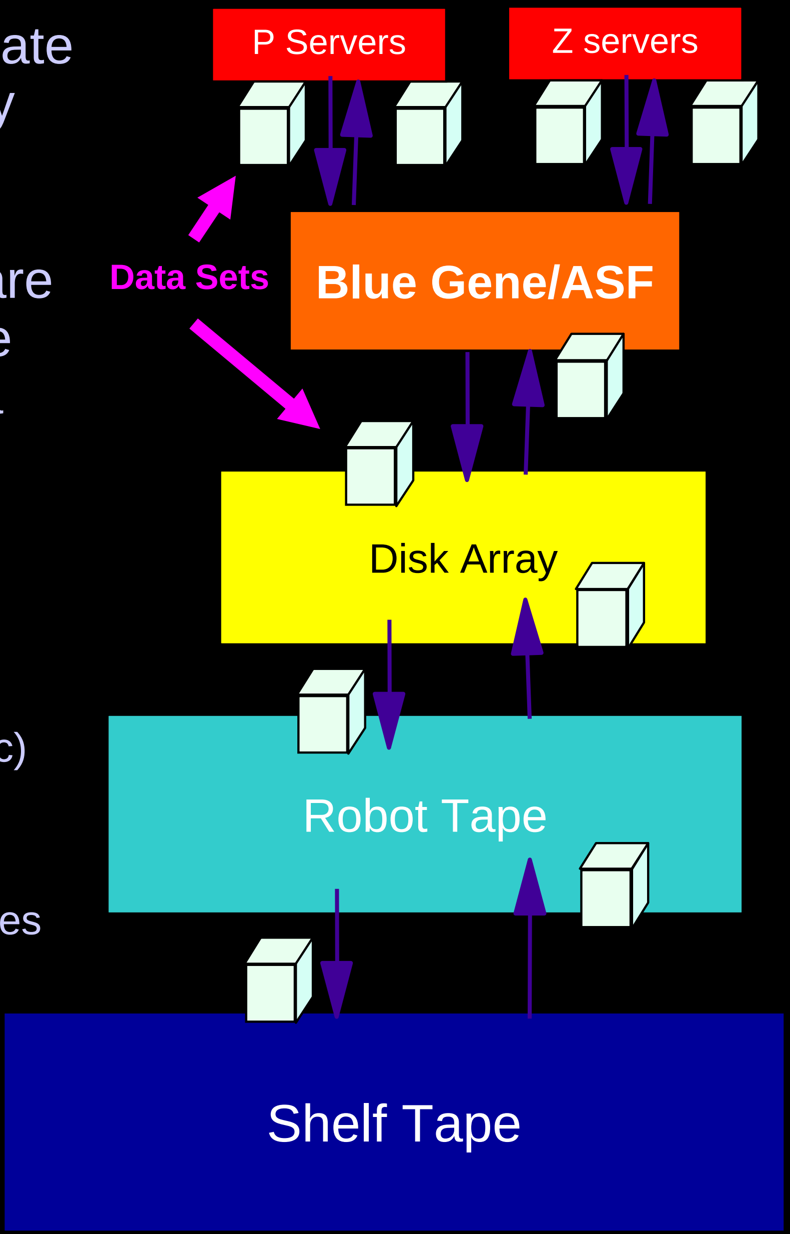
# Blue Gene / Active Storage Fabrics (BG/ASF) Concept

- Manage Blue Gene hardware as a scalable, solid state storage device with embedded processing capability

- Integrate this "Active Storage Fabric" with middleware such as DB2, MySql, GPFS, etc, at the data/storage interface using a parallel, key-value in-memory data store

- Transparently accelerate server Jobs with ASF:
  - Create libraries of reusable, low overhead **E**mbedded **P**arallel **M**odules (**EPM**s) (scan, sort, join, sum, etc)
  - EPMs directly access middleware data (tables, files, etc) in key/value DB based on middleware data model
  - Middleware makes ASF datasets available via legacy interfaces allowing incremental parallelization and interoperability with legacy middleware function and host programs

- Integrate BG/ASF with information life-cycle and other standard IT data management products



# BG/ASF Acceleration Opportunities

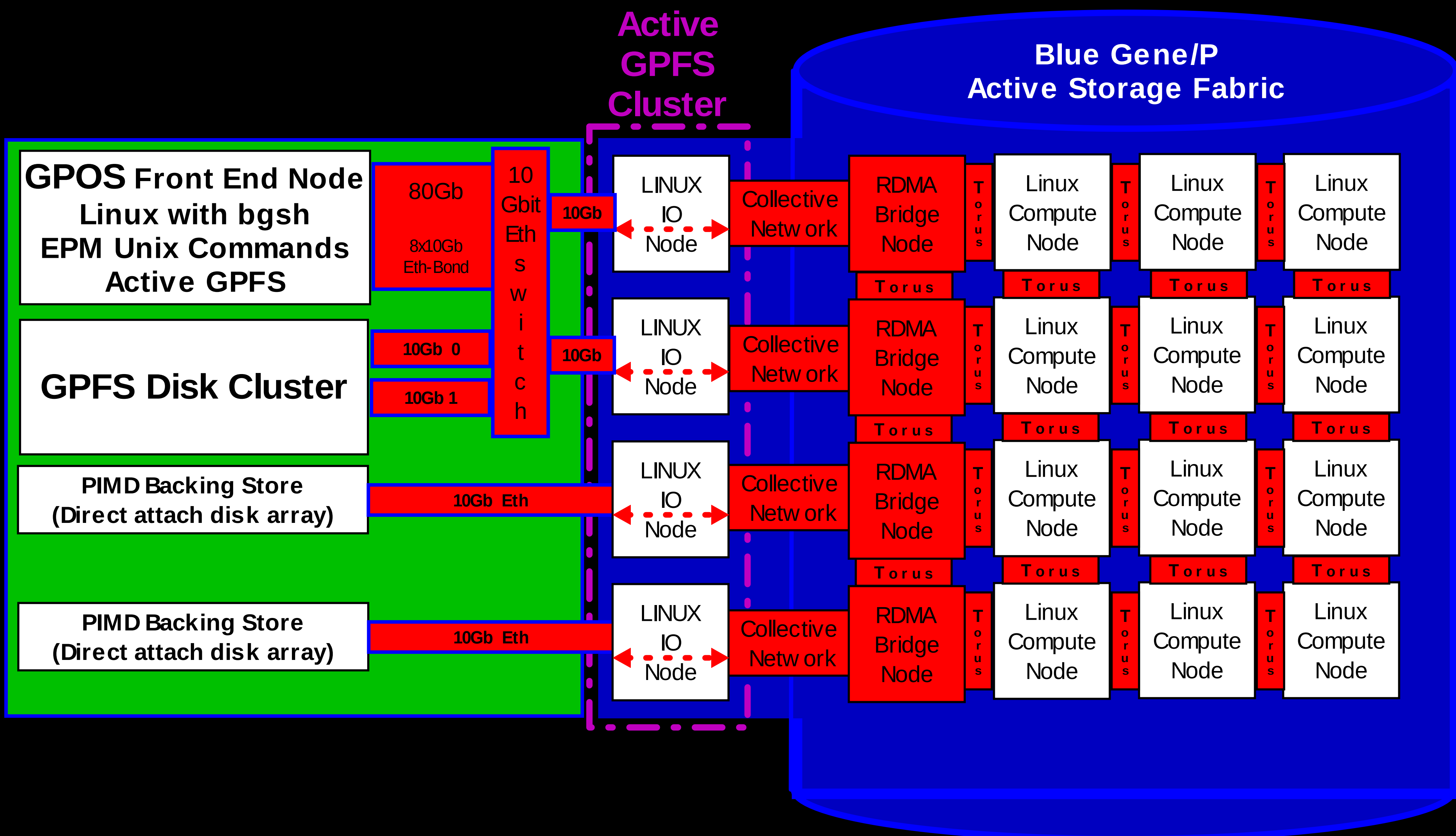
## Embarrassingly Parallel

- **Table scan**
- **Batch serial**
  - Floating point intensive
  - Job farm/cloud
- **Other applications without internal global data dependencies**

## Network Dependent

- **Join**
- **Fast sort**
  - "order by", "group by" queries
- **Map-Reduce**
- **Aggregation operations**
  - count(), sum(), min(), max(), avg(), …
- **Data analysis/OLAP**
  - Aggregation with "group by"…
  - Real-time analytics
- **HPC applications**

# BG/ASF : Accelerating Unix Utilities in Active GPFS



# Blue Gene/L ASF: Early Performance Comparison with Host Commands

**Benchmark process*:**

- **Generate a 22 GB synthetic “Indy Benchmark” data file**
- **grep for records with “AAAAAA” (outputs 1/3’d of the data which is used as input to sort)**
- **sort the output of grep**

* NOTE: All ASF data taken on untuned system, non-optimized and with full debug printfs, all p55 data taken with unix time on command line)

| Function (Measured at Host shell and break out of EPM components) | BG/ASF 512 nodes | pSeries p55 (1 thread) |
|---|---|---|
| **Total host command time: create Indy File:** | **(22GB) 22.26 s** | **686s** |
| **Total host command time: unix grep of Indy File:** | **25.1s** | **3180s** |
| File Blocks to PIMD Records (Embedded grep) : | 7.0 s | |
| Grep core function (Embedded grep) : | 2.5s | |
| PIMD Records to File Blocks (Embedded grep) : | 7.6s | |
| Total time on Blue Gene/ASF fabric (Embedded grep) : | 18.8s | |
| **Total host command time: sort output of grep:** | **60.41s** | **2220s** |
| File Blocks to PIMD Records (Embedded sort) : | 2.9s | |
| Sort core function (Embedded sort) : | 12.2s | |
| PIMD Records to File Blocks (Embedded sort) : | 15.8s | |
| Total time on Blue Gene/ASF fabric (Embedded sort) : | 54.7s | |

# Blue Gene/L ASF: Early Scalability Results

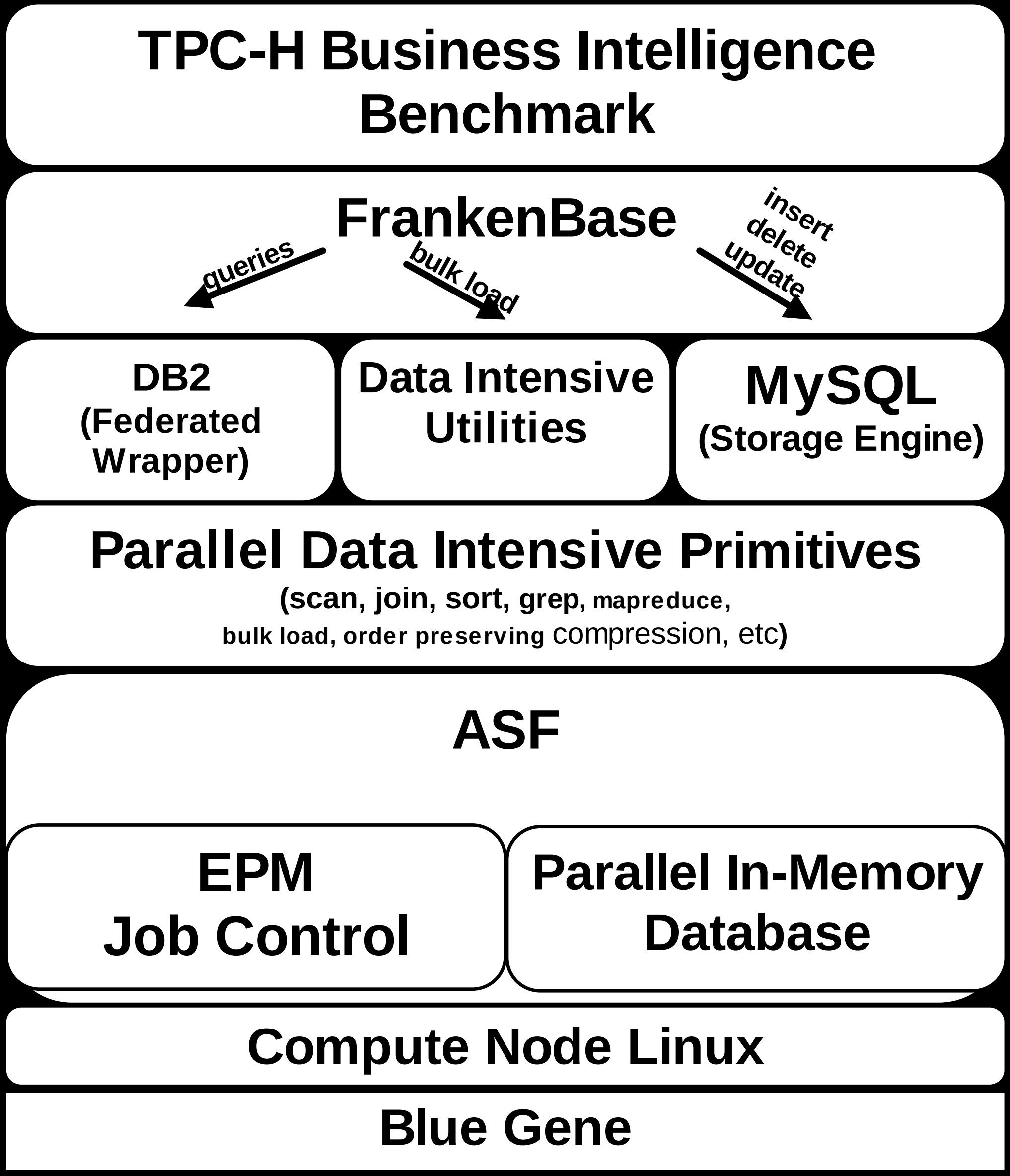
**Benchmark process*:**
- **Generate a 22 GB synthetic "Indy Benchmark" data file**
- **grep for records with "AAAAAA" (outputs 1/3'd of the data which is used as input to sort)**
- **sort the output of grep**

* NOTE: All ASF data taken on untuned system, non-optimized and with full debug printfs, all p55 data taken with unix time on command line)

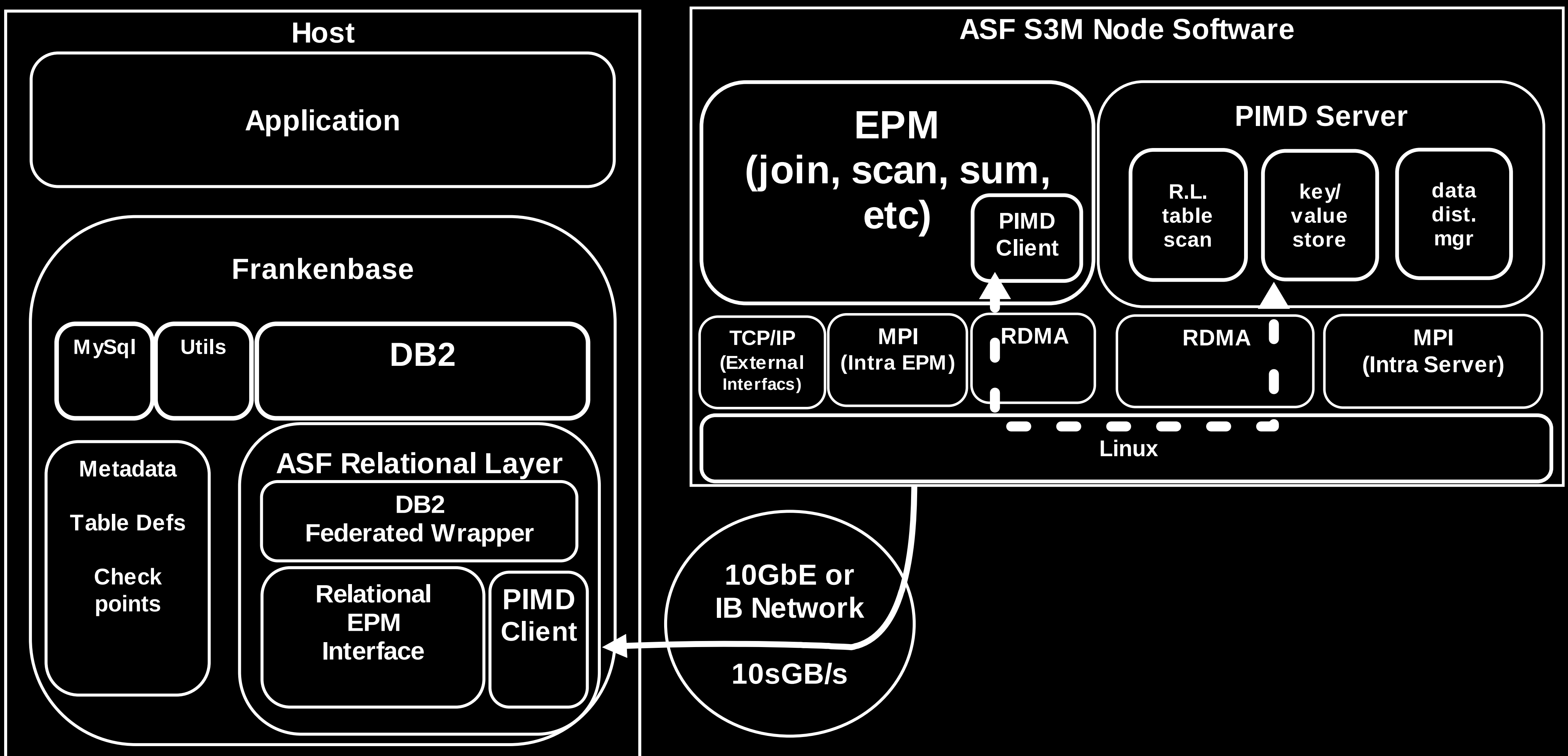
| Function | BG/ASF 512 nodes | BG/ASF 8K nodes |
|---|---|---|
| **Total host command time: create Indy File:** | **(22GB) 22.26 s** | **(1TB) 197s** |
| **Total host command time: unix grep of Indy File:** | **25.1s** | **107s** |
| File Blocks to PIMD Records (Embedded grep) : | 7.0 s | 18.1s |
| Grep core function (Embedded grep) : | 2.5s | 5.1s |
| PIMD Records to File Blocks (Embedded grep) : | 7.6s | 19.9s |
| Total time on Blue Gene/ASF fabric (Embedded grep) : | 18.8s | 50.2s |
| **Total host command time: sort output of grep:** | **60.41s** | **120s** |
| File Blocks to PIMD Records (Embedded sort) : | 2.9s | 6.8s |
| Sort core function (Embedded sort) : | 12.2s | 14.8s |
| PIMD Records to File Blocks (Embedded sort) : | 15.8s | 22.4s |
| Total time on Blue Gene/ASF fabric (Embedded sort) : | 54.7s | 66.55s |

# BG/ASF Relational Database Acceleration

- Explore accelerating RDBMS systems by integrating at the data/storage interface
  - MySQL as an active storage engine
    - Defines tables, does inserts/updates
  - DB2 using Infosphere Federation
    - Offloads queries including scans, joins
    - Some exploitation of DB2 optimizer
- ASF Relational Layer adds
  - External table definition
  - Row scan capability to PIMD using query embedded field information – streaming, parallel, predicate evaluation
  - Embedded parallel join built using MPI and row scan
  - Potential for aggregation

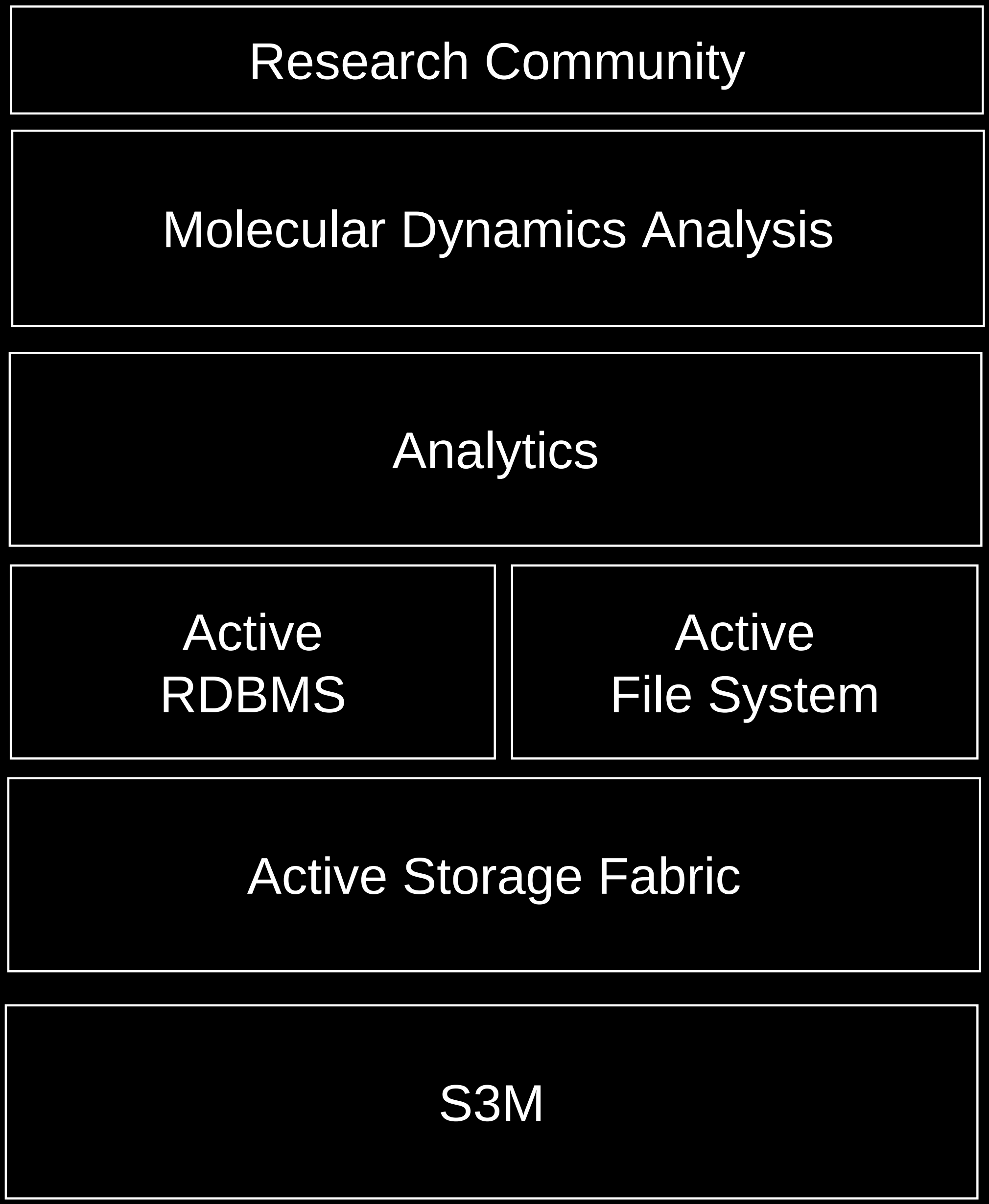


# Frankenbase Component Diagram



# Use case: Shared Molecular Dynamics Scientific Warehouse

- **Very large data volumes which will increase with HPC system capability – expensive to move and store relative to the cost of production**
- **Datasets are often reused for many investigations beyond the initial inquiry – a mix of ad-hoc analytics are employed**
- **Currently, analytic capability and collaborative opportunities can be limited by the ability to access massive molecular dynamics datasets and the resources needed to perform ad-hoc queries**
- **Use of the ASF approach to implement a shared, active, data warehouse is expected to increase the scientific impact of a future super computer facilities**



# Summary

- **Current Status**
  - Demonstrated ASF accelerating unix utilities in Active GPFS
  - Built BG/P ASF node environment (Linux, MPICH, RDMA, PIMD, etc)
  - Completed initial function for ASF based DB2 acceleration
    - ASF Relational Layer
      - Tuple management system for storing table rows in key/value store
      - Query specific compilation for tuple predicates, projections and joins
      - PIMD server based row scan service supporting predicates and projections
      - Parallel join
      - DB2 Federated wrapper – allows offload of DB2 selects (scans, joins)
      - MySQL to parse CREATE TABLE and produce tuple metadata
      - TPC-H table generation
      - PIMD Check point / reload infrastructure

# Backup

# Active Storage Fabrics Key Enablers

- Parallel In-Memory Database (PIMD)
  - The shared data storage model
- Embedded Parallel Modules (EPMs)
  - The storage embedded parallel programming model

# Parallel In Memory Database (PIMD)

- Parallel In Memory Database Overview
  - Key/Value database
  - Client/Server architecture
  - Clients can be parallel programs running on the same nodes as servers
  - … or on external nodes
  - Multiple datasets are cataloged by name (called "Partitioned Data Sets" (PDSs))
  - Multiuser support with UNIX-like PDS access management
  - Clients access records via a target server node determined by a hash function.
  - Server group retains control over actual data locality but optimizes for hash target

- Requirements/Challenges
  - 10k-100k nodes each with 1/P of the data (1GB…16GB per node)
  - Manage distribution including dynamic redistribution of data
  - Minimize overheads – esp. IPC between Client<->Server and Internal (MPI)

- Future research
  - Fault tolerance for platforms beyond Blue Gene/P
  - Lock management and transactions
  - Persistence
  - Security

# Embedded Parallel Module (EPMs)

- Parallel programs that use the Parallel In Memory Database (PIMD)
  - Currently fairly normal MPI programs
  - Use the PIMD Client interface to open PIMD Partitioned Data Sets (PDS).
  - Input and Output operands should be in PIMD PDSs
  - Use locality aware PIMD iterators to work on records that are on or near the node the EPM task is running on.
  - Worst case, an application driven data decomposition is an all-to-all away

- Current Runtime Environment
  - Linux on Blue Gene/P compute nodes – EPMs and the PIMD Server group share the same BG/P nodes.
  - Client <-> Server communications via standard iWARP RDMA (soon, OFED verbs)
  - Client internal communications via MPICH over TCP/IP on Blue Gene/P Networks
  - EPM job Scheduling currently mpirun; expect to move to Load Leveler soon

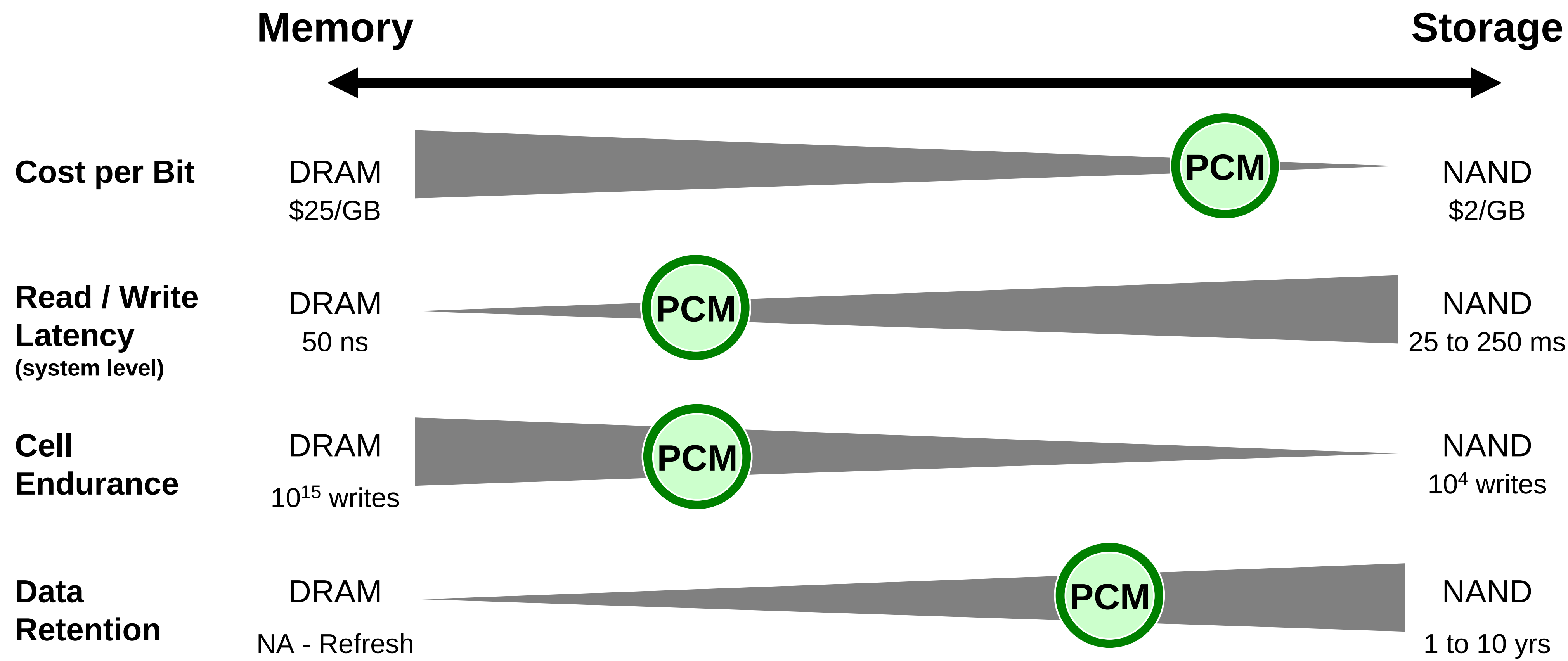
- EPMs encourage parallel solution modularity
  - Exchange operands via PIMD –- an analog to UNIX file system and pipes
  - Initial EPMs will include a subset of UNIX text utilities

- Future research
  - Can be executed in pre-constructed MPI partitions already internally connected and to connected to PIMD Servers (These we call “Virtual Partitions”)

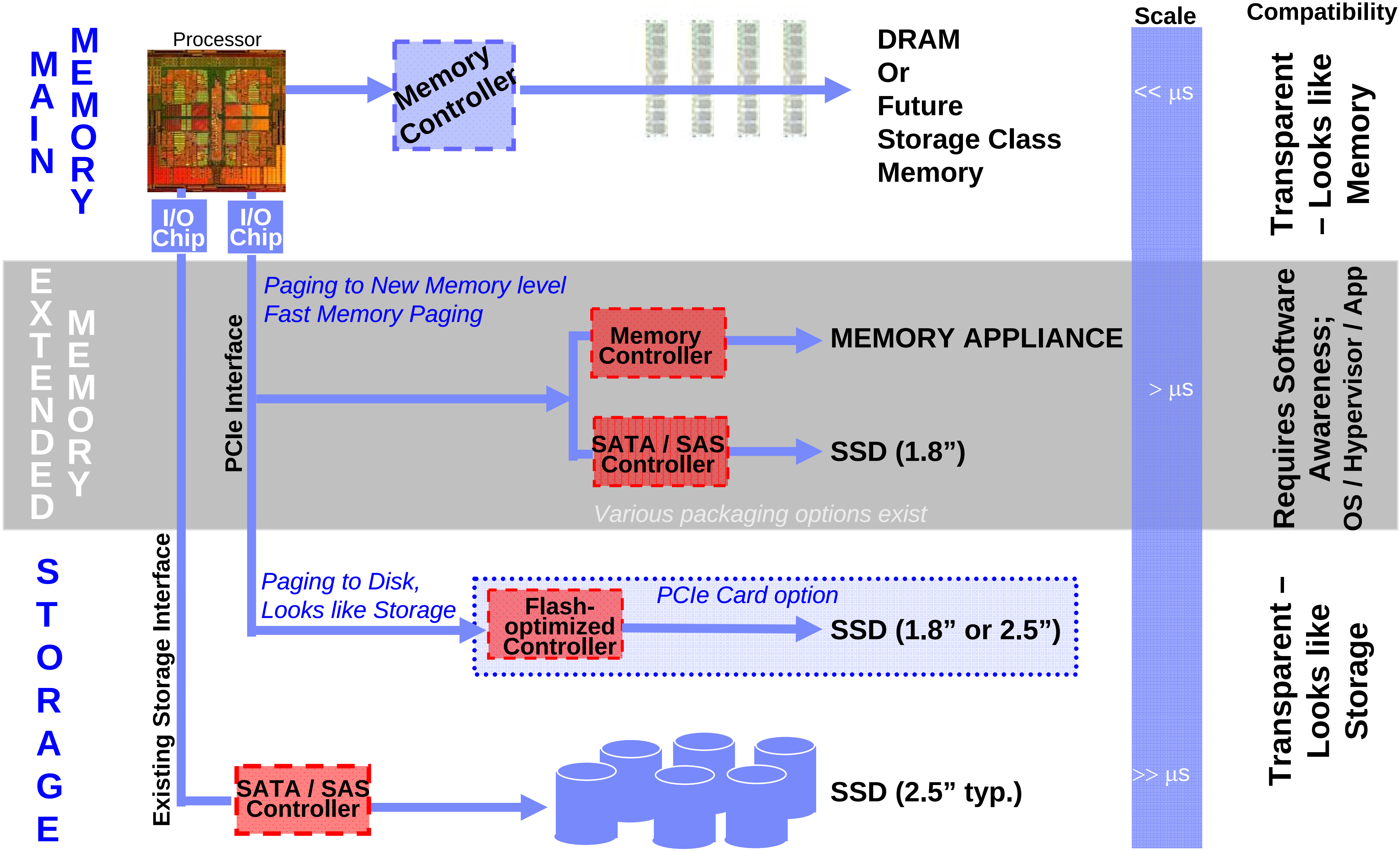
# The Application Landscape

**What can be done with a material with many of the advantages of DRAM but projected cost and retention closer to Flash?**

Note: The numbers shown below are to act as reference points, not absolute targets/requirements.

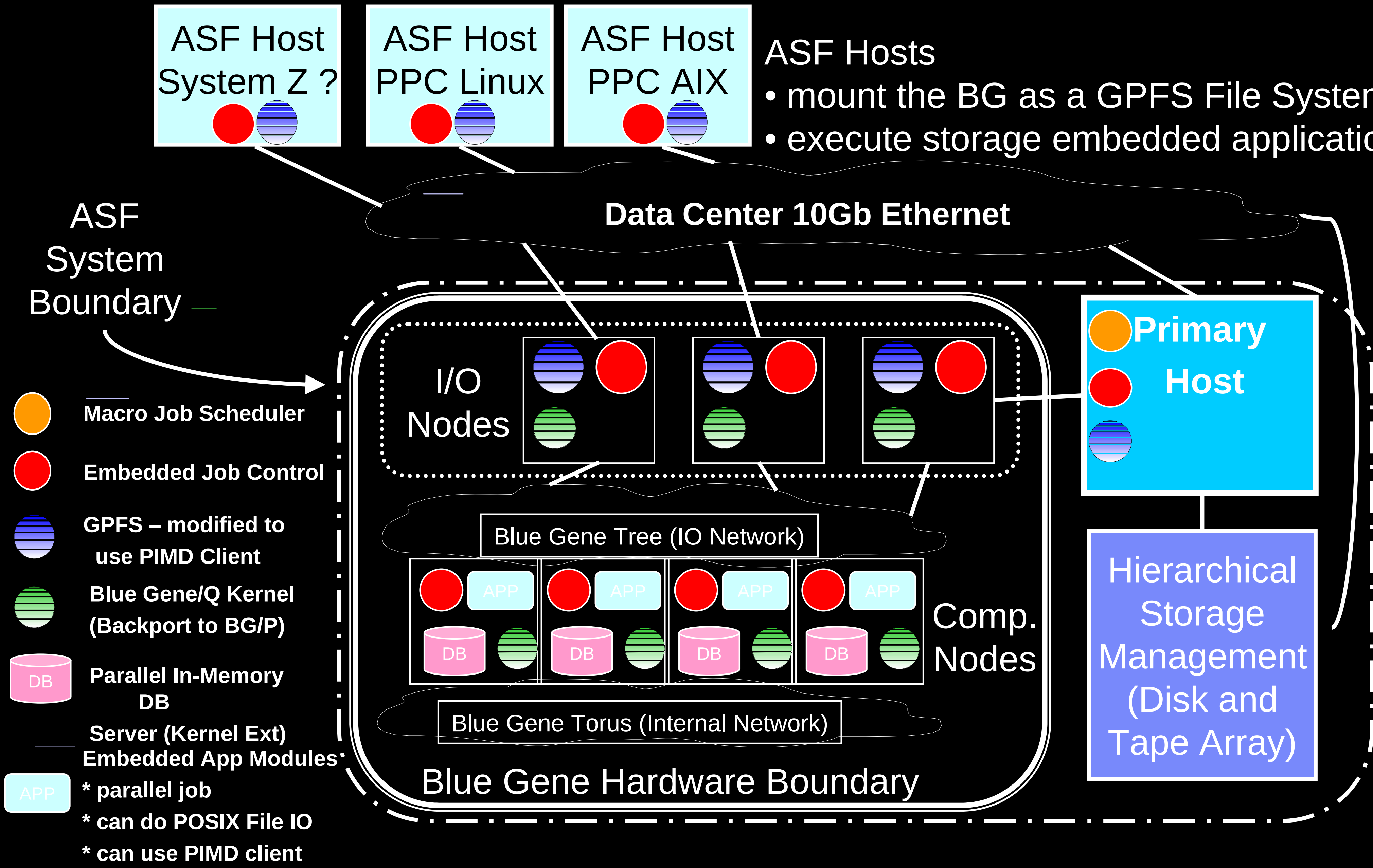


# The Design Space:

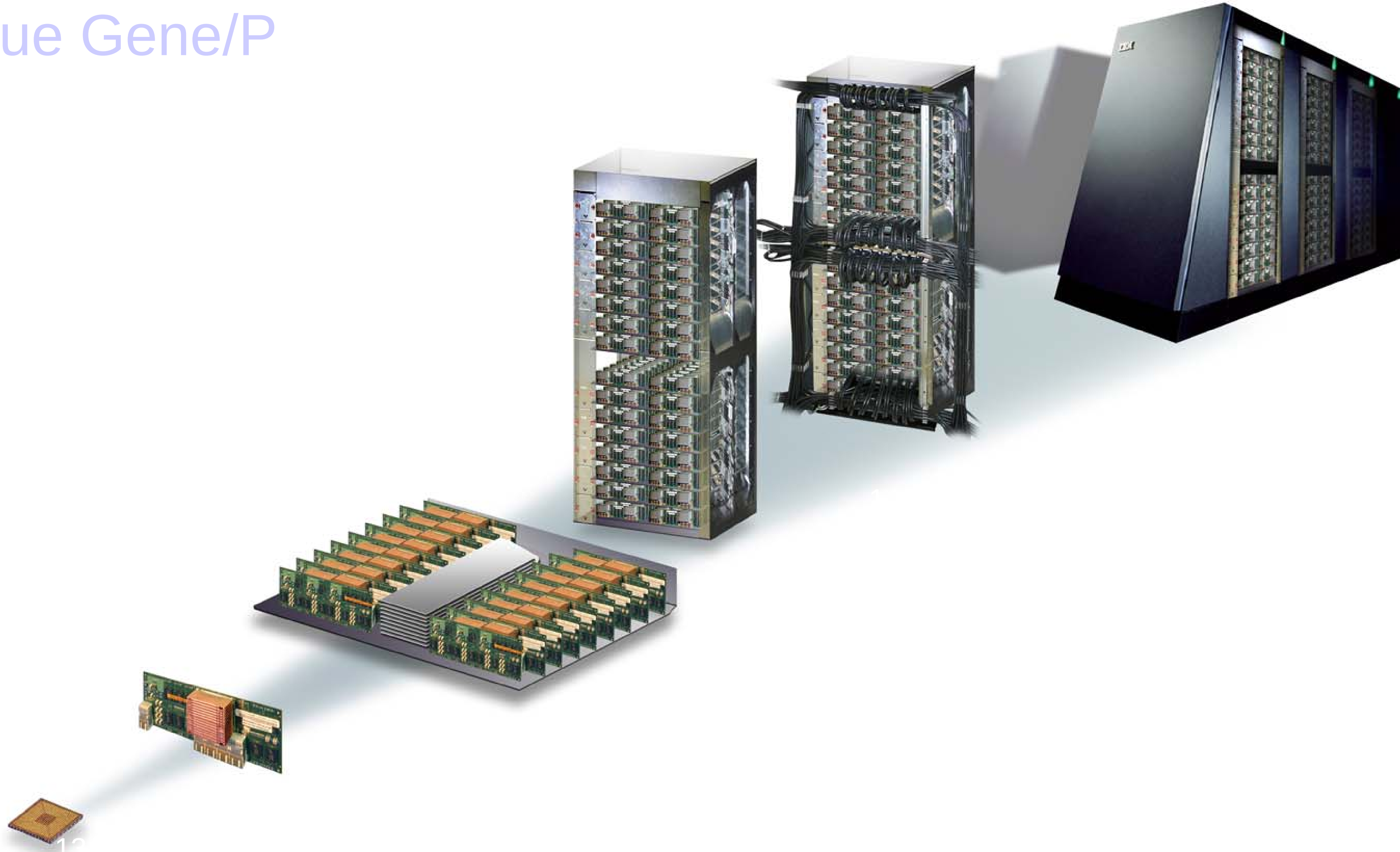


# BG/ASF : Accelerating Unix Utilities in Active GPFS

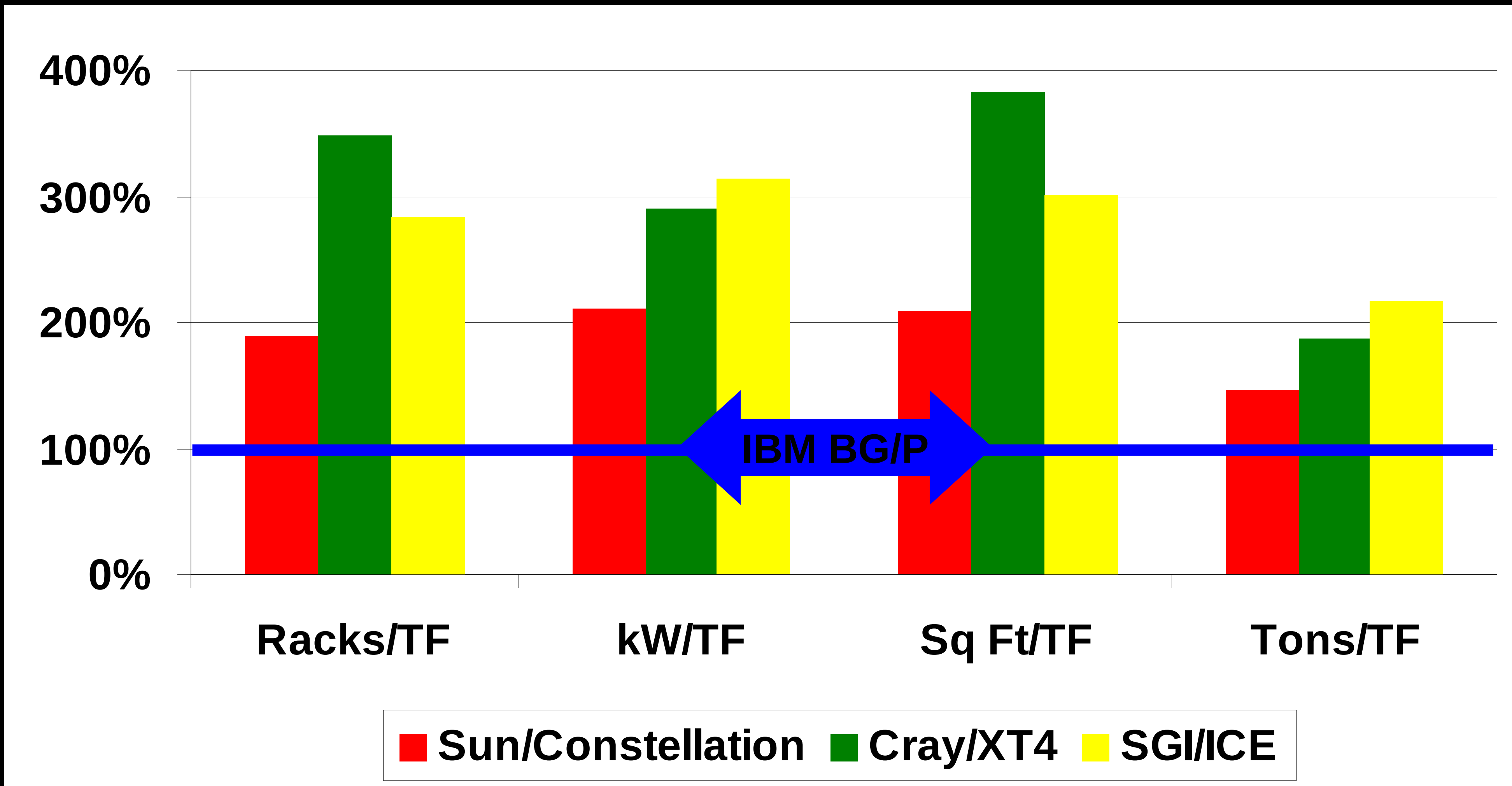
ASF Hosts
- mount the BG as a GPFS File System
- execute storage embedded applications



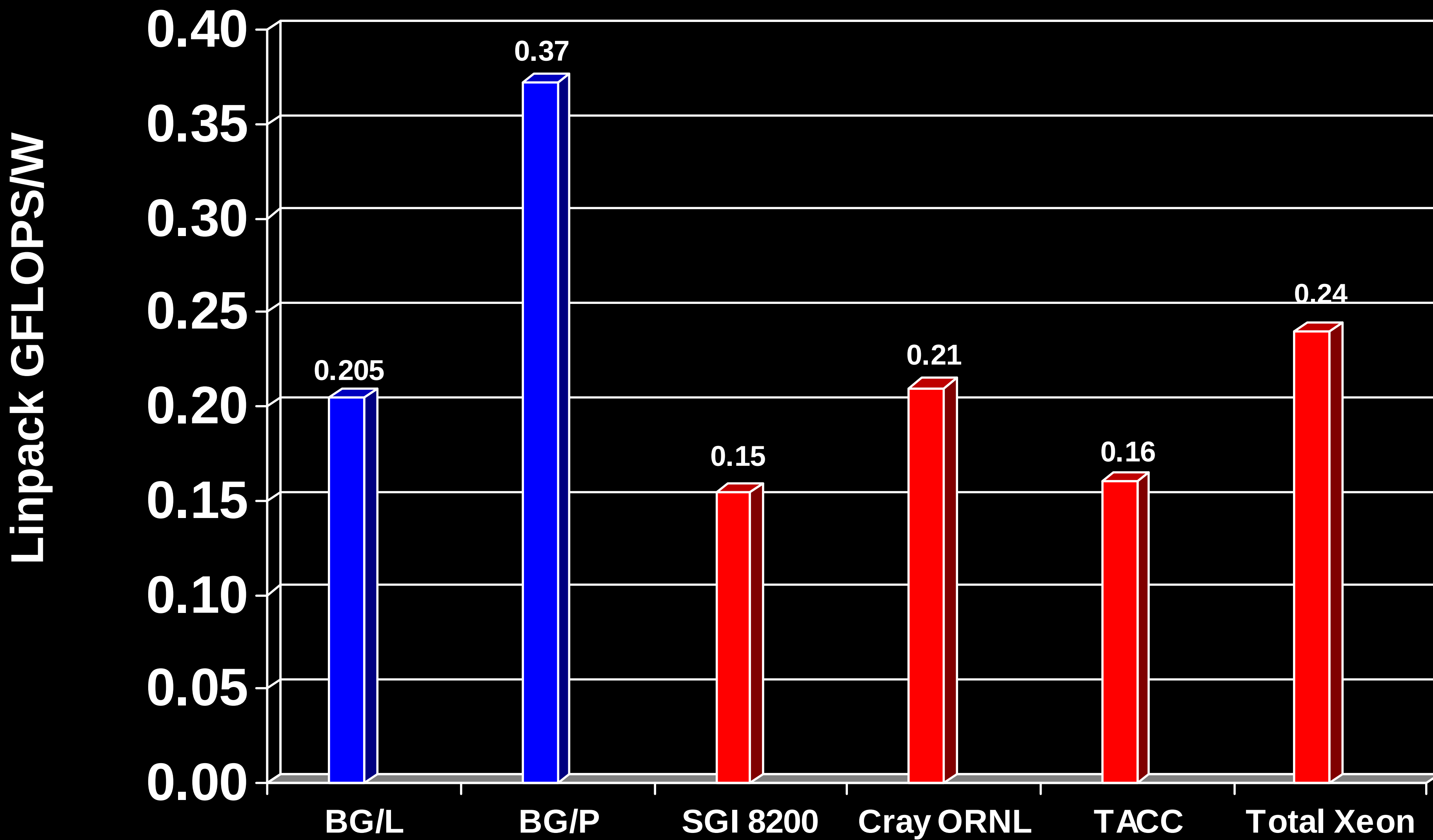
# Blue Gene/P



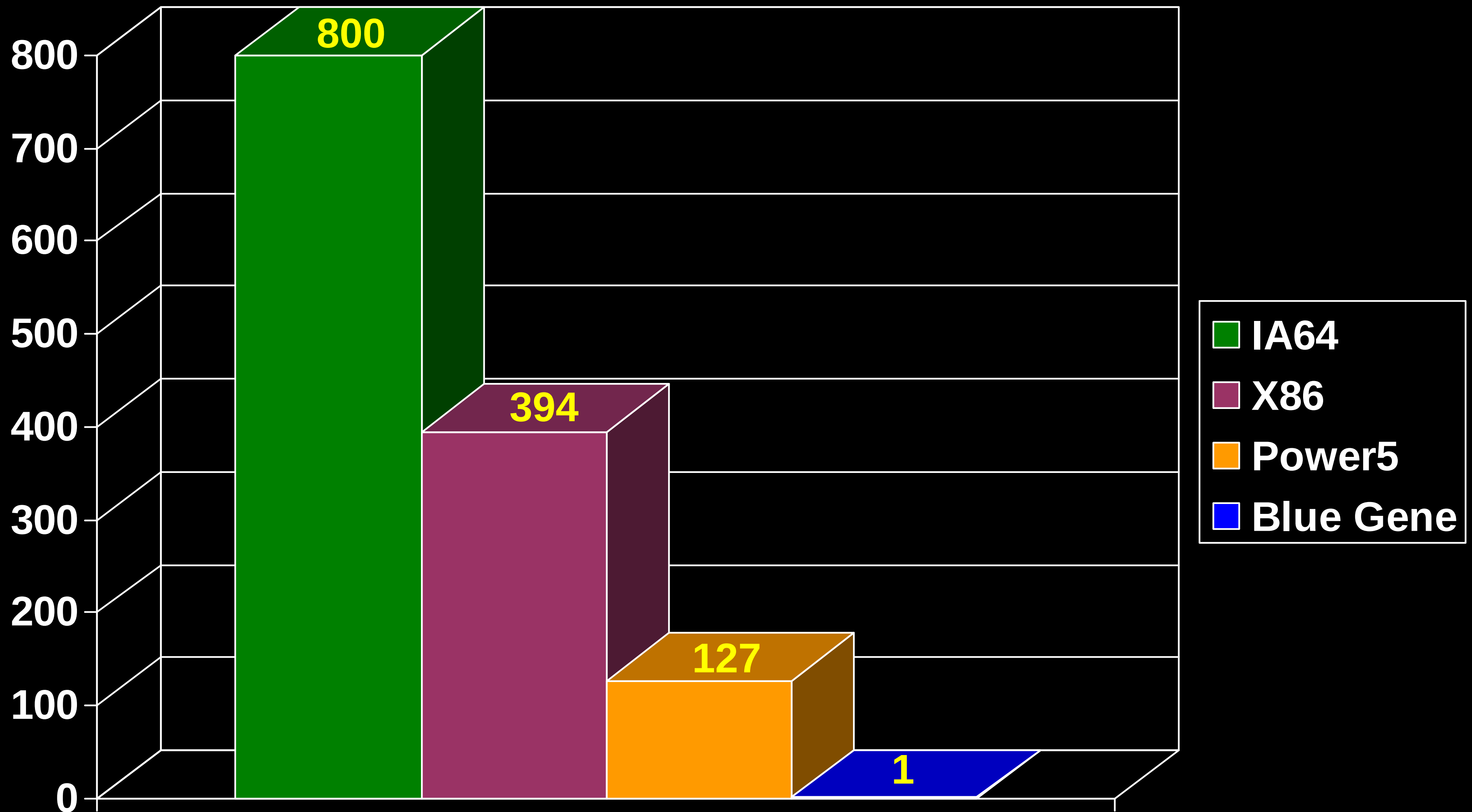
# Relative power, space and cooling efficiencies (Published specs per peak performance)



# Green 500


Linpack GFLOPS/W
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.205
0.37
0.15
0.21
0.16
0.24
BG/L
BG/P
SGI 8200
Cray ORNL
TACC
Total Xeon

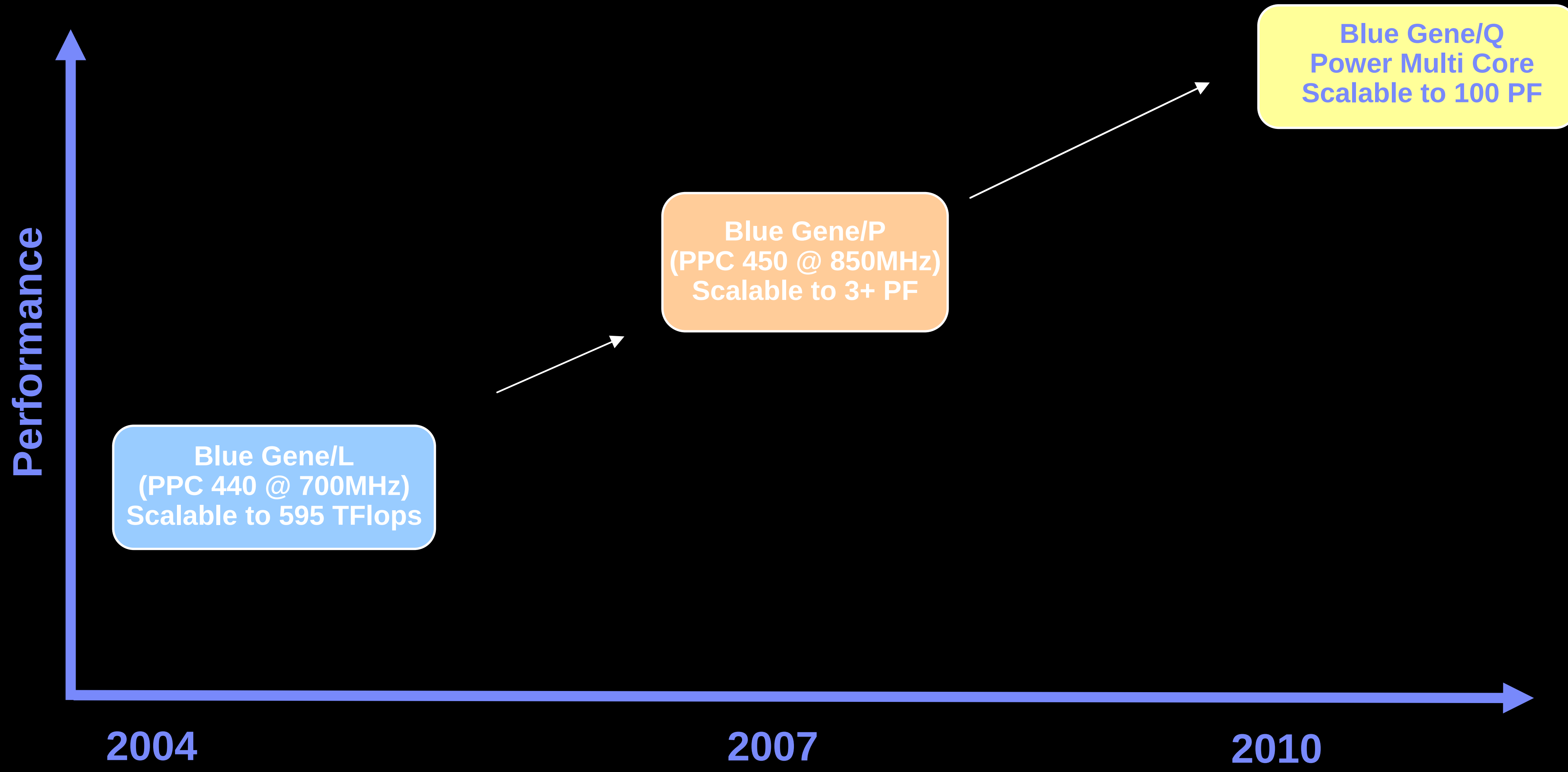
# Failures per Month per @ 100 TFlops (20 BG/L racks)
## unparalleled reliability




*Results of survey conducted by Argonne National Lab on 10 clusters ranging from 1.2 to 365 TFlops (peak); excluding storage subsystem, management nodes, SAN network equipment, software outages*

# Blue Gene Technology Roadmap




Note: All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.