Distributed Load Balancing Strategies with Charm++

Presented by

Simeng Liu, Kavitha Chandrasekar

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Outlines

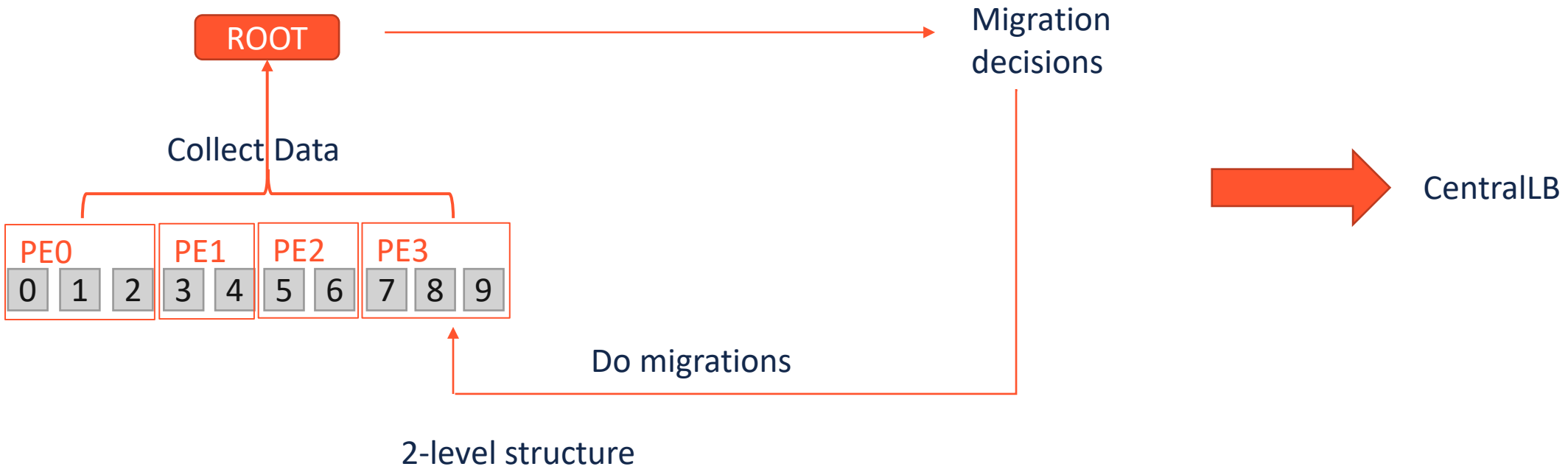
- Load Balancers Analysis
  - Prefix
  - Orthogonal recursive bisection (ORB)
  - Diffusion
- Application Characteristics
  - Iterative
  - Spatial locality with coordination
  - Well-scaled

Charm++ Load Balancing Infrastructure

Tool: migrate chare objects

Structures:

- TreeLB (2-4 levels)



Charm++ Load Balancing Infrastructure

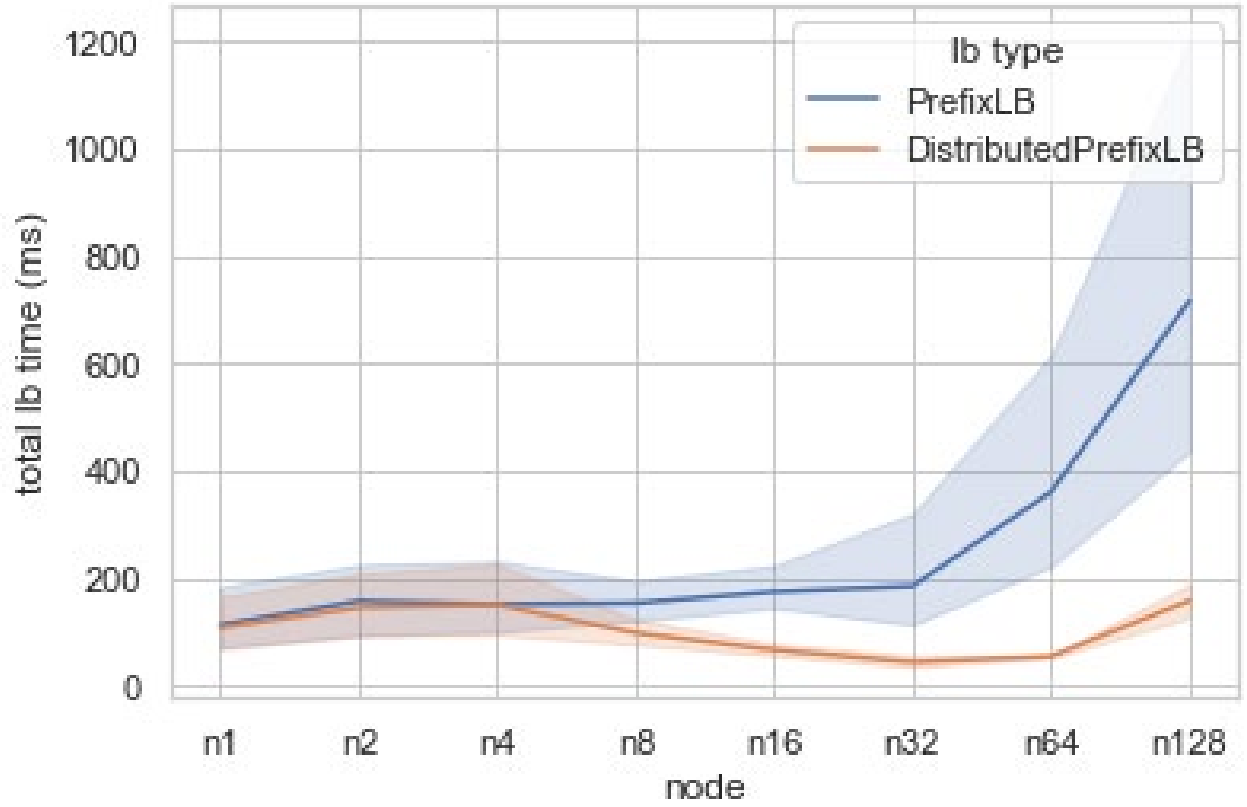
Tool: migrate chare objects

Structures:

- TreeLB (2-4 levels)
- DistributedLB
  - each PE makes individual decisions

Prefix Based LB with ParaTreeT

Comparison of CentralLB and DistributedLB implementations:



Stampede2
SKX

48 cores / node
6144 cores on 128 nodes

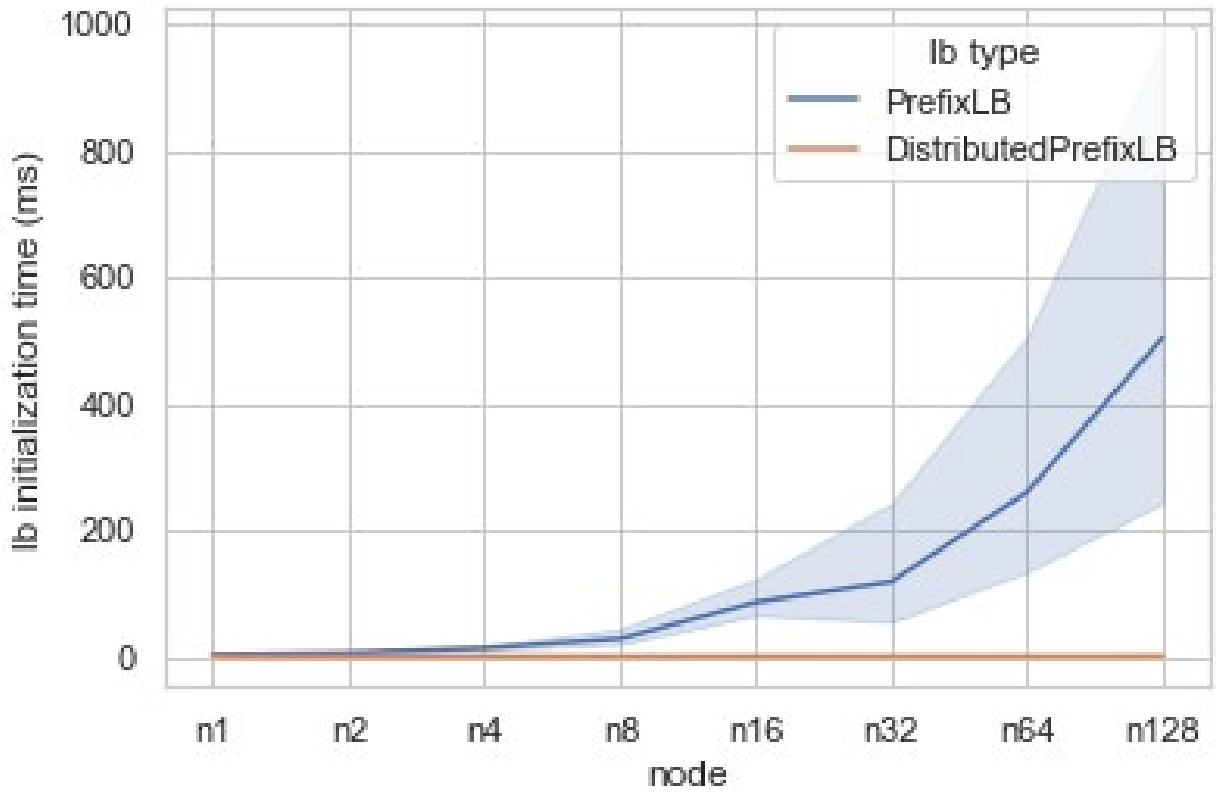
Prefix Based LB with ParaTreeT

Composition of LB time:

Initialization time	Strategy time	Migration
Collect object data - Measured runtime - Communication graph	Apply LB strategy and make migration decisions	Migrate objects

Prefix Based LB with ParaTreeT

Initialization time analysis:



Let

P := the number of PEs

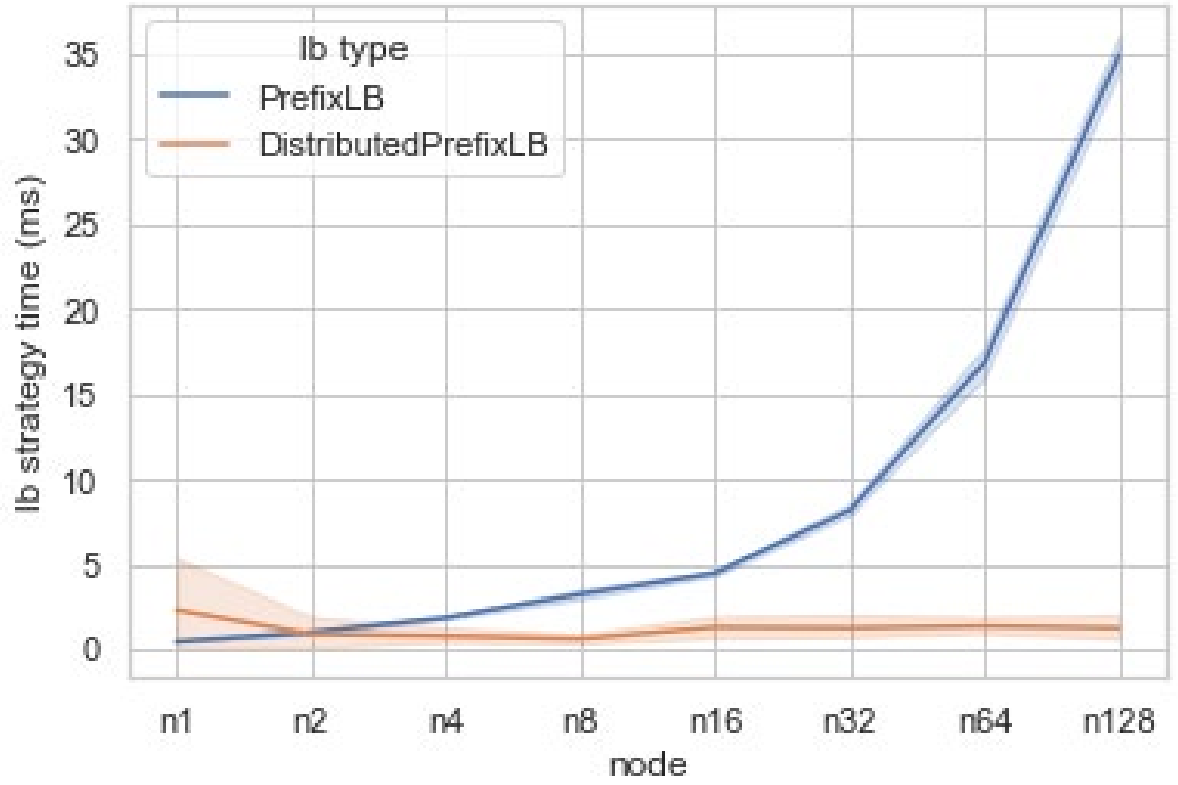
O := the number of objects

Reduction

LB	Runtime	Bandwidth
PrefixLB	log P	O
DistributedPrefixLB	1	N/A

Prefix Based LB with ParaTreeT

Strategy time analysis:



Let

P := the number of PEs

O := the number of objects

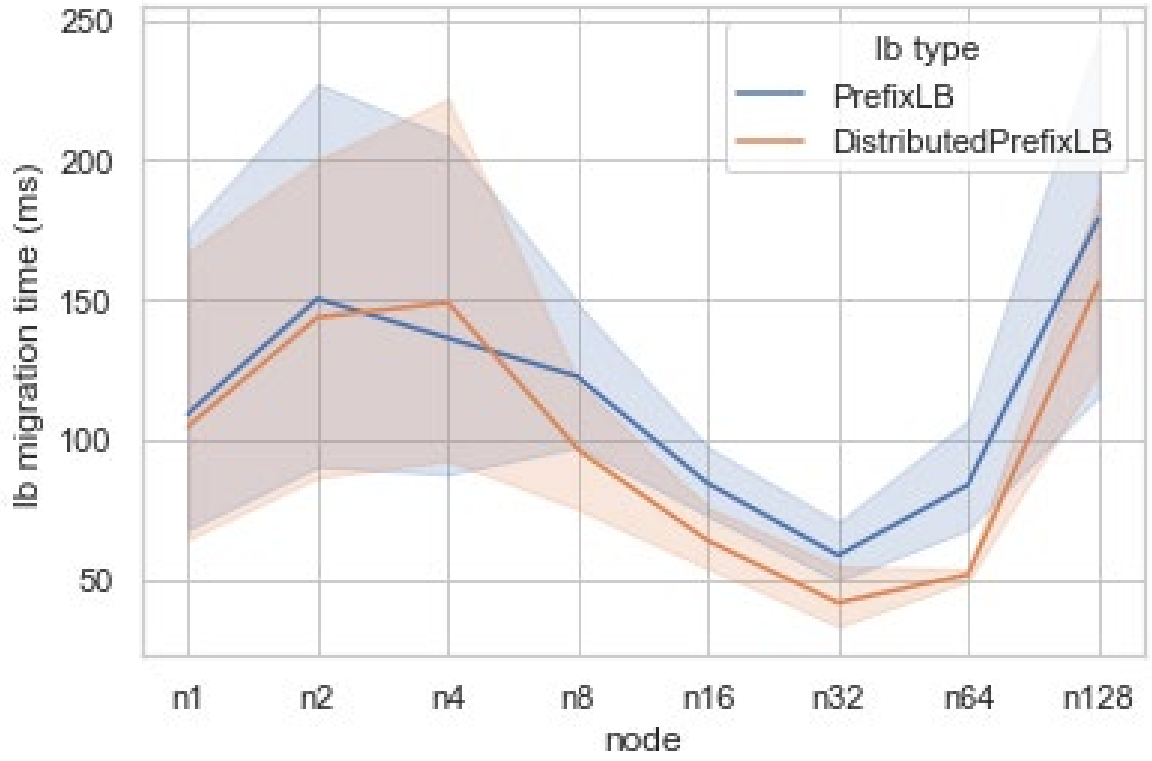
Sorting

LB	Runtime	Bandwidth
PrefixLB	OlogO	N/A
Distributed PrefixLB	logP	log P

Recursive doubling

Prefix Based LB with ParaTreeT

Migration time analysis:



Let

P := the number of PEs
O := the number of objects
M:= number of objects need migration

LB	Runtime	Bandwidth
PrefixLB	M/P	M
Distributed PrefixLB	M/P	M

ORB Based LB with ParaTreeT

Algorithm:

- Goal:

  - Partition the universe into number of PE blocks with even loads

- Centralized:

  - Use selection algorithm to find a splitting coordinate along the longest dimension of the subspace

- Distibuted:

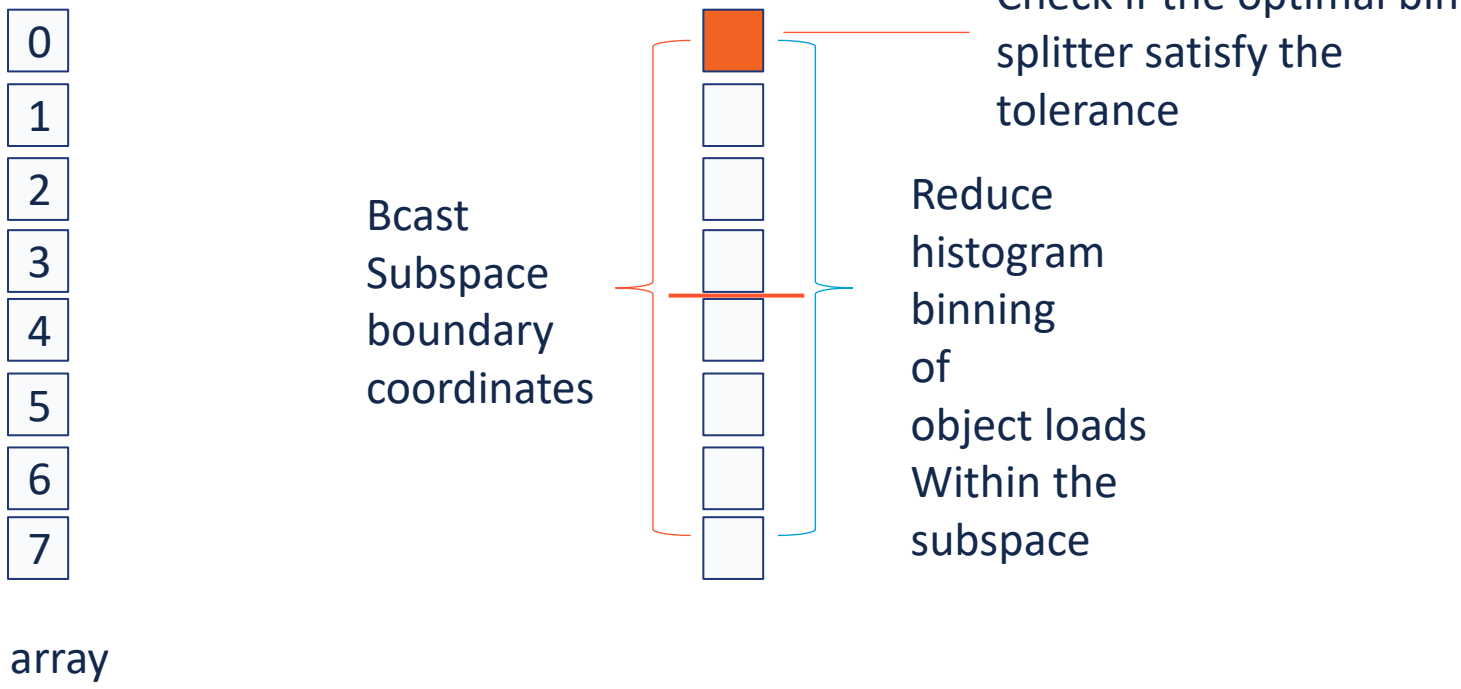
  - One partition: find a splitting coordinate

ORB Based LB with ParaTreeT

Algorithm:

- Distibuted:

  - One partition: find a splitting coordinate



ORB Based LB with ParaTreeT

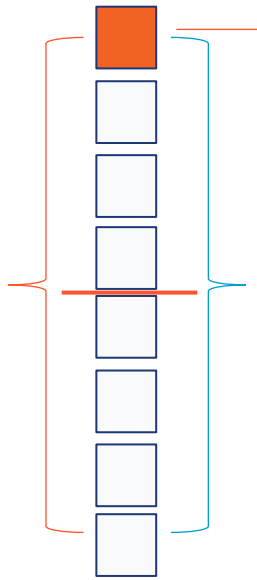
Algorithm:

- Distibuted:

  - One partition: find a splitting coordinate

PE array

Bcast Subspace boundary coordinates



Reduce histogram binning of object loads Within the subspace

Check if the optimal bin splitter satisfy the tolerance

leader PE

If yes, done

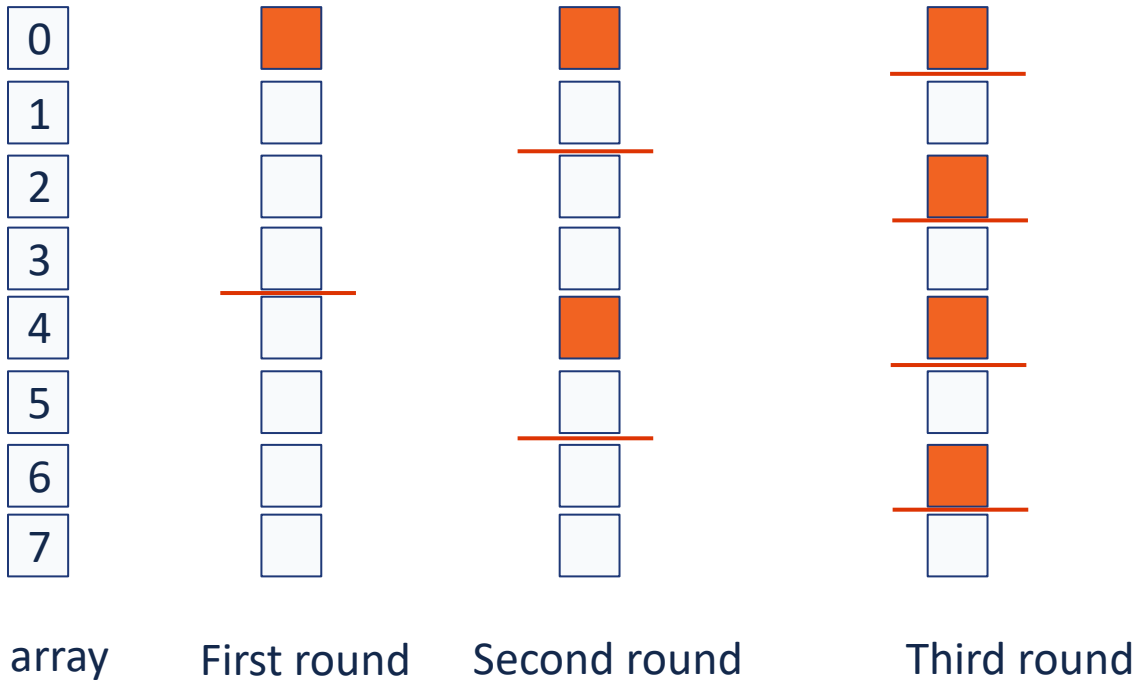
If the optimal bin have less that THRESHOLD objects:
Collect coordinations of those objects directly

ORB Based LB with ParaTreeT

Algorithm:

- Distibuted:

  - make number of PE partitions

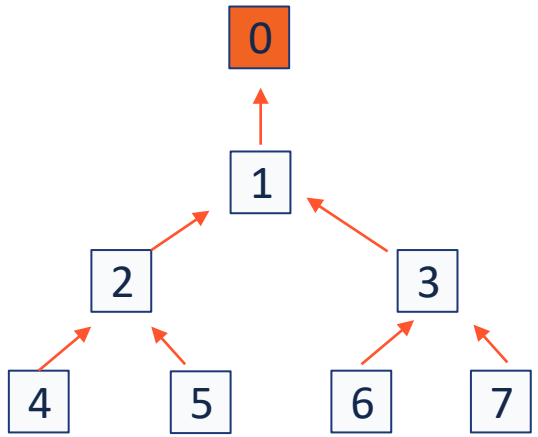


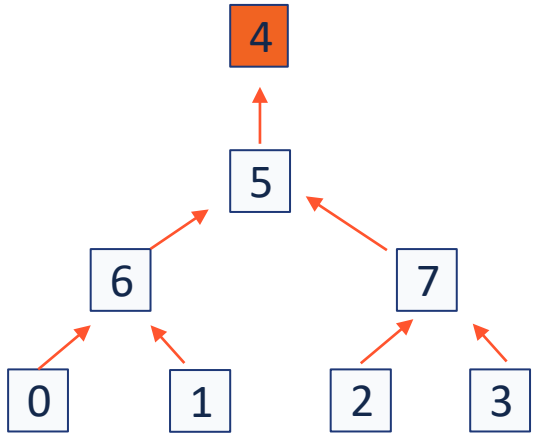
ORB Based LB with ParaTreeT

Algorithm:

- Distibuted:

  - make multiple reductions with different roots

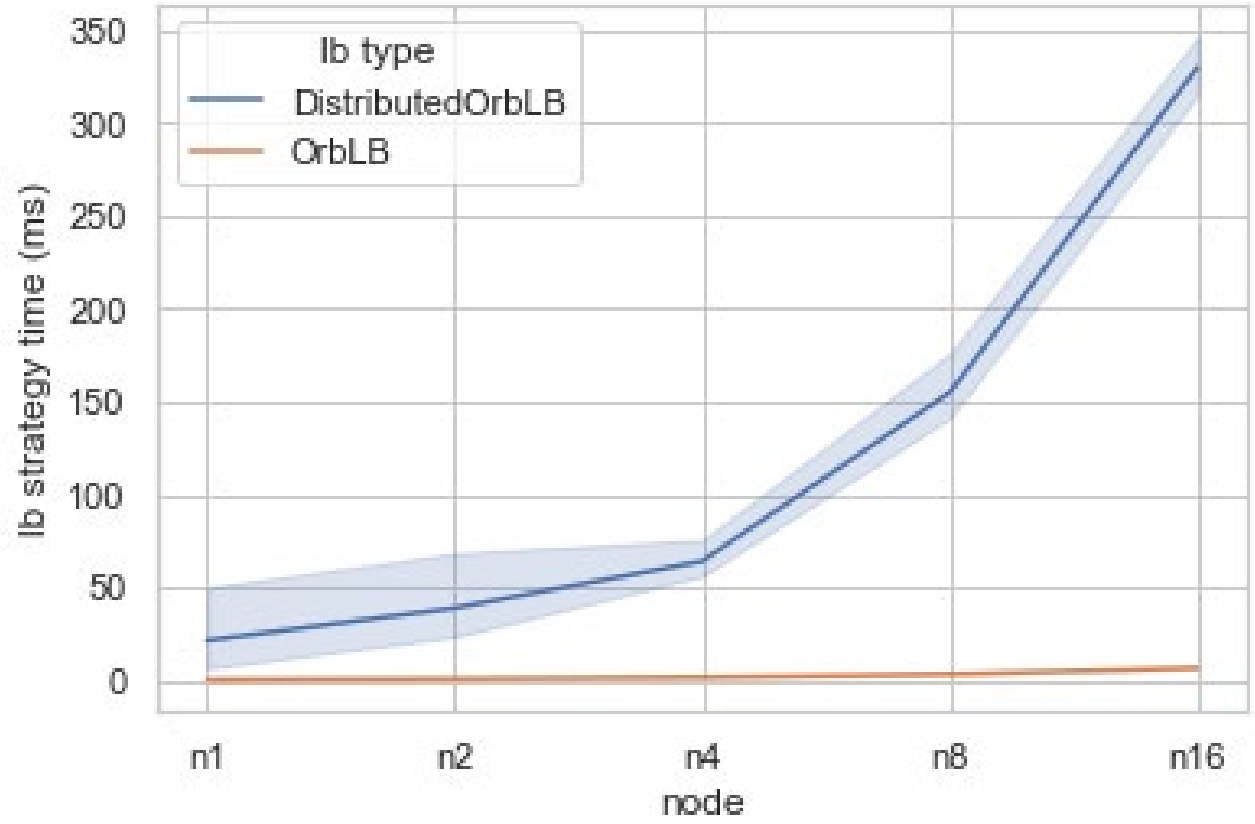




Second round

ORB Based LB with ParaTreeT

Strategy time analysis: 9478 objects 768 cores 12



Let

P := the number of PEs

V := the number of objects

LB	Runtime	Bandwidth
OrbLB	VlogP	N/A
Distributed OrbLB	V(logP)^2	VlogP

Summary

- Analysis of the LB runtime with a three-stage decomposition

  - Try the DistributedLB if strategy runtime <= log(P)

- Will Improve DistrbutedOrbLB
  - ORB to the node level
  - Diffusion within each node