Placement Group Scheduling on Hierarchical Topology

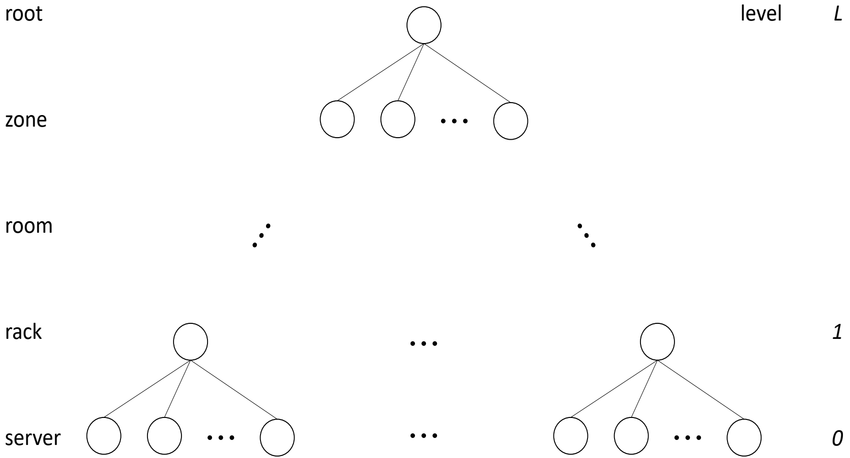
Asser N. Tantawi

IBM T.J. Watson Research Center
Yorktown Heights, NY, USA

August 27, 2021

Outline

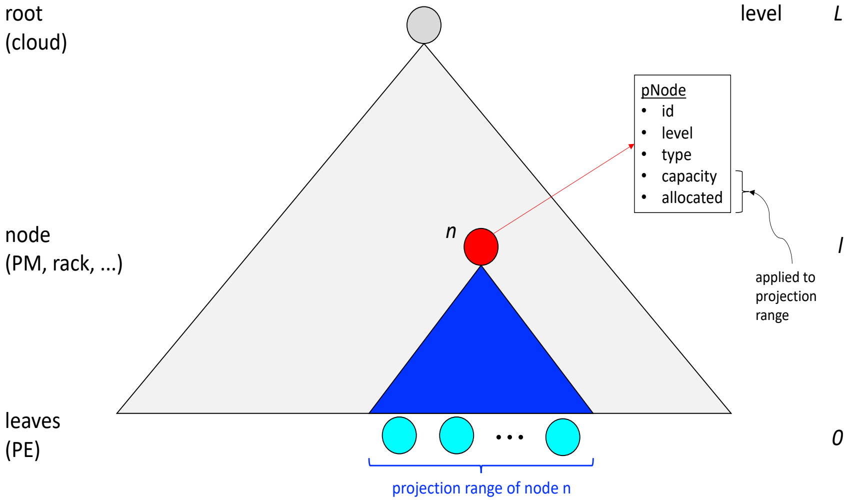
Model cloud as a hierarchy



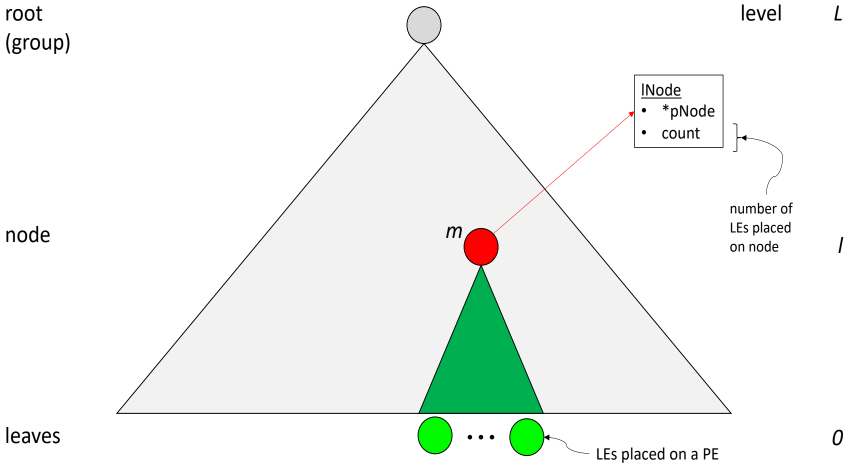
Definitions

- $PhysicalEntity(PE)$
  - target for placement, host, PM, server, device, VM
- $LogicalEntity(LE)$
  - object to be placed, VM, container, Pod, task, volume
- $LevelConstraint(LC)$
  - constraint specifying desired placement at a given level
- $PlacementGroup(PG)$
  - a collection of homogeneous LEs and a LC(s), applied to the collection
- $Placement(P)$
  - assignment of LEs in a PG to PEs, satisfying the LC(s) in the PG
- $PhysicalTree(pTree)$
  - tree topology of physical infrastructure (leaves are PEs)
- $LogicalTree(lTree)$
  - tree topology of a particular placement P of a PG (subset of $pTree$, nodes with nonzero assignments)

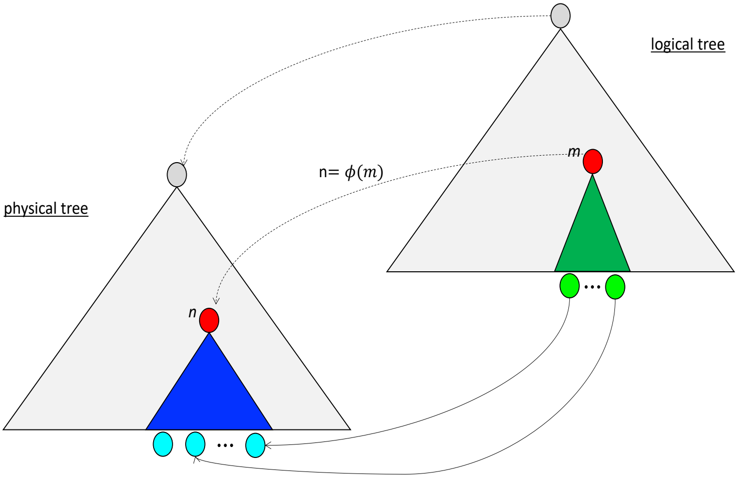
Physical tree



Logical tree



Placement: mapping from logical to physical



Level constraint

Example

```yaml
kind: GroupPlacement
spec:
  group:
    name: MyApp
    size: 20
    type: bx2-16x64
  constraints:
    - level: rack
      affinity: pack
      soft: true
```

group size

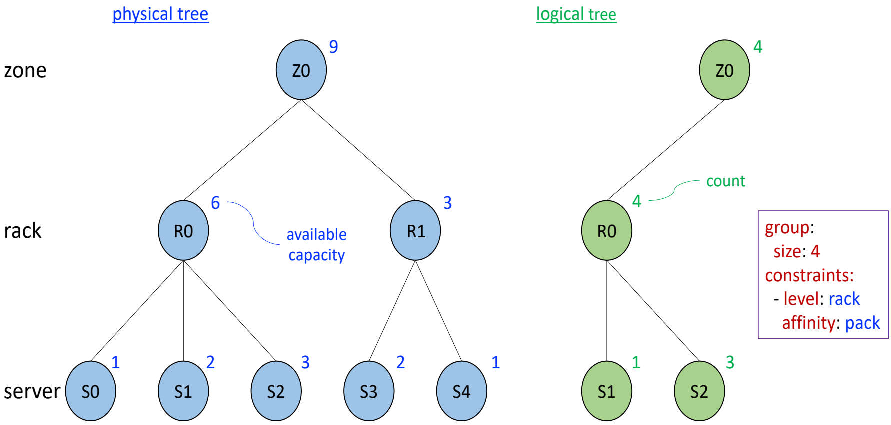
- specified, fixed
- unspecified, estimated, dynamic

type

- homogeneous
- specifies resources requested

level: zone | room | rack | server
affinity: spread | pack
soft: true | false

Example placement



Example placement

```
pTree:
root -> ( rack-0 -> ( server0 server1 server2 ) rack-1 -> ( server3 server4 ) )
pNodes:
pNode: ID=root; level=2; cap=[80 1280]; alloc=[44 352]
pNode: ID=rack-1; level=1; cap=[32 512]; alloc=[20 160]
pNode: ID=rack-0; level=1; cap=[48 768]; alloc=[24 192]
pNode: ID=server3; level=0; cap=[16 256]; alloc=[8 64]
pNode: ID=server4; level=0; cap=[16 256]; alloc=[12 96]
pNode: ID=server0; level=0; cap=[16 256]; alloc=[12 96]
pNode: ID=server1; level=0; cap=[16 256]; alloc=[8 64]
pNode: ID=server2; level=0; cap=[16 256]; alloc=[4 32]

PG: ID=pg0; size=4; demand=[4 32]; lc=lc0
LC: ID=lc0; level=1; affinity=Pack; isHard=false

lTree:
root -> ( rack-0 -> ( server1 server2 ) )
lNodes:
lNode: ID=root; count=4
lNode: ID=rack-0; count=4
lNode: ID=server2; count=3
lNode: ID=server1; count=1
```

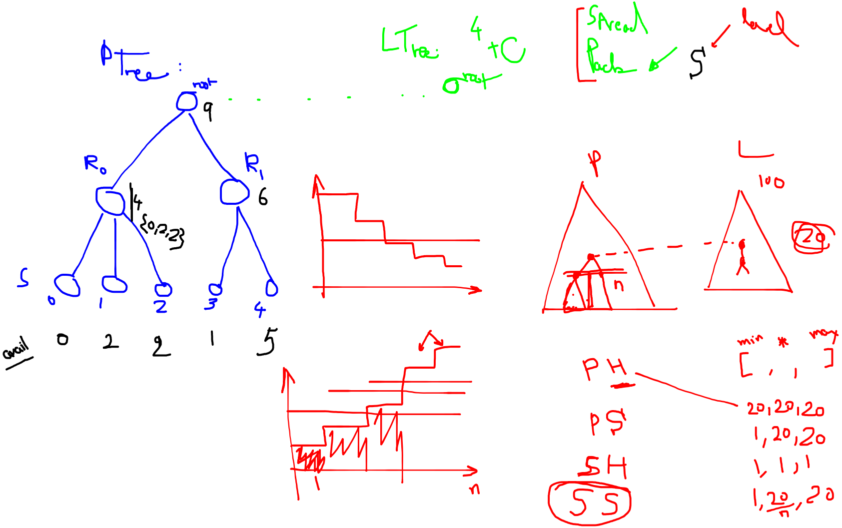
Algorithm: High level

Algorithm 1 Logical assignment of placement group

```
procedure PLACEGROUP(pTree, pg) return lTree
    numRemaining ← pg.size
    root ← PLACEATNODE(pTree.root)                          ▷ recursive DFS
    return newLTree(root)

procedure PLACEATNODE(pNode) return lNode
    lNode ← newLNode(pNode, 0)
    numFit ← ⌊pNode.available/pg.demand⌋
    numPlaced ← NUMBERTOPLACE(pg, pNode.level, numFit)
    if numPlaced = 0 then return lNode                       ▷ skip pNode
    if pNode.level = 0 then                                  ▷ pNode is a leaf node
        numRemaining ← numRemaining − numPlaced
    else
        C ← ORDERNODES(pNode.children, pg.constraint); count ← 0
        for all c ∈ C ∧ numRemaining > 0 do
            node ← PLACEATNODE(c)
            if node.count > 0 then
                lNode.addChild(node); count ← count + node.count
        numPlaced ← count
    lNode.count ← numPlaced
    return lNode
```

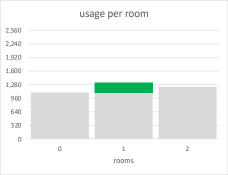
Algorithm: High level



Algorithm: Questions/Choices

- *orderNodes*()
  - heuristics used
  - should depend on level constraint (level, affinity)
- *numberToPlace*()
  - best choice of number to place, given the number that fits and the range
  - representation of number to place {minimum, desired, maximum}
  - dependence on spread/pack affinity
  - dependence on soft/hard constraints
- evaluation of placement
- resizing group
  - increase: best incremental placement, given current
  - decrease: best to delete, given current

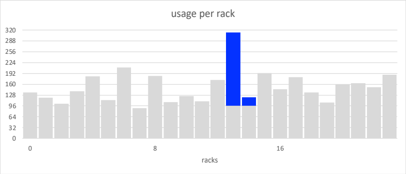
Experiment: Pack

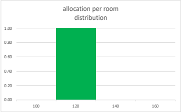


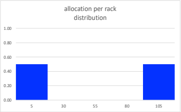
Pack
Rack
Soft

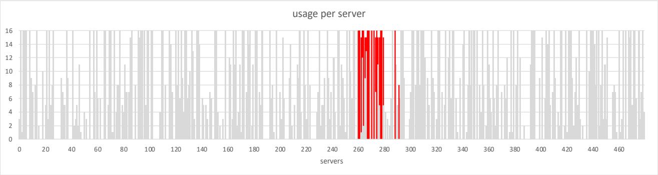
Infrastructure:
- 1 zone
- 3 rooms/zone
- 8 racks/room
- 20 servers/rack

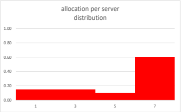
Placement Group:
- 120 VMs
- 1/8 server

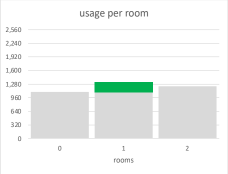








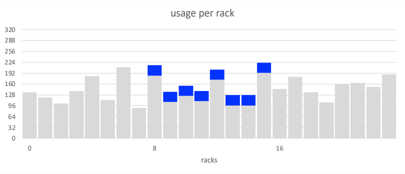


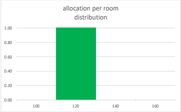


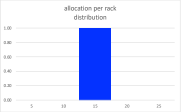
Spread
Rack
Soft

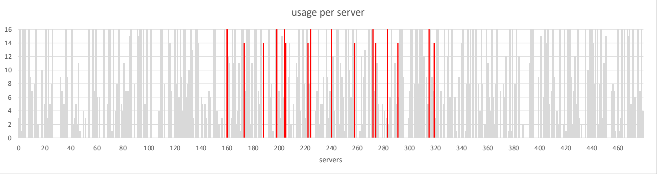
Infrastructure:
- 1 zone
- 3 rooms/zone
- 8 racks/room
- 20 servers/rack

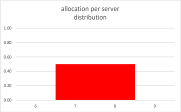
Placement Group:
- 120 VMs
- 1/8 server



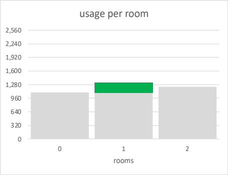








Experiment: Spread



Spread
Rack
Soft

Spread
Server
Soft

<u>Infrastructure</u>:
1 zone
3 rooms/zone
8 racks/room
20 servers/rack
<u>Placement Group</u>:
120 VMs
1/8 server