Architectural constraints required to attain 1 Exaflop/s for scientific applications

Abhinav Bhatele, Pritish Jetley, Hormozd Gahvari, Lukasz Wesolowski, William D. Gropp, Laxmikant V. Kale

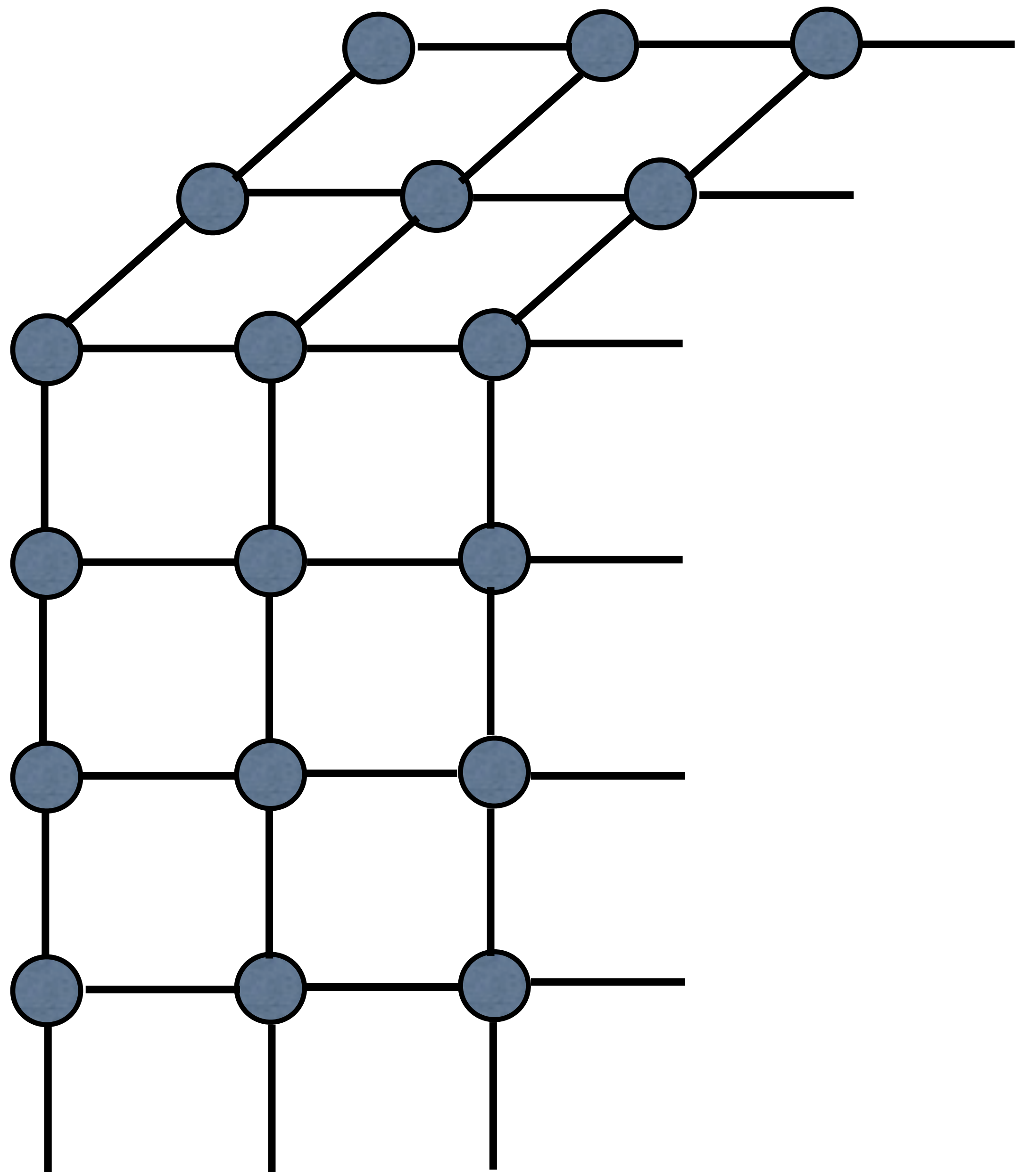
Department of Computer Science
University of Illinois

Motivation

- First Teraflop/s computer (ASCI Red, 1997), first Petaflop/s computer (RoadRunner, 2008), Exaflop/s 2018 ?
- Hardware challenges: power/energy, memory, communication
- Software challenges: algorithms and implementations that will scale
- Architectural features to attain 1 Exaflop/s ?

A possible exascale machine

- 2^{20} = 1,048,576 nodes
- 2^{10} cores per node
- 10 Gflop/s cores, time to compute a flop, t_c = 0.1 ns
- 10.74 Exaflop/s peak performance



Modeling methodology

- Estimate the floating point calculations/operations per iteration,

$$T_{comp} = \frac{1}{\eta} \times f(N, P_c) \times n \times t_c$$

- Time for communication based on number and size of messages

$$T_{comm} = M \times (t_s + h(N, P_c) \times t_w)$$

- Using total number of floating point operations and time per iteration, $\frac{flops}{T} > 10^{18}$

PPL UIUC

Applications

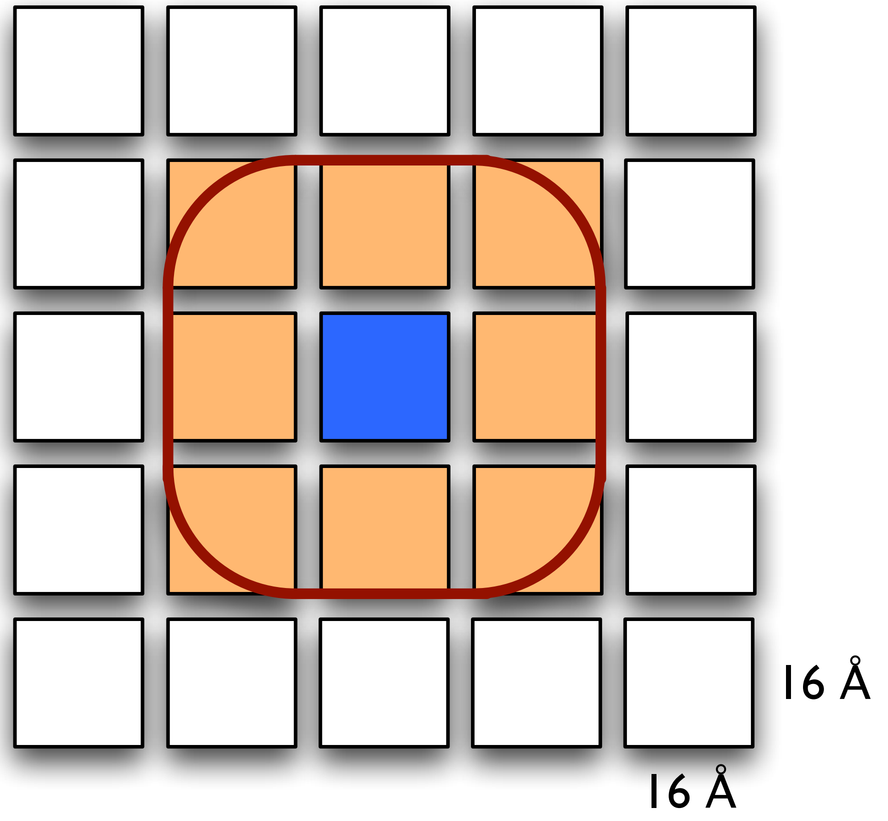
- Molecular Dynamics
  - Short-range forces, spatial decomposition
- Cosmological Simulations
  - Tree algorithms
- Unstructured grid problems
  - Finite element solvers

Molecular Dynamics

- Spatial decomposition

Algorithm 1 Computation in one time step of MD

Receive atoms from neighboring processors
for $i = 1$ to N_p **do**
  for $j = 1$ to N_i **do**
    if atoms are within cutoff radius, r_c **then**
      Compute forces on pairs of atoms
    end if
  end for
end for
Update atom positions and velocities



Weak scaling of MD

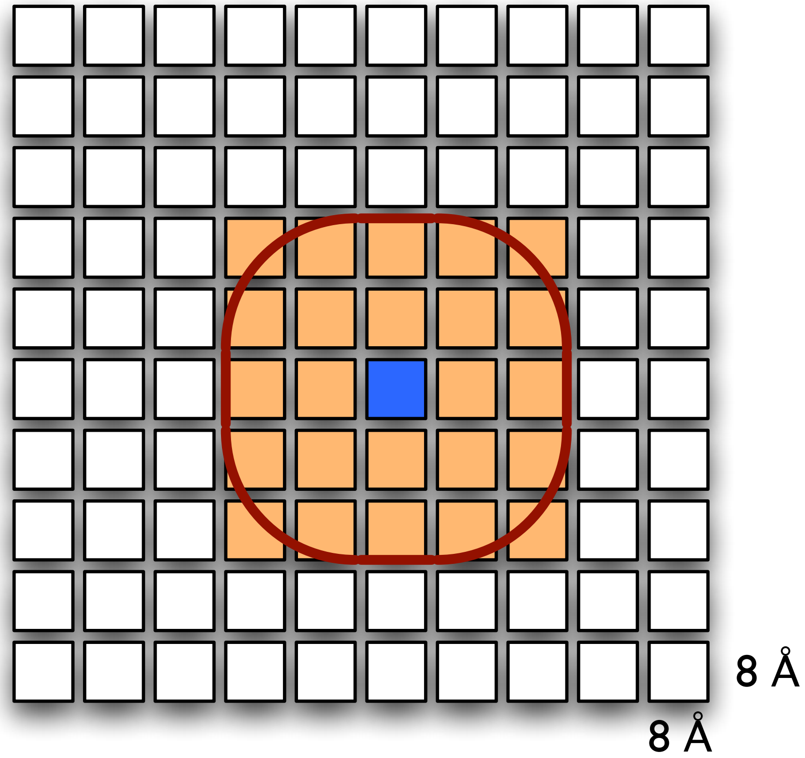
- Size of molecular system = 100 * 2^{30} = 107 billion atoms
- Number of floating point operations = 33547 * N

$$\frac{flops}{T} > 10^{18}$$
$$\frac{33547 \times N}{10^{18}} > T$$

- Putting N = 100 * 2^{30},

$$T < 3.6 \times 10^{-3}$$

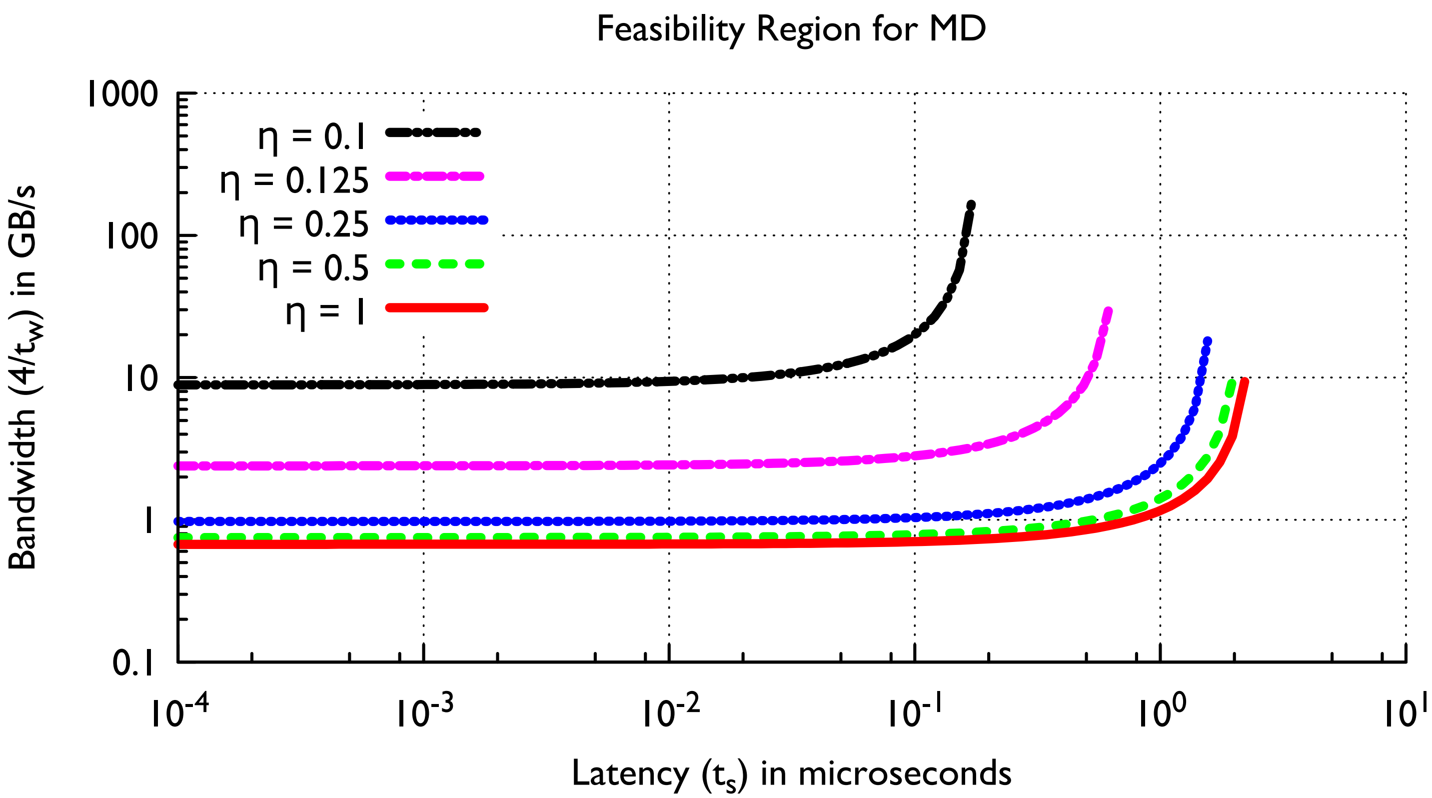
- 100 atoms per cell
- Split the cells in two of the three dimensions
- Each cell communicates with 5*5*3 = 75 other cells
- For a block of 8*8*16 cells placed on a node only the ones on the boundary communicate inter-node



Inferring network parameters

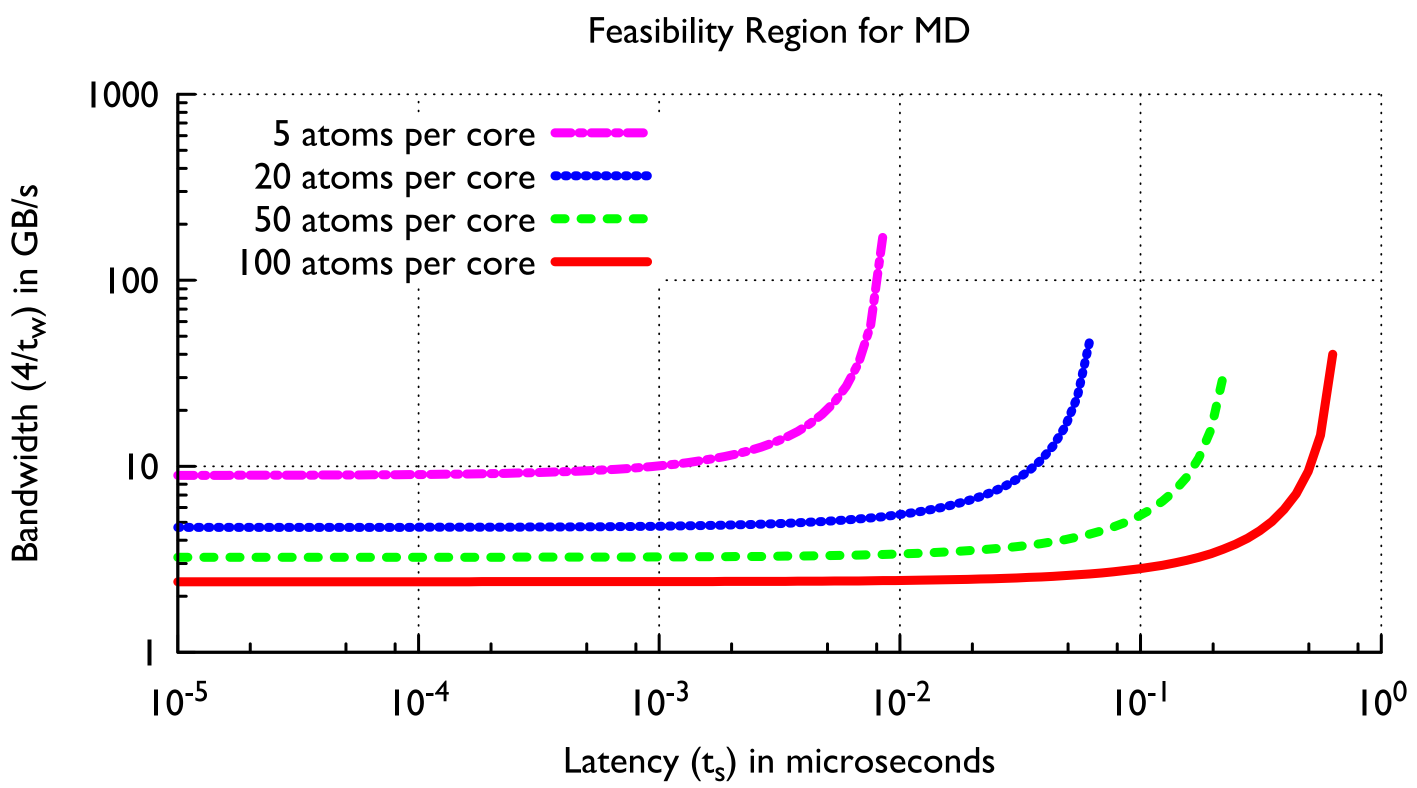
$$\frac{1}{\eta} \times \frac{N}{P_c} \times 33547 \times t_c + 1376 \times \left(t_s + \frac{N}{P_c} 4 t_w \right) < 3.6 \times 10^{-3}$$

$$t_s + 400 t_w < 2.62 \times 10^{-6} - \frac{1}{\eta} \times 2.44 \times 10^{-7}$$



Smaller problem sizes

# Atoms	Atoms/core	Time (ms)
107 billion	100	3.602
53.6 billion	50	1.801
21.5 billion	20	0.720
5.4 billion	5	0.180



Computational Cosmology

- Several approaches to computing trajectories of bodies under gravitational attraction
  - Direct, all-pairs
  - Tree-based, approximate methods
  - Structured grid/AMR methods
- We consider locality-aware tree codes

Modeling problem size

- What problems will be of interest given an exascale-level machine
- Extrapolate from current state-of-the-art simulations
- About 2^{13} particles are required per core for good parallel efficiency
- Given O(N log N)/P work per core, about 6350 particles per core are needed at exascale (total 6.8 trillion)

Barnes-Hut computation

- How many flops per iteration are there with 6.8 trillion particles?
- Analyze algorithm:
  - Domain decomposition => distributed spatial tree
  - Every processor core gets a number of leaves
  - For each leaf l, Traverse(l, root)

Tree traversal

```
Traverse(leaf l, node n) {
  if(IsLeaf(n)) {
    LeafForces(l, n);
  }
  else if(Side(n)/|r(n)-r(l)| < Θt)
{
    CellForces(l, n);
  }
  else {
    foreach(node c in Children(n)) {
       Traverse(l, c);
  }
}
```

Tree traversal

If d = Depth(n), then Side(n) = $c/2^d$

For Θ_t >= 0.5, a maximum of E(d) = 33 of 125 neighboring cells expanded at depth d

```
Traverse(leaf l, node n) {
  if(IsLeaf(n)) {
    LeafForces(l, n);
  }
  else if(Side(n)/|r(n)-r(l)| < Θt)
{
    CellForces(l, n);
  }
  else {
    foreach(node c in Children(n)) {
       Traverse(l, c);
  }
}
```

Tree traversal

If d = Depth(n), then Side(n) = $c/2^d$

For $\Theta_t >= 0.5$, a maximum of E(d) = 33 of 125 neighboring cells expanded at depth d

Number of CellForces invocations per leaf:

8*E(d)-E(d-1) = 231

```
Traverse(leaf l, node n) {
  if(IsLeaf(n)) {
    LeafForces(l, n);
  }
  else if(Side(n)/|r(n)-r(l)| < Θt)
{
    CellForces(l, n);
  }
  else {
    foreach(node c in Children(n)) {
       Traverse(l, c);
  }
}
```

Tree traversal

Number of LeafForces invocations per leaf:

33

If d = Depth(n), then Side(n) = $c/2^d$

For Θ_t >= 0.5, a maximum of E(d) = 33 of 125 neighboring cells expanded at depth d

Number of CellForces invocations per leaf:

8*E(d)-E(d-1) = 231

```
Traverse(leaf l, node n) {
  if(IsLeaf(n)) {
    LeafForces(l, n);
  }
  else if(Side(n)/|r(n)-r(l)| < Θt)
{
    CellForces(l, n);
  }
  else {
    foreach(node c in Children(n)) {
       Traverse(l, c);
  }
}
```

Total computation

- Number of floating point operations per iteration,

$$312 \times 77 \times N \times \lg \frac{N}{B} + 38 \times 33 \times B \times N$$

- To attain a rate of 1 Exaflop/s,

$$\frac{24024 \times N \lg(N/B) + 1254 \times BN}{T} > 10^{18}$$

- T < 6.52s

Total communication

Total communication

- Could obtain communication from number of expansions E(l) for every level l

Total communication

- Could obtain communication from number of expansions E(l) for every level l
- However, cores on an SMP node can reuse remote data through software caching

Total communication

- Could obtain communication from number of expansions E(l) for every level l
- However, cores on an SMP node can reuse remote data through software caching
- Communication with remote data caching:
  - Each SMP node holds a cube of space
  - Cores holding particles near surface of cube request remote data - other cores reuse data
  - Find each SMP node's *halo* of requests at each level of tree

Communication analysis

Leaf level:

$$12n_b^2 + 36n_b + 8$$

1 level above leaves:

$$12(n_b/2)^2 + 36(n_b/2) + 8$$

2 levels above leaves:

$$12(n_b/4)^2 + 36(n_b/4) + 8$$

3 levels above leaves:

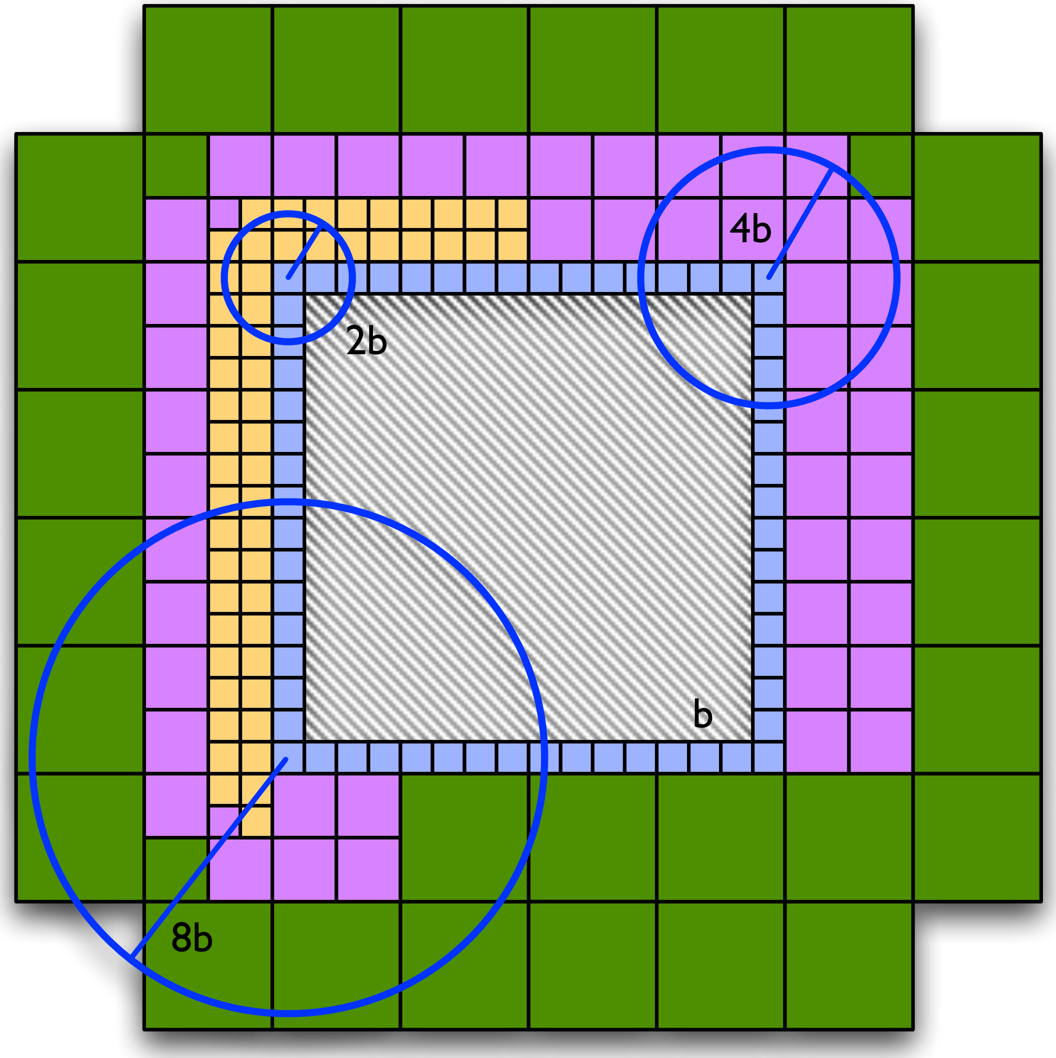
$$12(n_b/8)^2 + 36(n_b/8) + 8$$

…

Total:

$$C_1^{\text{cell}} = \sum_{i=0}^{\lg n_b} \left(12 \left(\frac{n_b}{2^i} \right)^2 + 36 \left(\frac{n_b}{2^i} \right) + 8 \right)$$

$$= 16n_b^2 + 72n_b + 8 \lg n_b - 32 \quad \text{cells}$$



Upper-level calls

Upper-level calls

- Previous reasoning valid as long as edge length of requested calls <= $c/(P_n)^{1/3}$

Upper-level calls

- Previous reasoning valid as long as edge length of requested calls <= $c/(P_n)^{1/3}$
- Use reasoning similar to calculation of E(l) to get number of larger, upper-level cells requested per SMP node,

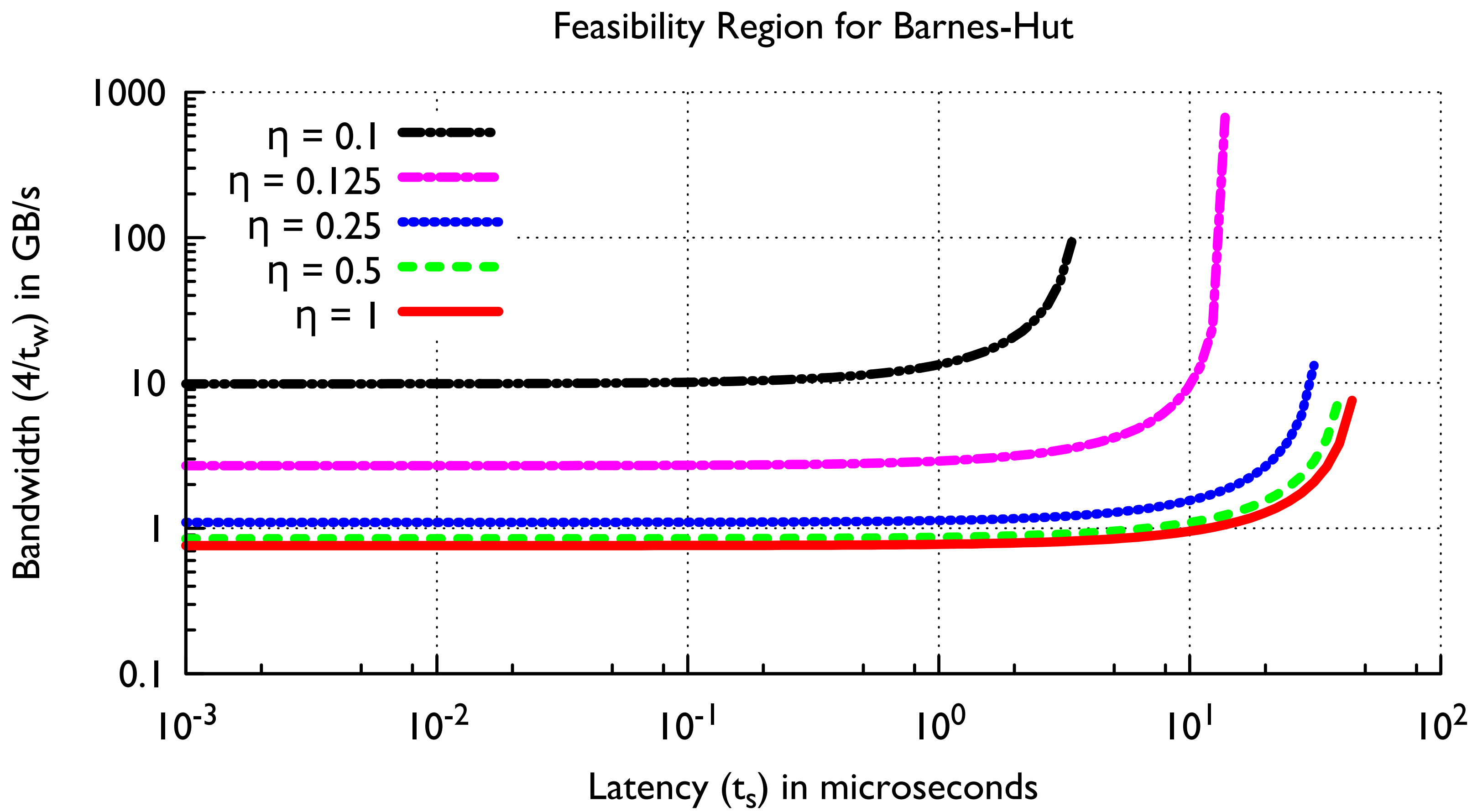
$$C_2^{\text{cell}} = 31\left(\frac{\lg P_n}{3} - 1\right) \quad \text{cells}$$

$$T_{comm} = 15946(t_s + 56t_w) + 93968(t_s + 100t_w)$$

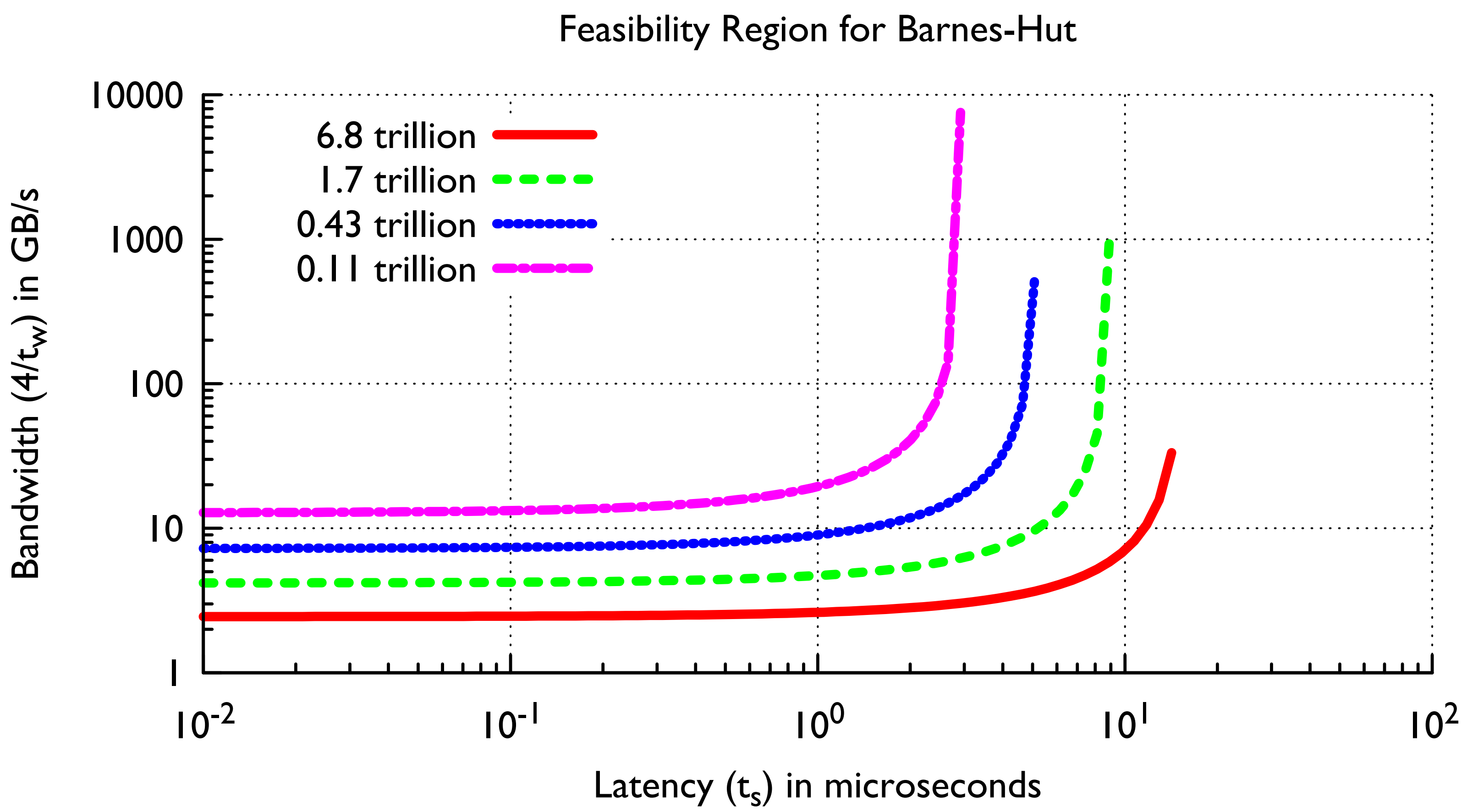
Inferring network parameters

$$\frac{6.52 \times 10^{18}}{P_c} \times \frac{t_c}{\eta} + (1.1 \times 10^5 t_s + 1.03 \times 10^7 t_w) < 6.52$$

$$t_s + 93.62 t_w < 59.2 \left(1 - \frac{0.093}{\eta}\right) \times 10^{-6}$$



Smaller problem sizes



Summary

- Modest communication requirements for MD and cosmology at exascale:
  - each communicated value used for large number of flops
- Smaller problem sizes lead to tighter constraints
- Current latency and bandwidth (XT5): ~4 µs, 9.6 GB/s
- Required: 1 µs latency and 10 GB/s bandwidth