General Speed Up Analysis

- Comparing Computer and Hadrian's Wall Cases

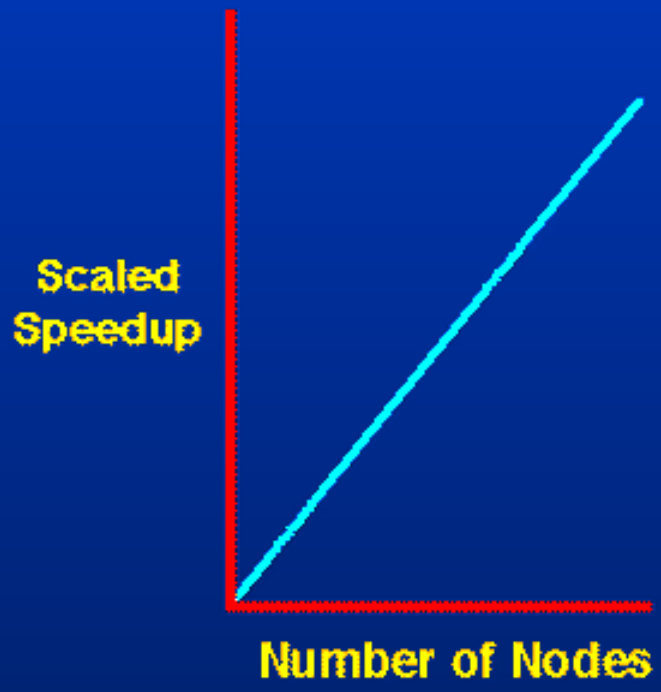
$$\text{Speedup } S = \varepsilon N$$

$$\varepsilon = 1 - \frac{\text{constant}}{n^{1/d}} \cdot \frac{t_{comm}}{t_{calc}}$$

General	*Hadrian's Wall*
n = Grain Size	n = number of bricks laid by each mason $n \propto l$
d = Problem Dimension	d = 1 for a one dimensional wall (d = 2 for laying tiles on floor of Hadrian's Palace)
t_{calc} = Time to do each calculation	Time to lay one brick
t_{comm} = Time to communicate unit of information between nodes	Time to discuss/adjust brick laid at join between domains assigned adjacent masons

Speed Up as a Function of Grain Size

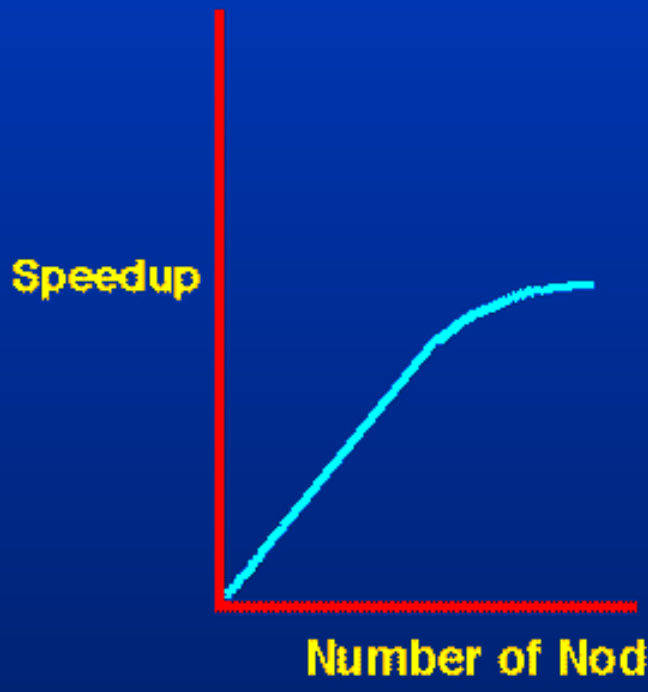
(a) Fixed Grain Size *n*



Amount of problem in each node fixed.

Problem size α N

(b) Fixed Problem Size



n gets small; communication and control dominate.

Grain size *n* α 1/N

Nature's Concurrent Computers

- At the finest resolution, collection of neurons sending and receiving messages by axons and dendrites
- At a coarser resolution
  Society is a collection of brains sending and receiving messages by sight and sound
- Ant Hill is a collection of ants (smaller brains) sending and receiving messages by chemical signals
- **Lesson: All Nature's Computers Use Message Passing**
- **With several different Architectures**

Neural Network

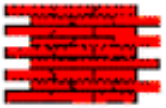
The Web is also just message passing

Parallel Processing for Hadrian's Wall

Comparison of The Complete Problem to the subproblems formed in domain decomposition

For Hadrian's Wall. the complete problem:

is similar to the subtask performed by an individual bricklayer



Changed is

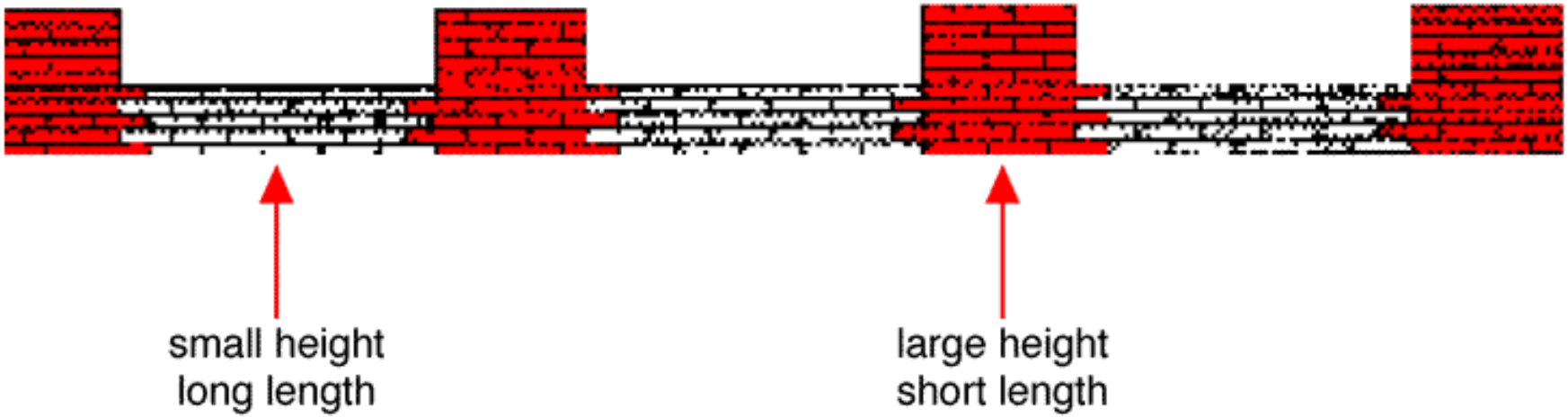
- Geometry
- Boundary Conditions

- *The case of Programming a Hypercube*
  - Each node runs software that is similar to sequential code
  - e.g., FORTRAN with geometry and boundary value sections changed

1984 Slide – today replace hypercube by cluster

Hadrian's Wall Illustrating an Irregular but Homogeneous Problem

- Geometry irregular but each brick takes about the same amount of time to lay.
- Decomposition of wall for an irregular geometry involves equalizing number of bricks per mason, not length of wall per mason.



Equal work not
Equal area of underlying domain
is load balancing requirement

Some Problems are Inhomogeneous Illustrated by:

An Inhomogeneous Hadrian Wall with Decoration

- Fundamental entities (bricks, gargoyles) are of different complexity
- Best decomposition dynamic

- Inhomogeneous problems run on concurrent computers but require dynamic assignment of work to nodes and strategies to optimize this
- (we use neural networks, simulated annealing, spectral bisection etc.)

Global and Local Parallelism Illustrated by Hadrian's Wall

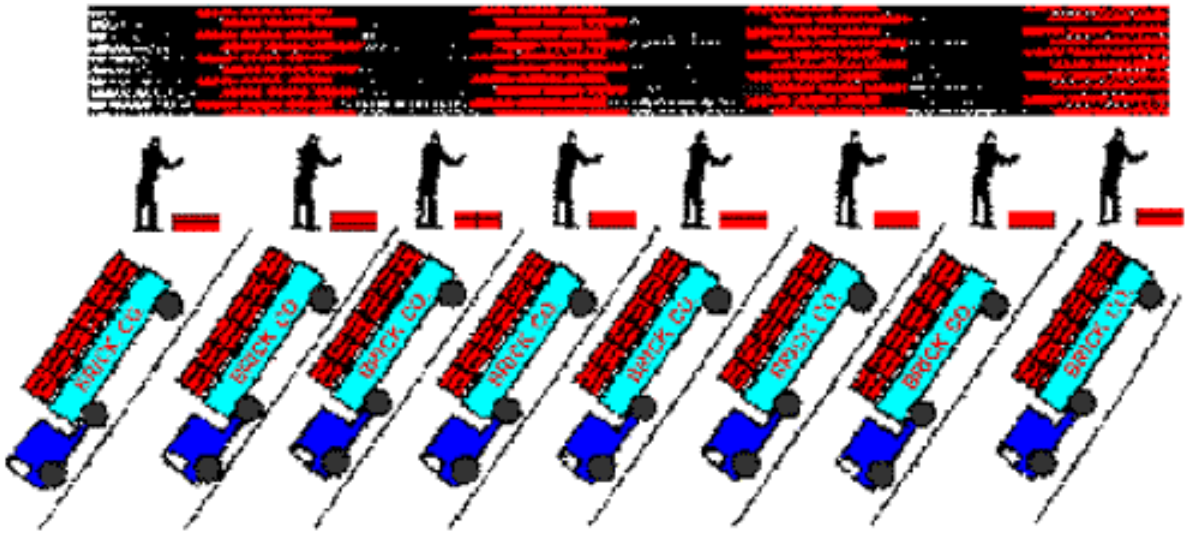
- Global Parallelism
  - Break up domain
  - Amount of Parallelism proportional to size of problem (and is usually large)
  - Unit is Bricklayer or Computer node
- Local Parallelism
  - Do in parallel local operations in the processing of basic entities
    - e.g. for Hadrian's problem, use two hands, one for brick and one for mortar while ...
    - for computer case, do addition at same time as multiplication
  - Local Parallelism is limited but useful
- Local and Global Parallelism Should both be Exploited

Between CPU's
Called Outer Parallelism

Inside CPU or Inner Parallelism

Parallel I/O Illustrated by Concurrent Brick Delivery for Hadrian's Wall

Bandwidth of Trucks and Roads Matches that of Masons



- Disk (input/output) Technology is better matched to several modest power processors than to a single sequential supercomputer

Key to MapReduce

- Concurrent Computers natural in databases, transaction analysis

Comparison of Concurrent Processing in Society and Computing

- Problems are large - use domain decomposition
  Overheads are edge effects
- Topology of processor matches that of domain - processor with rich flexible node/topology matches most domains
- Regular homogeneous problems easiest but
  irregular or Inhomogeneous } work with proper decomposition/planning
- Can use local and global parallelism
- Can handle concurrent calculation and I/O
- Nature always uses message passing as in parallel computers (at lowest level)