A Perspective on The Path Forward

(Why I'm not *too* worried about Exascale)

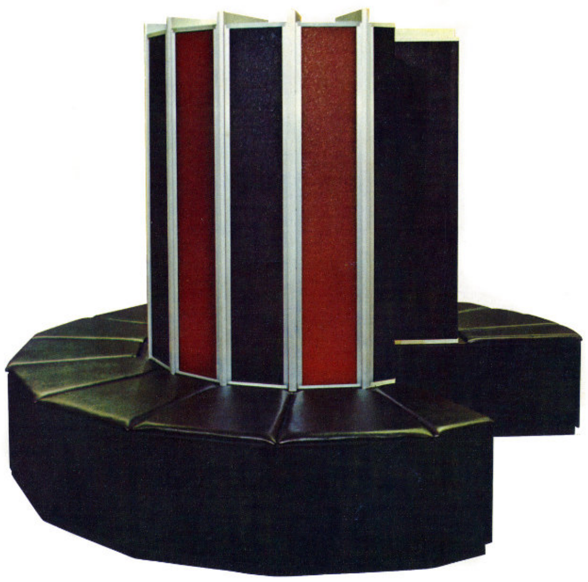
Steve Scott

Cray CTO

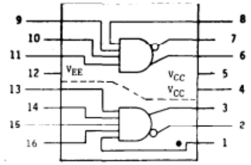
SOS20

March 23, 2016

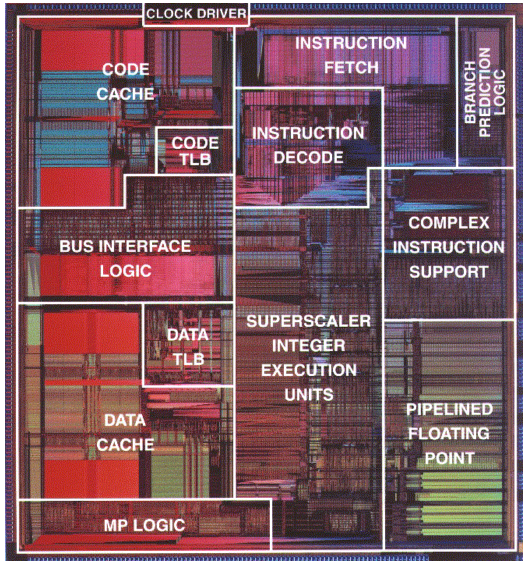
Technology Drives Architecture



Cray 1, 1976

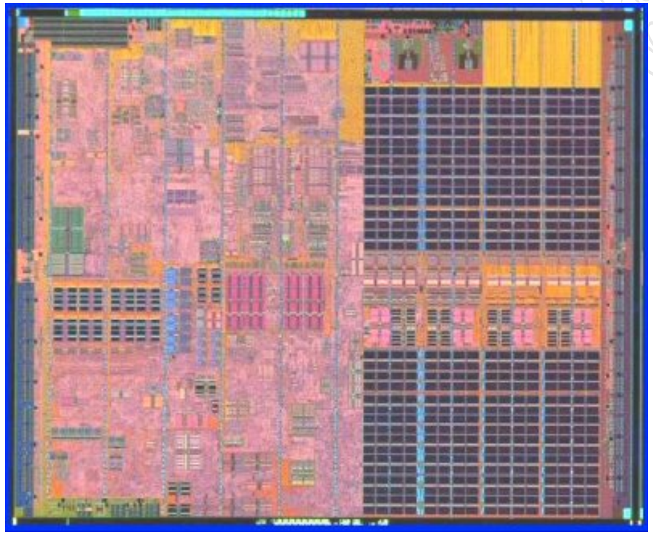


- ECL 5/4 NAND gate ICs (95%)
- 75K gates. (3400 PCBs!)
- RISC design
- Vector ISA
- Memory latency 11 clocks



Intel Pentium, 1993

- CMOS VLSI IC
- 3M transistors
- CISC design
- Scalar ISA
- Deep pipelines, complex predictions



Intel Pentium 4 Cedar Mill, 2006

- 184M transistors!
- *Very* CISC design
- 31-stage pipeline
- 3.6 GHz in 65nm
- Last of its breed….

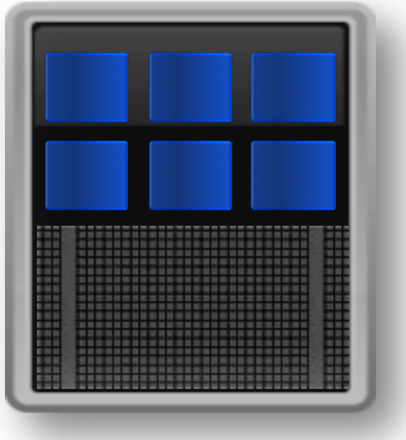
And then Dennard scaling ended…

Power constrained

Communication much more expensive than Computation

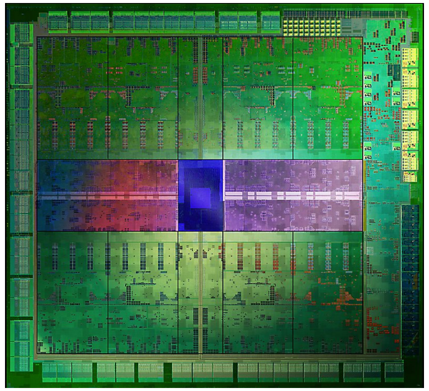
New Processor Landscape

Driven by power efficiency



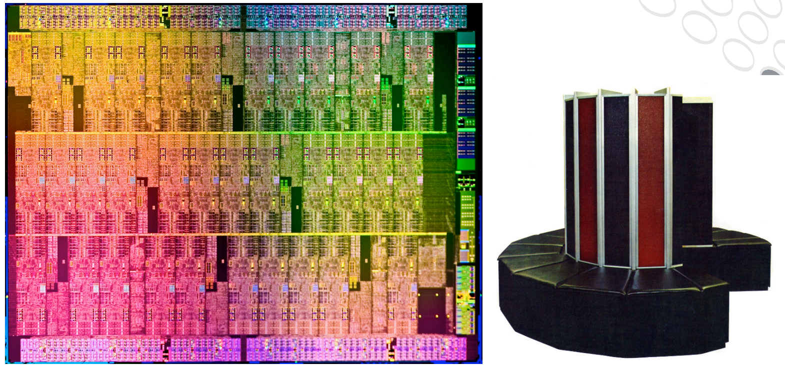
Multicore

Stop making it worse



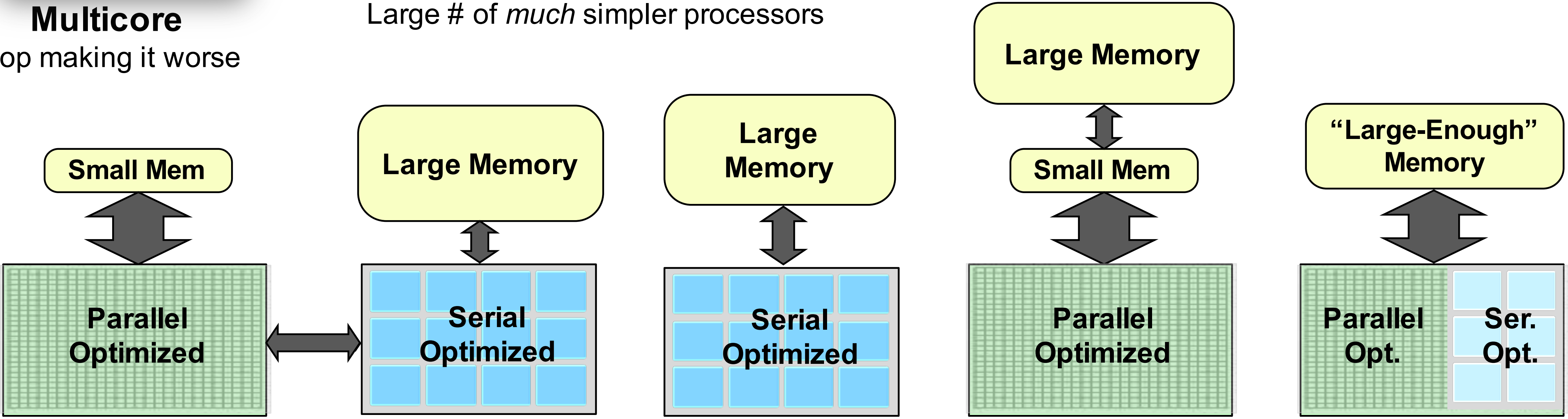
GPU computing (Nvidia Kepler)

Large # of *much* simpler processors



Vector Computing (Intel Xeon Phi)

Parallelism with low complexity & control overhead



Small Mem

Parallel Optimized

Large Memory

Serial Optimized

Large Memory

Serial Optimized

Large Memory

Small Mem

Parallel Optimized

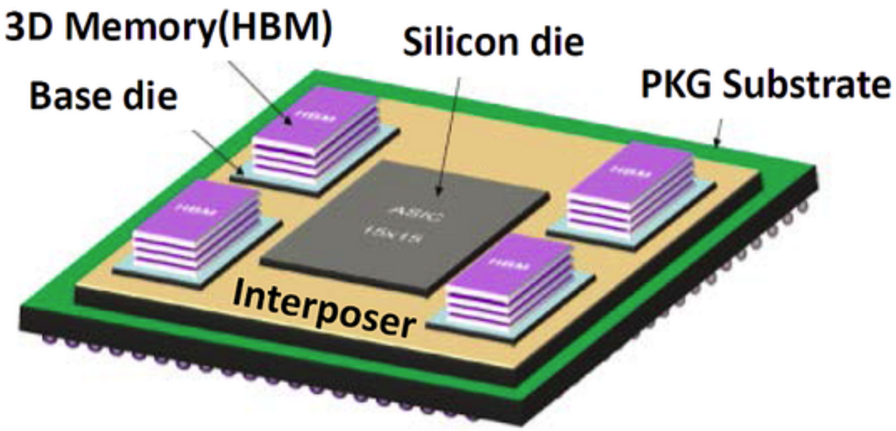
“Large-Enough” Memory

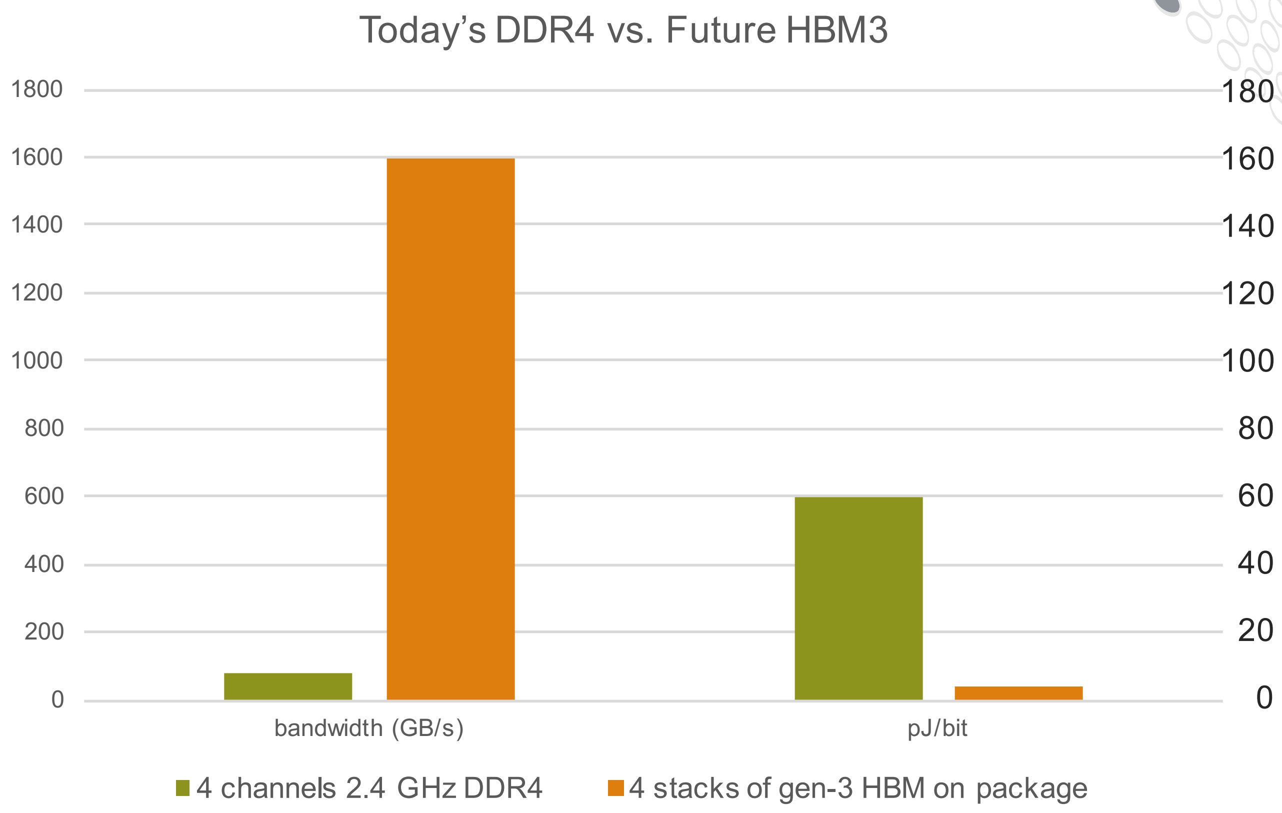
Parallel Opt.

Ser. Opt.

On-Package Memory Can Restore Balance

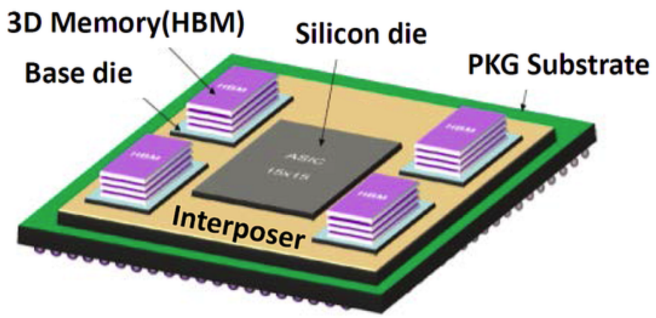
- Standard DDR memory BW has not kept pace with CPUs
- Expect processors to adopt stacked, on-package memory
- HBM:
  - 10x higher BW, 10x less energy/bit
  - Much lower latency
  - Costs ~2x DDR4 per bit
  - JDEC standard with multiple sources



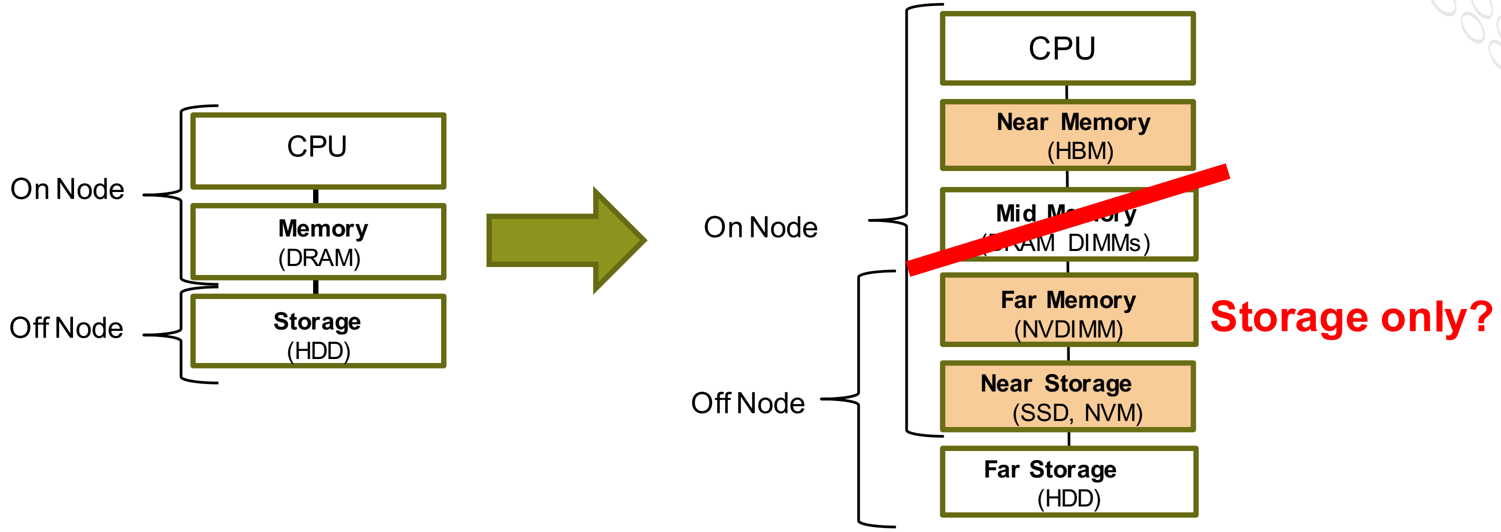


May drive us to smaller, simpler nodes that are balanced with on-package memory

Deeper Memory and Storage Hierarchy



Node memory moving on package



On Node

Off Node

CPU

Memory (DRAM)

Storage (HDD)

On Node

Off Node

CPU

Near Memory (HBM)

~~Mid Memory (DRAM DIMMs)~~

Far Memory (NVDIMM)

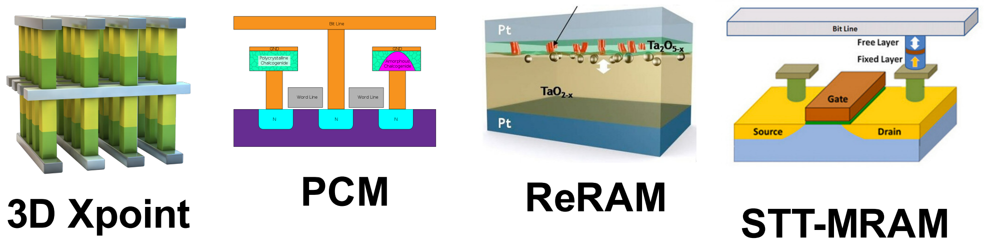
Storage only?

Near Storage (SSD, NVM)

Far Storage (HDD)

Cold storage moving to disk

Primary storage moving to Flash



3D Xpoint

PCM

ReRAM

STT-MRAM

New technologies coming to bridge memory-Flash gap

Storage Will Scale

- **APEX requirement: Time to checkpoint 80% memory < (0.005)*JMTTI**
  - Extrapolate to Exascale sytem:
  - Assume saving 80% of 32 PB of memory and a JMTTI of 10 hours
    ⇒ requires checkpoint bandwidth of ~150 TB/s (doable with distributed Flash)
  - Primary resiliency issue is dealing with *undetected* errors…

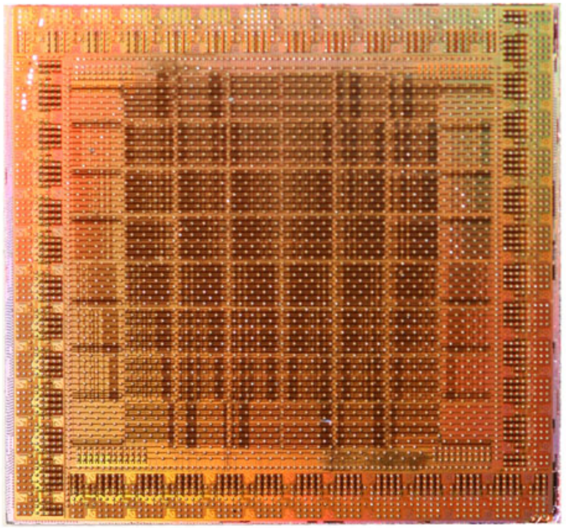
- **Storage latencies dropping faster than compute increasing**
  - Flash O(100) faster than disk
  - NVRAM is O(100) faster than Flash

- **But there's lots of work to do on storage *architecture*..**
  - Reducing software overheads for Flash and NVRAM timescale
  - Metadata scaling and resiliency (relax Posix consistency?)
  - Namespace flexibility
  - Support for non-POSIX file systems (KVS, NoSQL, Spark RDDs, etc.)

Cost- and Power-Efficient Networks

- **Cray pioneered the use of high radix routers in HPC**
  - Became optimal due to **technology shift**
    - Faster signaling permits narrower links
  - Reduced network diameter (number of hops)

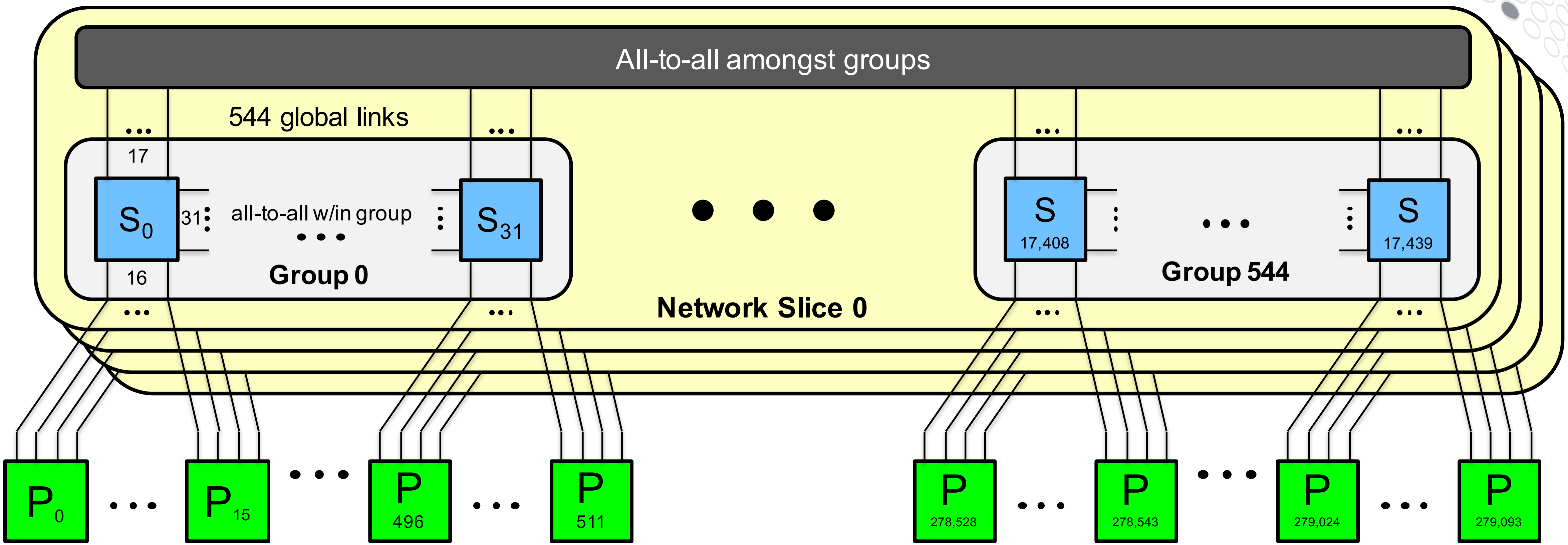
    ⇒ Lower latency and cost
  - But… higher radix network require longer cable lengths



First 64 port router
Cray X2 (2005)

- **Optics enables longer cable lengths**
  - Now cost-effective above a few meters (and dropping)
  - Cost, bandwidth and power are insensitive to cable length
- **Future systems will based on hybrid, electrical-optical networks**
  - Cost-effective, scalable global bandwidth
  - Very low network diameter (small number of hops) ⇒ very energy efficient

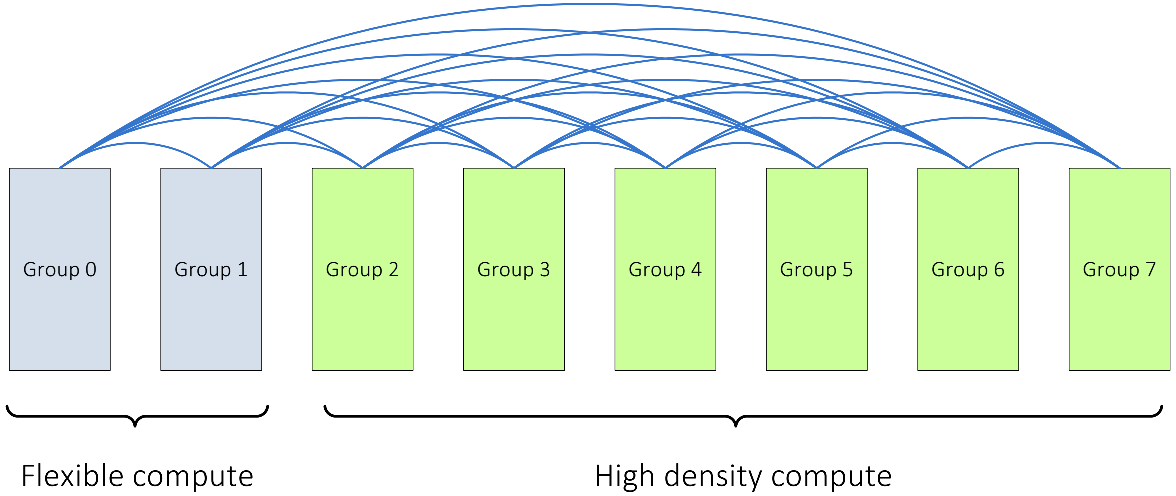
Example Dragonfly Network with a 64-port Switch



- Scales to 279K endpoints, with a network diameter of 3 hops!
- Only a *single* hop over a long (optical) link
- Narrow links allow sliced network for configurable bandwidth

Next-Gen Shasta System Infrastructure

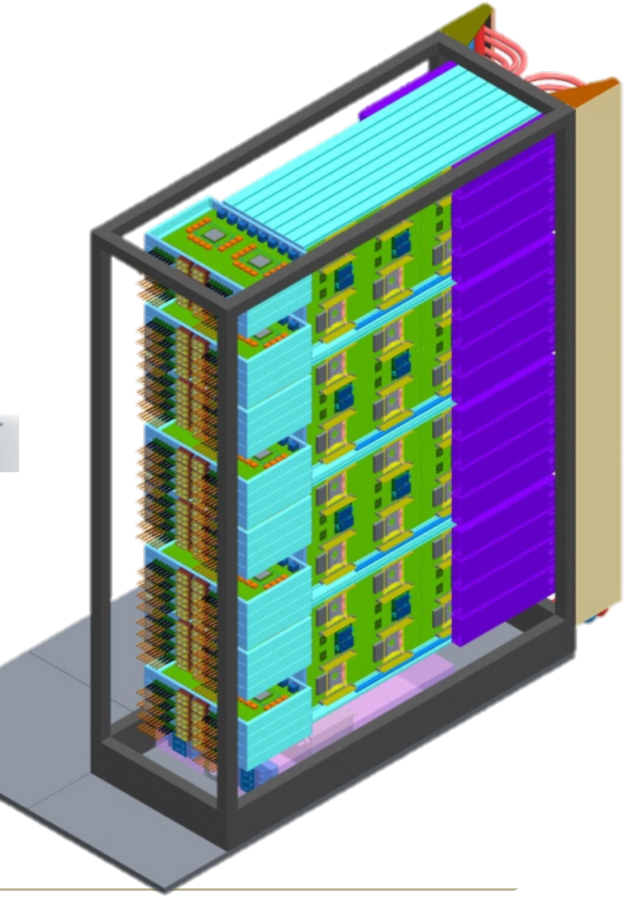
- **Single system with choice of:**
  - Cabinet type and cooling infrastructure
  - Processor type
  - Software stack
  - Interconnect
- **Extensible to Exascale and Beyond**
  - Power & cooling headroom
  - Network and processor configurability



Group 0 Group 1 Group 2 Group 3 Group 4 Group 5 Group 6 Group 7

Flexible compute

High density compute



Summary of Future Machines

- **Computers are not getting faster… just wider**
  - O(EF) with O(GHz) clocks → O(B) way parallelism!
- ***Vertical* locality much more important than *horizontal* locality**

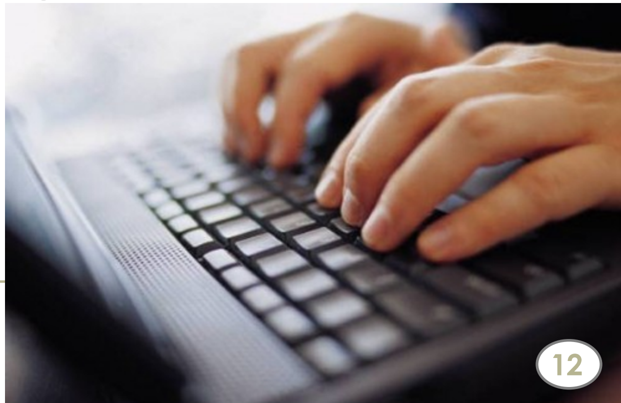
Dimension	Latency Hit	Bandwidth Hit	Energy Hit
Within node	~200x	~200x	> 500x
Across nodes	~25x	~8x	~5x

** If include local NVM, within node grows, across nodes shrinks*

- **Parallelism is multi-dimensional (and heterogeneous?)**
  - Vectorization + threading + multi-node
  - Processors optimized for serial performance ***or*** power efficiency (not both)
- **Interconnects won't look that different than today**

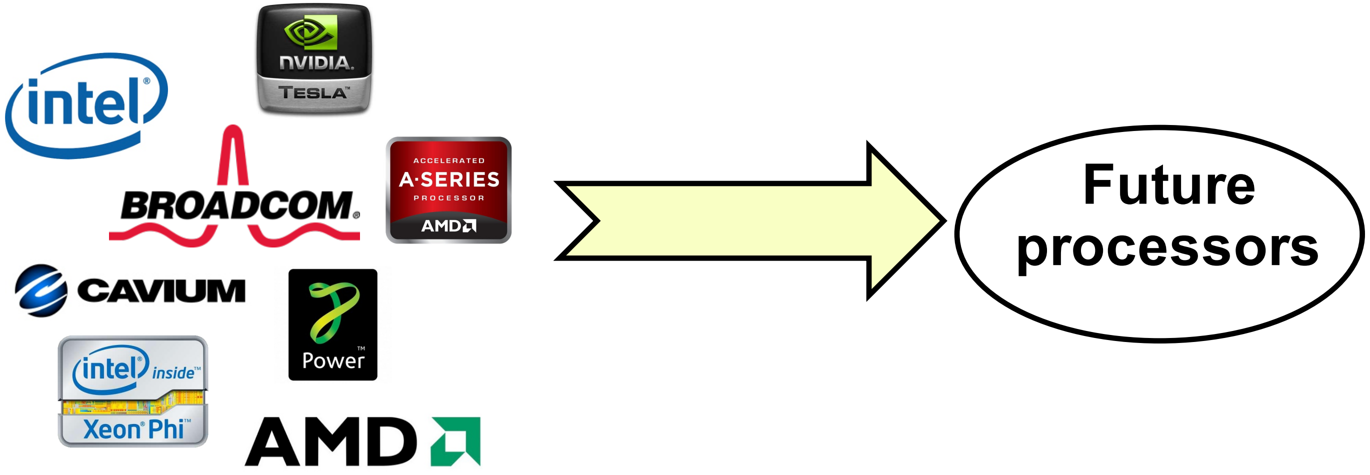
Implications for Programmers

- **May need to move to more threading on the node**
  - All-MPI often won't deliver maximum performance
- **Must vectorize low-level loops**
  - 8-30x performance improvement on array operations
- **Must avoid serial scalar code**
  - Inherently slower and less power-efficient
  - On "accelerated" nodes, either
    - creates traffic between accelerator and host, or
    - runs 3-4x slower than on a serial-optimized core
- **Must pay a *lot* more attention to locality within node**
  - Think about data placement and movement
  - Consider "sub-optimal" algorithms that limit data motion



Would like to code for future machines in a portable way

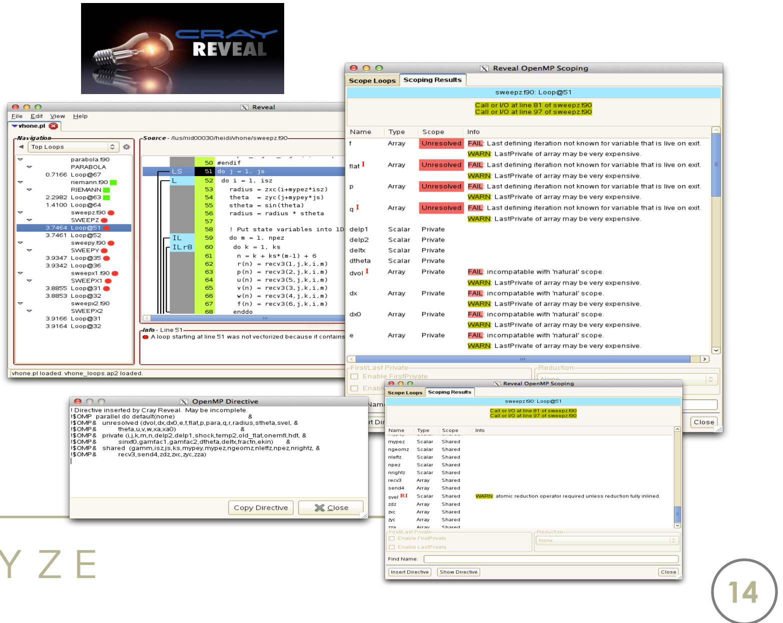
- Spatial and Temporal Portability



- Separation of labor
  - Programmer *exposes* parallelism and locality
  - Compiler, tools, and runtime map onto specific hardware
  - Optimized libraries for various platforms

Bold Prediction:

- **Future HPC Programming Model: MPI + OpenMP**
- **Can we make this easier?**
  - Threading, vectorization, data placement
- **Recent poll at NERSC found 80% of apps use single level of parallelism**
- **Why & when to convert to hybrid programming model?**
  - When code becomes network bound
  - Load balancing and synchronization overheads become large
  - Excessive memory used by straight MPI
  - To take advantage of hybrid compute nodes
- **Programming tools are going to be critical**
  - Exposing parallelism (especially higher in call chain)
  - Data placement and movement in the memory hierarchy



```
parabola.f90
  PARABOLA
0.7166  Loop@67
riemann.f90
  RIEMANN
2.2982  Loop@63
1.4100  Loop@64
sweepz.f90
  SWEEPZ
3.7464  Loop@51
3.7461  Loop@52
sweepy.f90
  SWEEPY
3.9347  Loop@35
3.9342  Loop@36
sweepx1.f90
  SWEEPX1
3.8855  Loop@31
3.8853  Loop@32
sweepx2.f90
  SWEEPX2
3.9166  Loop@31
3.9164  Loop@32
```

```
50 #endif
51 do j = 1, js
52  do i = 1, isz
53    radius = zxc(i+mypez*isz)
54    theta  = zyc(j+mypey*js)
55    stheta = sin(theta)
56    radius = radius * stheta
57
58    ! Put state variables into 1D
59    do m = 1, npez
60     do k = 1, ks
61      n = k + ks*(m-1) + 6
62      r(n) = recv3(1,j,k,i,m)
63      p(n) = recv3(2,j,k,i,m)
64      u(n) = recv3(5,j,k,i,m)
65      v(n) = recv3(3,j,k,i,m)
66      w(n) = recv3(4,j,k,i,m)
67      f(n) = recv3(6,j,k,i,m)
68     enddo
```

A loop starting at line 51 was not vectorized because it contains

vhone.pl loaded. vhone_loops.ap2 loaded.

sweepz.f90: Loop@51

Call or I/O at line 81 of sweepz.f90
Call or I/O at line 97 of sweepz.f90

```
Name     Type    Scope       Info
f        Array   Unresolved  FAIL: Last defining iteration not known for variable that is live on exit.
                             WARN: LastPrivate of array may be very expensive.
flat I   Array   Unresolved  FAIL: Last defining iteration not known for variable that is live on exit.
                             WARN: LastPrivate of array may be very expensive.
p        Array   Unresolved  FAIL: Last defining iteration not known for variable that is live on exit.
                             WARN: LastPrivate of array may be very expensive.
q I      Array   Unresolved  FAIL: Last defining iteration not known for variable that is live on exit.
                             WARN: LastPrivate of array may be very expensive.
delp1    Scalar  Private
delp2    Scalar  Private
deltx    Scalar  Private
dtheta   Scalar  Private
dvol I   Array   Private     FAIL: incompatable with 'natural' scope.
                             WARN: LastPrivate of array may be very expensive.
dx       Array   Private     FAIL: incompatable with 'natural' scope.
                             WARN: LastPrivate of array may be very expensive.
dx0      Array   Private     FAIL: incompatable with 'natural' scope.
                             WARN: LastPrivate of array may be very expensive.
e        Array   Private     FAIL: incompatable with 'natural' scope.
                             WARN: LastPrivate of array may be very expensive.
```

```
! Directive inserted by Cray Reveal.  May be incomplete.
!$OMP  parallel do default(none)                              &
!$OMP&   unresolved (dvol,dx,dx0,e,f,flat,p,para,q,r,radius,stheta,svel, &
!$OMP&          theta,u,v,w,xa,xa0)                           &
!$OMP&   private (i,j,k,m,n,delp2,delp1,shock,temp2,old_flat,onemfl,hdt, &
!$OMP&          sinxf0,gamfac1,gamfac2,dtheta,deltx,fractn,ekin)      &
!$OMP&   shared  (gamm,isz,js,ks,mypey,mypez,ngeomz,nleftz,npez,nrightz, &
!$OMP&          recv3,send4,zdz,zxc,zyc,zza)
```

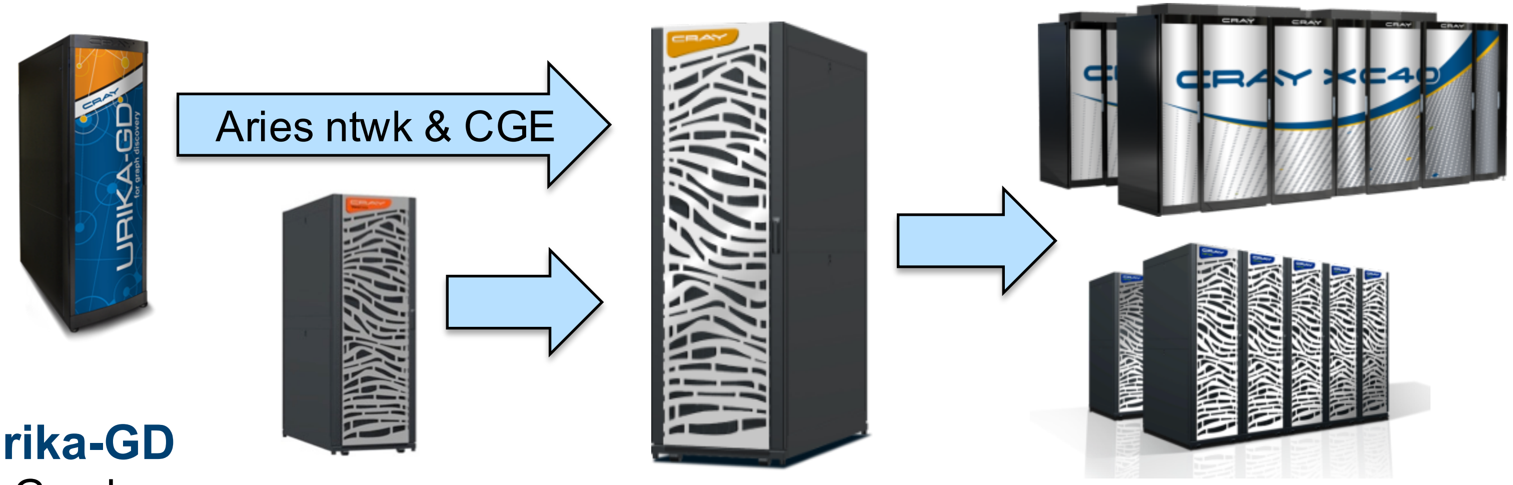
sweepz.f90: Loop@51

Call or I/O at line 81 of sweepz.f90
Call or I/O at line 97 of sweepz.f90

```
Name     Type    Scope   Info
mypez    Scalar  Shared
ngeomz   Scalar  Shared
nleftz   Scalar  Shared
npez     Scalar  Shared
nrightz  Scalar  Shared
recv3    Array   Shared
send4    Array   Shared
svel RI  Scalar  Shared  WARN: atomic reduction operator required unless reduction fully inlined.
zdz      Array   Shared
zxc      Array   Shared
zyc      Array   Shared
zza      Array   Shared
```

Beyond Classic HPC

Merging of HPC and Data Analytics

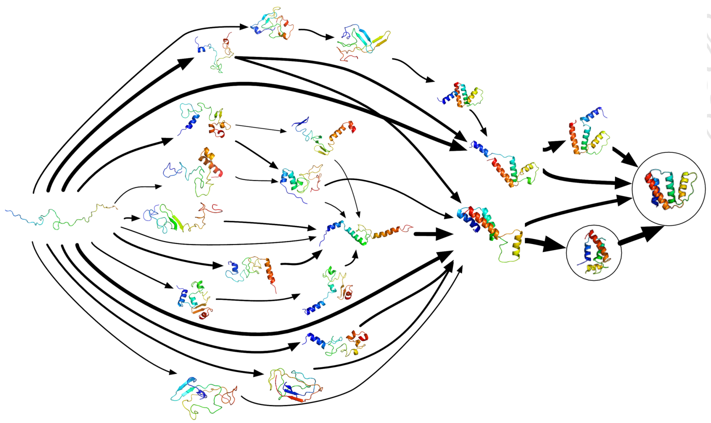


Aries ntwk & CGE

Urika-GD
Graph
analytics

Urika-XA
BDAS
(Hadoop, Spark)
NoSQL

"Athena"
BDAS +
Graph analytics +
HPC



HPC + Analytics workflows

HPC underneath the covers

Why combine HPC and Analytics solutions in a single box?

Thank You

Questions?