Distributed Load Balanced Scheduling in Datacenters

Systems Research Group (SRG)
Computer Laboratory
University of Cambridge

Smita Vijayakumar
sv440@cam.ac.uk
First Year PhD Student

Evangelia Kalyvianaki
ek264@cam.ac.uk
PhD Supervisor

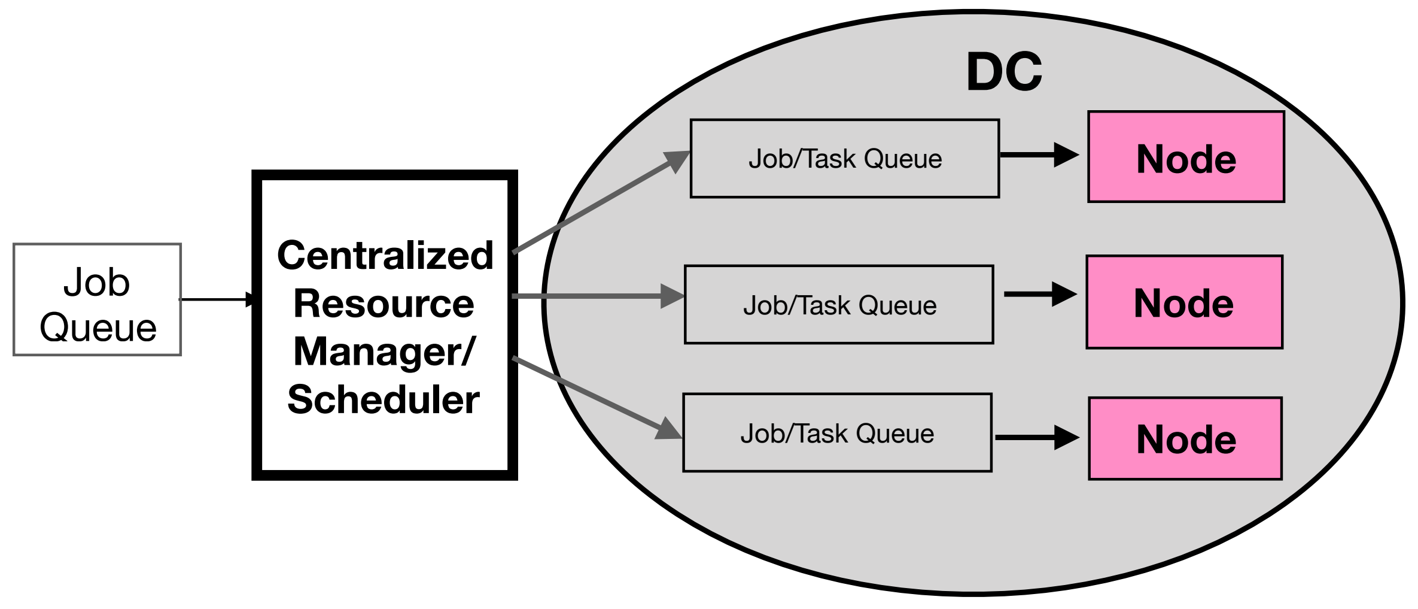
Anil Madhavapeddy
avsm2@cam.ac.uk
PhD Supervisor

Datacenter resources are under-utilised

- ☑ 60% VMs hosted on Azure have less than 20% average CPU usage! *[Resource Central, SOSP'17]*
- ☑ Average production server CPU and memory usage at Alibaba is 50% and 60% respectively [https://github.com/alibaba/clusterdata]
- ☑ A 100-megawatt data center that wastes even 1% of its computing cycles can nullify all the energy-saving measures of a small city *[Scalable system scheduling for HPC and big data, JPDC'17]*

Schedulers In Datacenter

Centralized

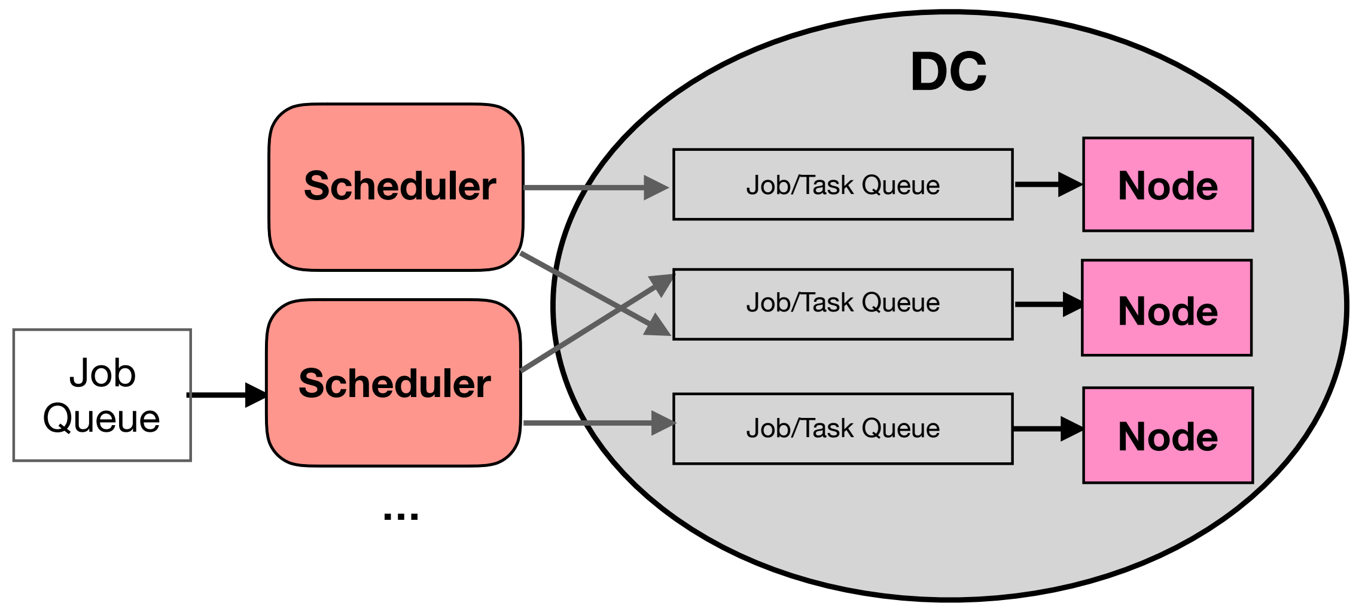


Machines send updates on their states ensuring scheduler has a global resource view

Examples - Mesos, Yarn, Apollo

- ✖ **Suffers from scheduler bottleneck**
- ✖ **Overhead of node information traffic**

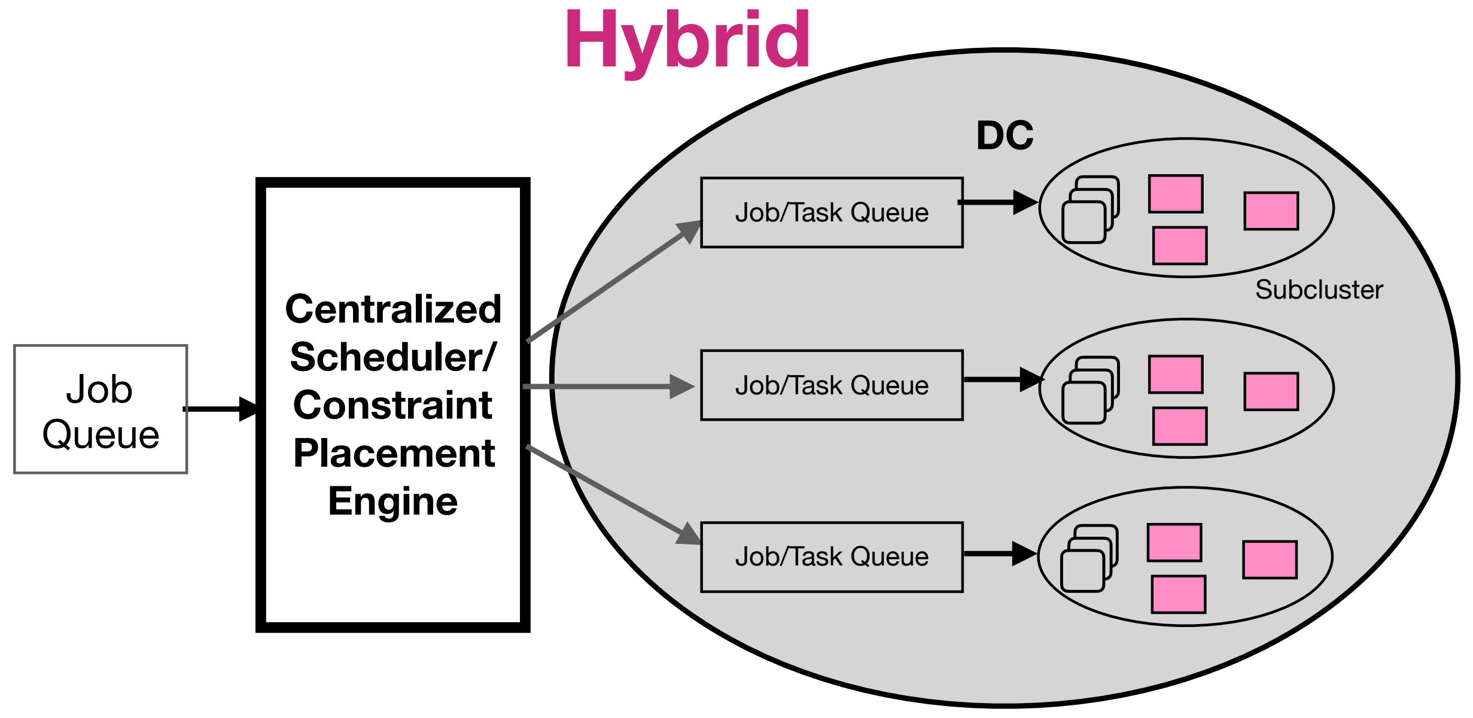
De-Centralized (Sparrow)



Scheduler samples a few nodes for placement

- ☑ **Fast and simple for jobs with short tasks**
- ☑ **Cluster might not be optimally used always**

Hybrid



Hierarchal Scheduling

Example - Hydra *[NSDI'19]*, **Medea** *[EuroSys'18]*, **Borg** *[EuroSys'15]*

- ☑ **Multi-level scheduling ensures better job/task placement**
- ☑ **Lesser node information traffic compared to Centralized**

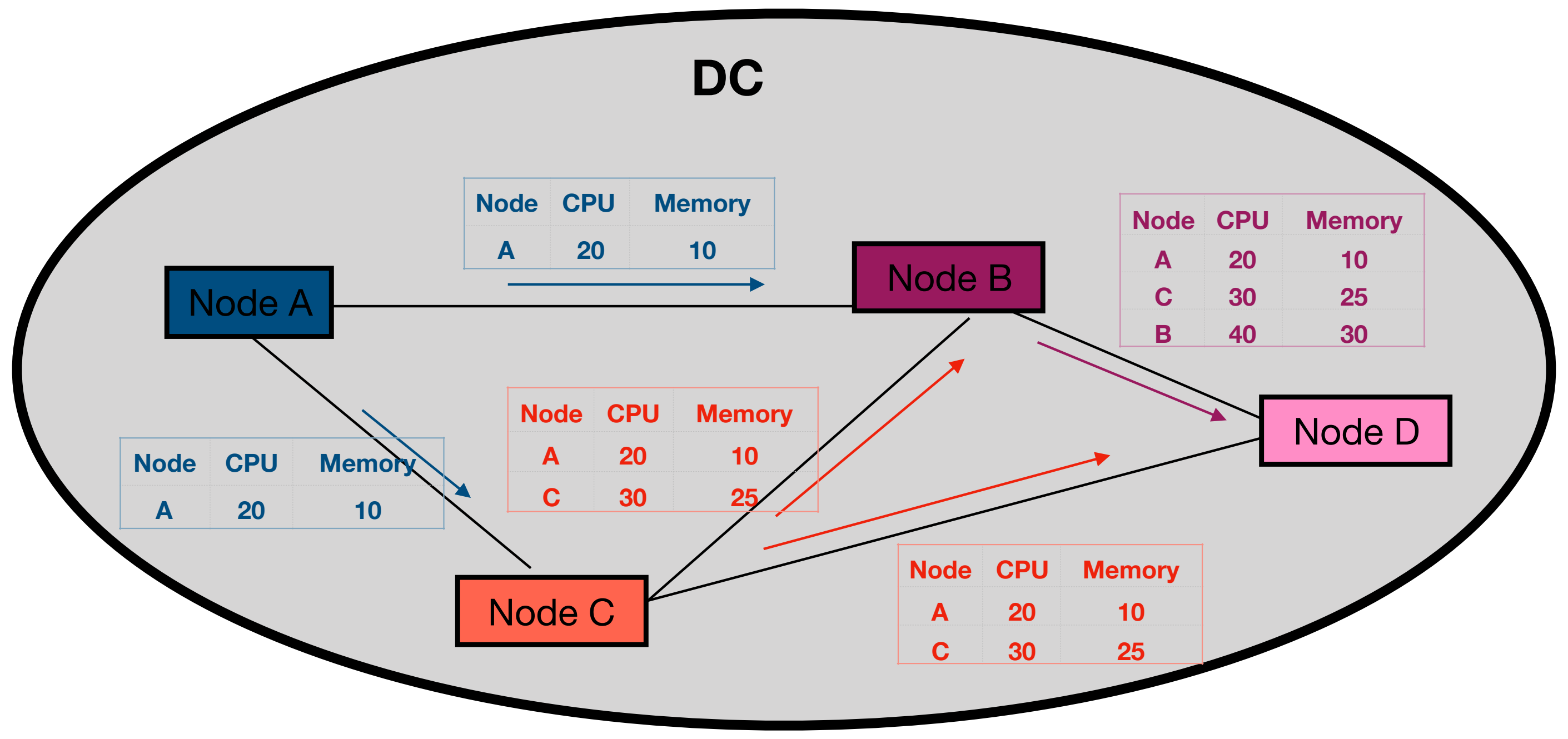
Is a De-Centralized Global Scheduling Possible?

- ☑ **Ensures no single scheduling bottleneck**
- ☑ **Information is available locally at every node and reasonably up-to-date**
- ☑ **Updates to global view converges in a timely**
- ☑ **Every node is the worker and the scheduler!**
- ☑ **Every node is intelligent - it knows how to rank the information according to the policy applied**

Timely Current Global View At Each Node

We propose a solution inspired by various routing protocols

- ☑ ***Nodes send resource information around***
- ☑ ***All nodes converge in a timely fashion to same resource information in their tables***
- ☑ ***Identical policy-based ranking algorithm runs on all nodes***



Node	CPU	Memory
A	20	10

Node	CPU	Memory
A	20	10
C	30	25
B	40	30

Node	CPU	Memory
A	20	10

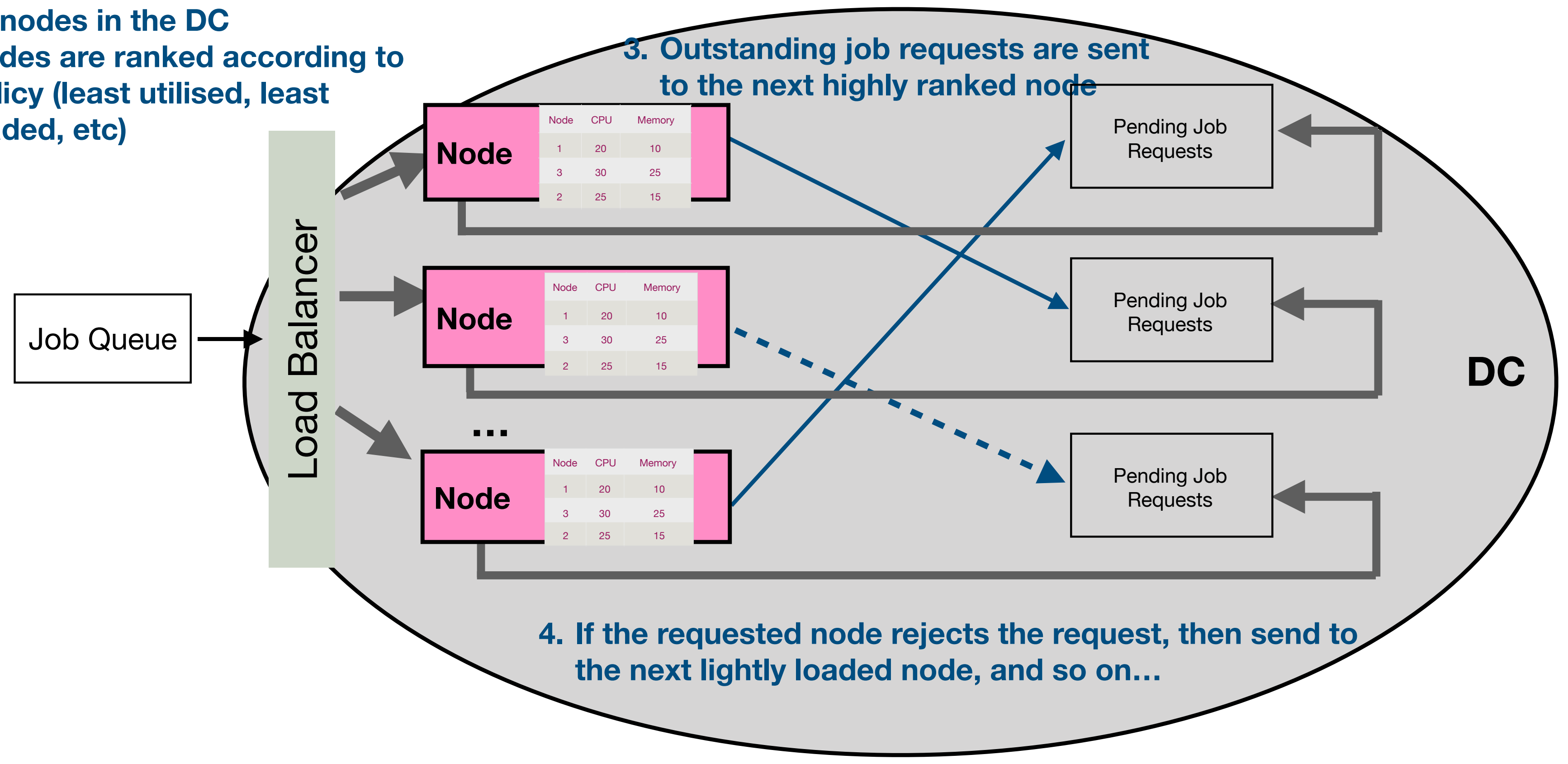
Node	CPU	Memory
A	20	10
C	30	25

Node	CPU	Memory
A	20	10
C	30	25

- ☑ ***Could be at cluster level instead of node level***
- ☑ ***Information sent could be***
  - ❖ ***Current Resource Utilisation***
  - ❖ ***Forecast of future resource utilisation based on learnt patterns***

Better Load Balancing And Utilisation Using Up-to-Date Timely Global View For Scheduling

1. All nodes have information on all nodes in the DC
2. Nodes are ranked according to policy (least utilised, least loaded, etc)



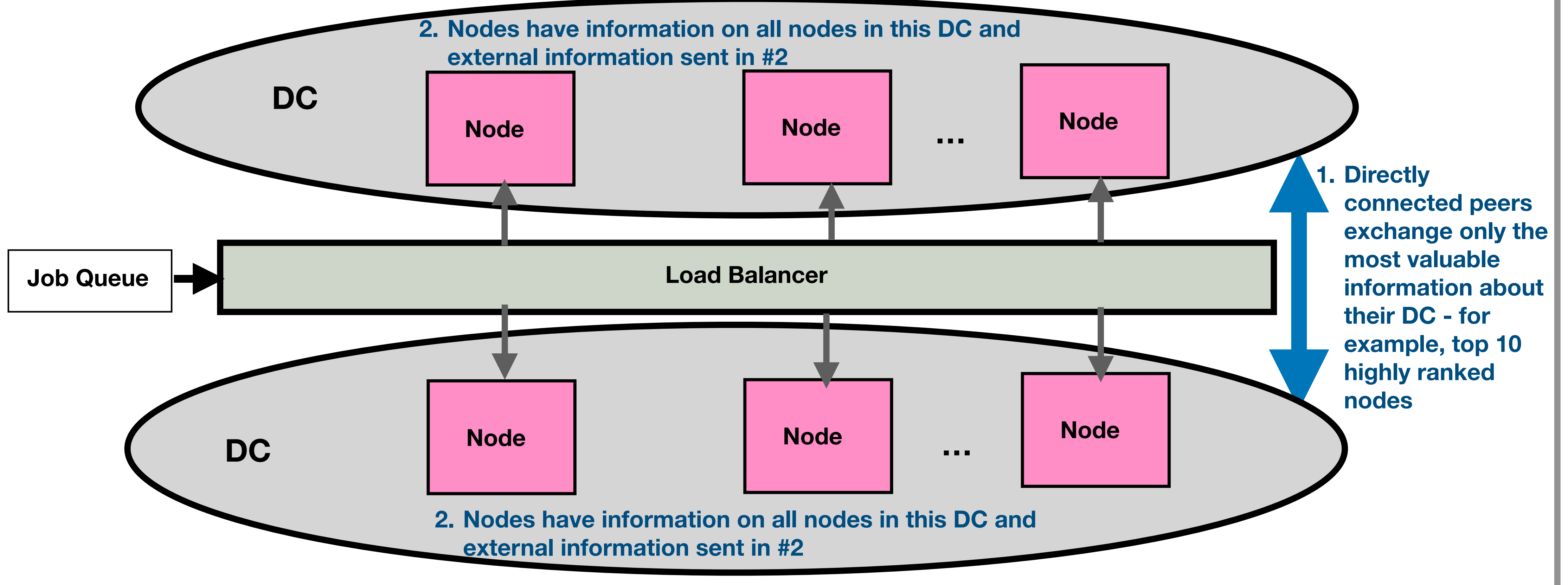
Node	CPU	Memory
1	20	10
3	30	25
2	25	15

3. Outstanding job requests are sent to the next highly ranked node
4. If the requested node rejects the request, then send to the next lightly loaded node, and so on…

Intra-DC Load Balanced Scheduling

Various Design Approaches

- ***Request to accept an incoming job is sent to a couple of nodes according to ranking, instead of just one.***
- ***Prediction and learning***
- ***Suggestions?***



2. Nodes have information on all nodes in this DC and external information sent in #2

1. Directly connected peers exchange only the most valuable information about their DC - for example, top 10 highly ranked nodes

2. Nodes have information on all nodes in this DC and external information sent in #2

Inter-DC Load Balanced Scheduling