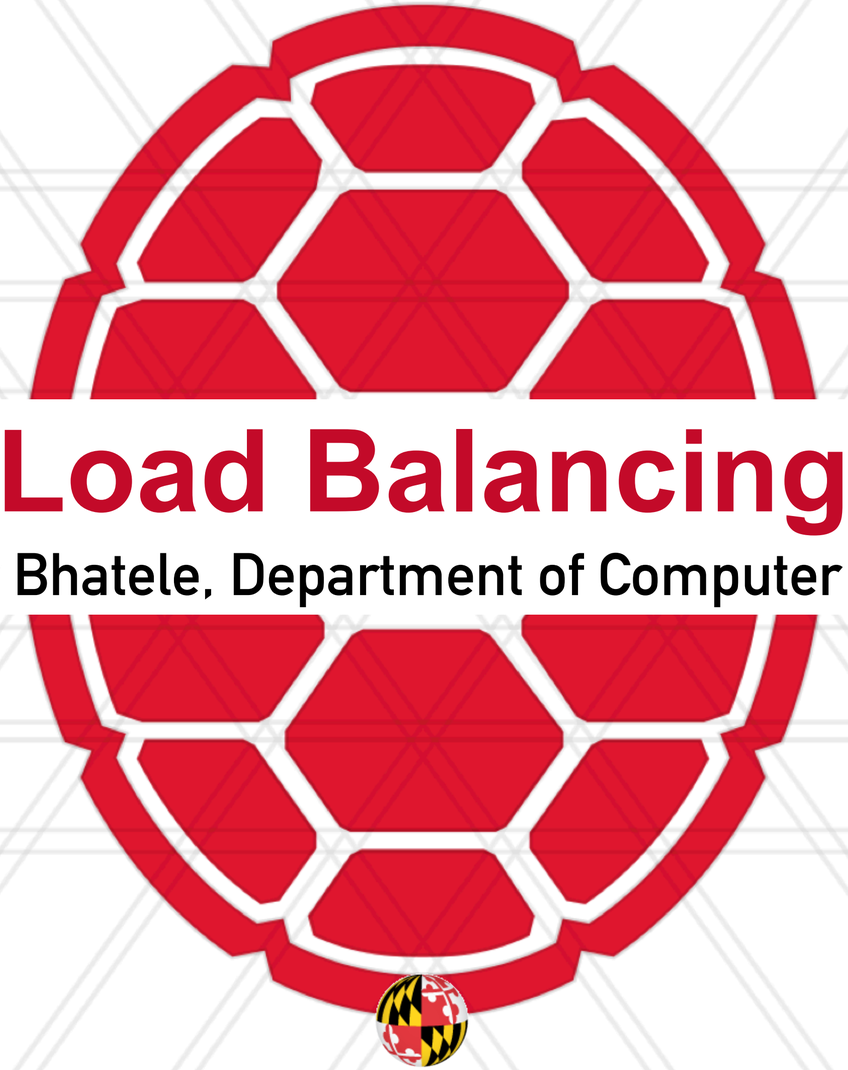
Introduction to Parallel Computing (CMSC416 / CMSC818X)



Load Balancing

Abhinav Bhatele, Department of Computer Science

UNIVERSITY OF MARYLAND

Announcements

- Assignment 5 will be posted today
  - Required only for CMSC818X students
- I will announce the plan for next week's lectures on piazza
- Virtual lecture (on zoom) on Tuesday Nov 22

Performance issues

- Sequential performance issues
- Load imbalance
- Communication performance issues / parallel overhead
- Algorithmic overhead / replicated work
- Speculative loss
- Critical paths
- Insufficient parallelism
- Bottlenecks

Load imbalance

Load imbalance

- Definition: unequal amounts of "work" assigned to different processes
  - Work could be computation or communication or both

Load imbalance

- Definition: unequal amounts of "work" assigned to different processes
  - Work could be computation or communication or both
- Why is load imbalance bad?
  - Overloaded processes can slow down everyone

Load imbalance

- Definition: unequal amounts of "work" assigned to different processes
  - Work could be computation or communication or both
- Why is load imbalance bad?
  - Overloaded processes can slow down everyone

$$\text{Load imbalance} = \frac{max\_load}{mean\_load}$$

Load balancing

- The process of balancing load across threads, processes etc.
- Goal: to bring the maximum load close to average as much as possible
- Determine if load balancing is needed
- Determine when to load balance
- Determine what information to gather/use for load balancing

Is load balancing needed?

- Need the distribution of load ("work") across processes
- Collect empirical information using performance tools
- Developer knowledge
- Analytical models of load distribution

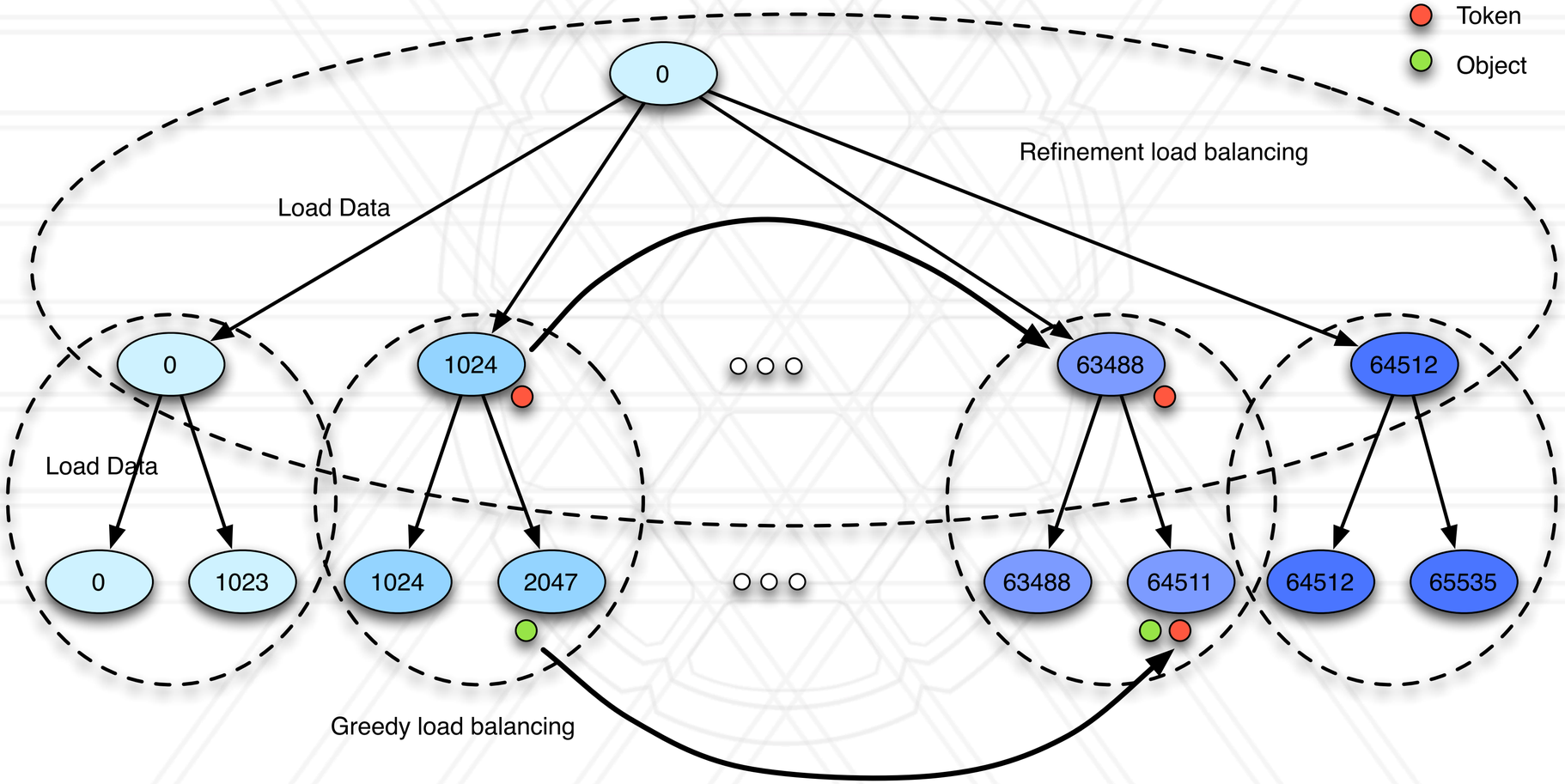
When to load balance?

- Initial work distribution or static load balancing
  - At program startup
  - Or sometimes in a separate run to determine load distribution
- Dynamic load balancing: does load distribution evolve over time?
  - During program execution

Information gathering for load balancing

- Centralized load balancing
  - Gather all load information at one process — global view of data
- Distributed load balancing
  - Every process only knows the load of a constant number of "neighbors"
- Hybrid or hierarchical load balancing

Hierarchical load balancing



What information is used for load balancing

- Computational load
- Possibly, communication load (number/sizes of messages)
- Communication graph

Load balancing algorithms

- Input: Amount of work (n_i) assigned to each process p_i
- Output: New assignments of work units to different processes
- Goals:
  - Bring maximum load close to average
  - Minimize the amount of data migration
- Secondary goals:
  - Balance (possibly reduce) communication load
  - Keep the time for doing load balancing to a minimum

Examples of static load balancing

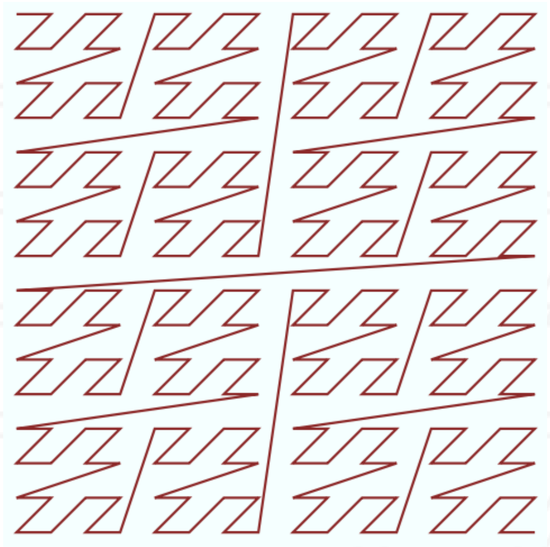
- Decomposition of 2D Stencil
- Using orthogonal recursive bisection (ORB)

http://datagenetics.com/blog/march22013/

https://en.wikipedia.org/wiki/Z-order_curve

Examples of static load balancing

- Decomposition of 2D Stencil
- Using orthogonal recursive bisection (ORB)



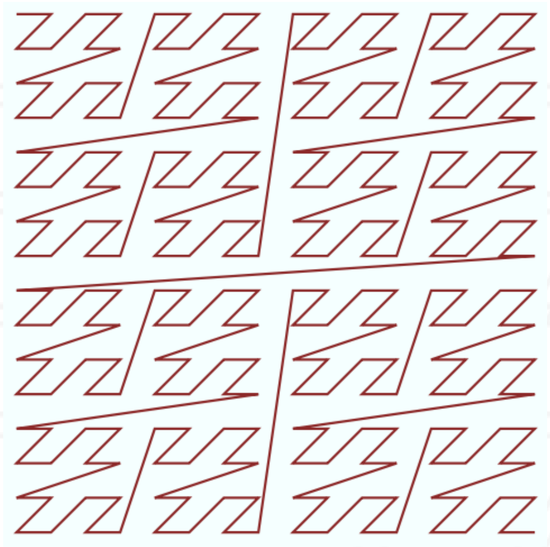
http://datagenetics.com/blog/march22013/

https://en.wikipedia.org/wiki/Z-order_curve

Examples of static load balancing

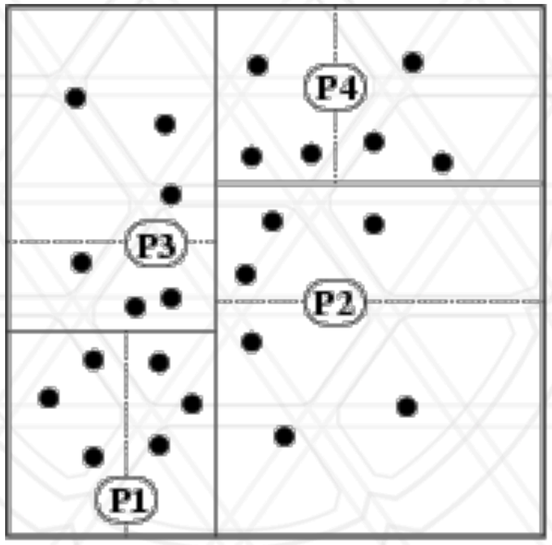
- Decomposition of 2D Stencil
- Using orthogonal recursive bisection (ORB)

Space-filling curves

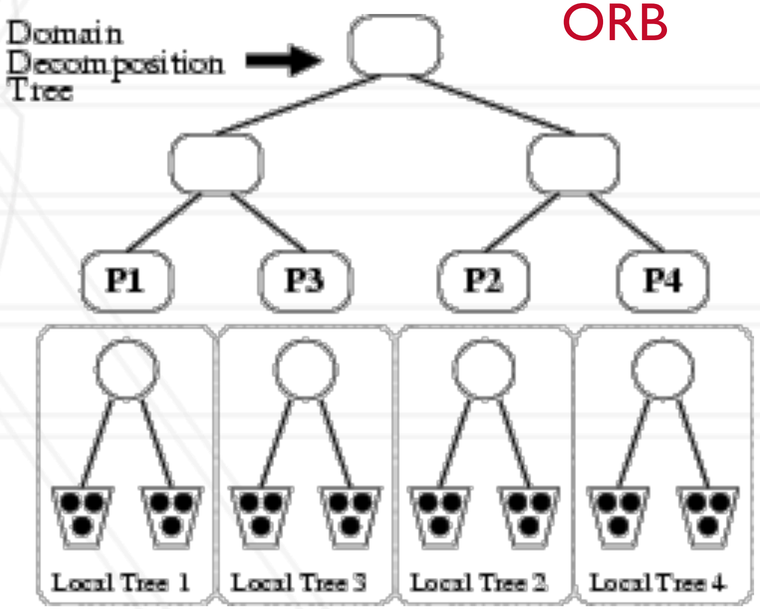


http://datagenetics.com/blog/march22013/

https://en.wikipedia.org/wiki/Z-order_curve



ORB



http://charm.cs.uiuc.edu/workshops/charmWorkshop2011/slides/CharmWorkshop2011_apps_ChaNGa.pdf

Simple greedy strategy

- Sort all the processes by their load
- Take some load from the heaviest process and assign it to the most lightly loaded process

Work stealing

- Decentralized strategy where processes steal work from nearby processes when they have nothing to do
- Each process has a queue of work items
  - Looks at the other processes' queues when there are no items remaining
- Implemented in Cilk among other languages

Other considerations

- Communication-aware load balancing
- Network topology-aware load balancing

UNIVERSITY OF
MARYLAND

Abhinav Bhatele

5218 Brendan Iribe Center (IRB) / College Park, MD 20742

phone: 301.405.4507 / e-mail: bhatele@cs.umd.edu