# Chic-sched: a HPC Placement-Group Scheduler on Hierarchical Topologies with Constraints

Presented by Saverio Scumaci

# Problem

- New resource scheduler challenges on shared infrastructure like the Cloud
  - HPC and AI work-loads with application constraints need to remain efficient

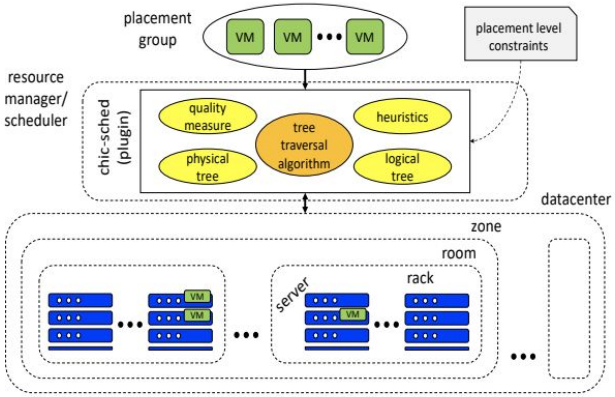



Fig. 1: Schematic view of placement group scheduling on a hierarchical topology

# Chic-Sched

- Novel placement group scheduler
- Designed for distributed workloads on hierarchical topologies with constraints
- Efficiently places groups of virtual machines (VMs) while adhering to various constraints, such as packing and spreading requirements
- Operates without retries, enabling fast scheduling even for large VM groups

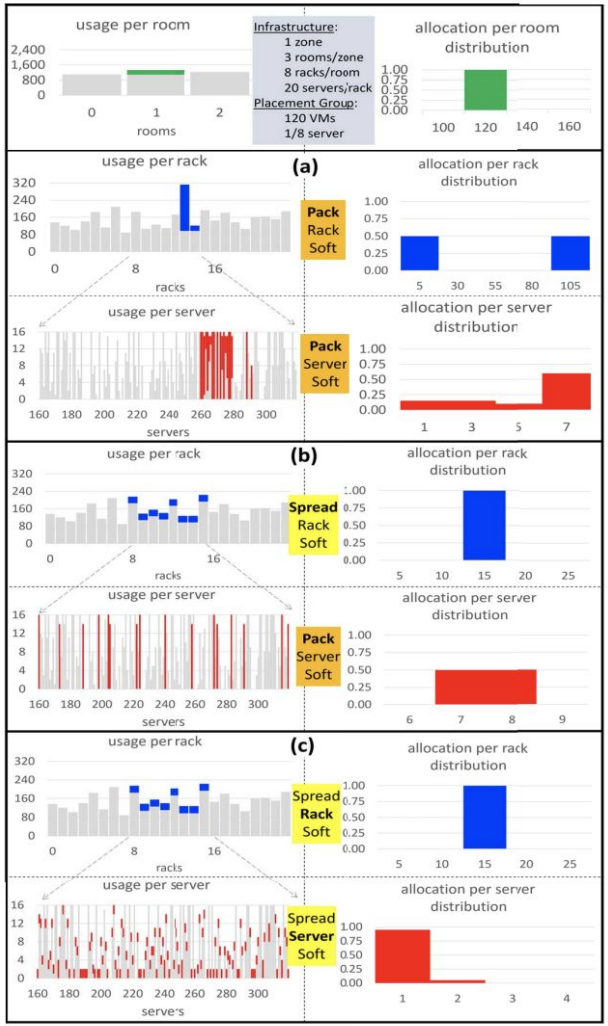



Fig. 3: Example: placement of groups with constraints.

# Deviation Measure (δ)

- Provides a quantitative assessment of how well the placement satisfies the specified constraints
  - range of [0, 1]
  - 0 signifies perfect placement
  - 1 indicates poor placement
- The specific formula for calculating δ depends on the type of level constraint being considered
  - The level constraint is "Pack," achieving a perfect Pack results in a deviation of 0.
  - The level constraint is "Spread," achieving a perfect Spread results in a deviation of 1

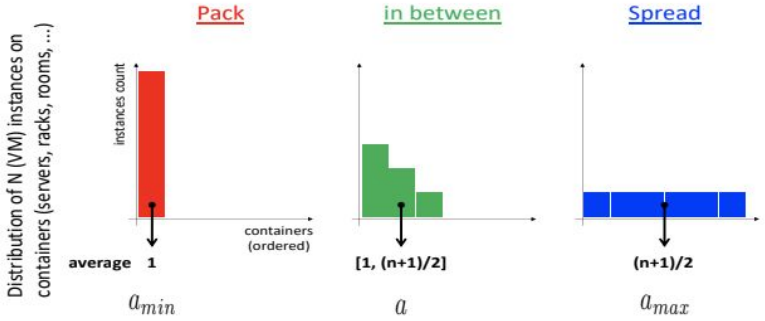
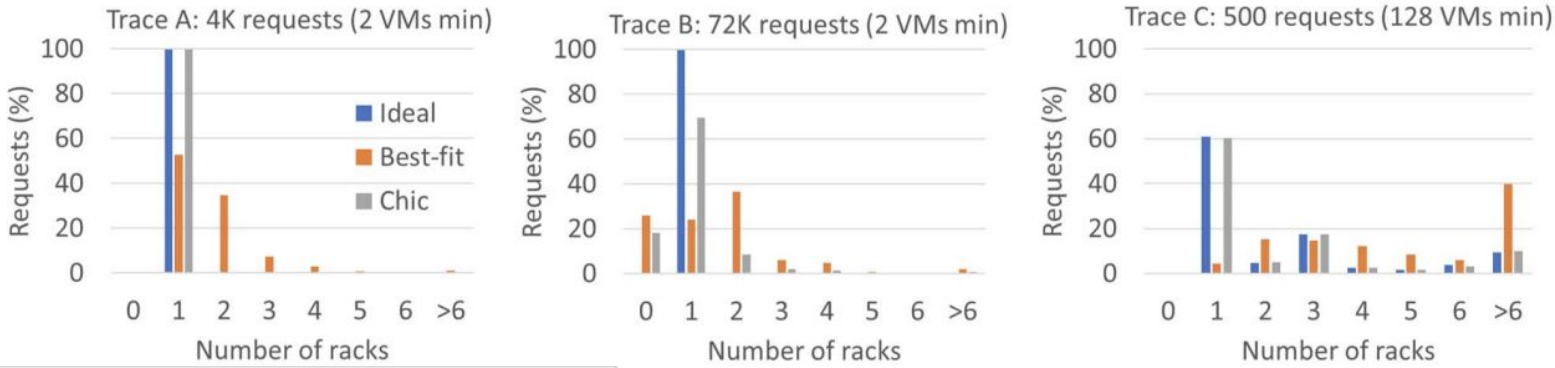


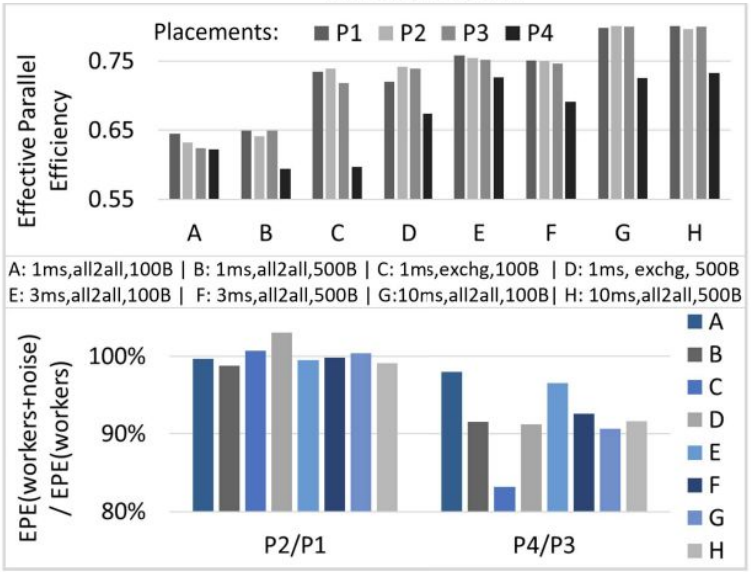

Fig. 4: Measure of deviation from Pack/Spread constraint.

$$\tilde{a} = \frac{a - a_{min}}{a_{max} - a_{min}}, \qquad \delta = \begin{cases} \tilde{a}, & Pack, \\ 1 - \tilde{a}, & Spread, \end{cases}$$

# Results

- Chic-sched consistently outperforms other common placement algorithms, such as bestFit and worstFit, in terms of placement quality
- Demonstrates better placement locality and fewer placement failures, particularly when evaluated with real-world cloud traces and workloads
- exhibits linear scalability in relation to data center size and group size variations.
    - Even in large-scale data centers, chic-sched remains efficient, ensuring quick and effective placement of VMs.



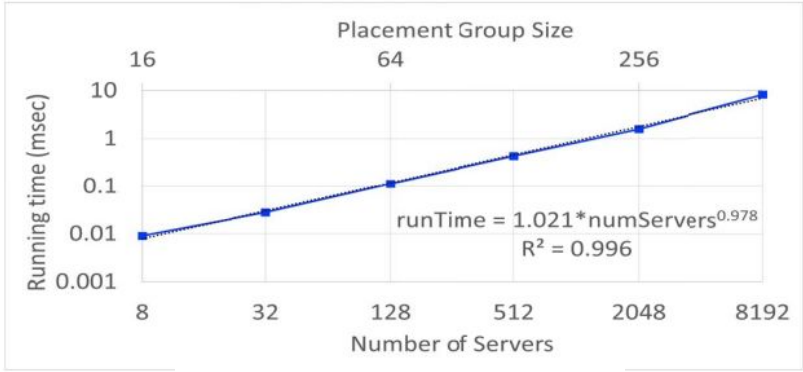




TABLE III: Chic-sched vs state-of-the-art schedulers

| Scheduler | Pack | Spread | Unidimensional constraints | Multi-level constraints |
|---|---|---|---|---|
| Mesos [19] | ✓ | - | - | - |
| YARN [20] | ✓ | - | - | - |
| Borg [21] | ✓ | ✓ | - | - |
| Medea [18] | ✓ | ✓ | ✓ | - |
| Kubernetes [22] | ✓ | ✓ | - | - |
| Quiet neighbor-hoods [23] | ✓ | - | ✓ | - |
| **Chic-sched** | ✓ | ✓ | ✓ | ✓ |

# Contributions

- Development and explanation of Chic-Shed Algorithm
- Performance evaluation of Chic-Shed
- new metric for measuring the quality of placements
    - Deviation Measure(δ)