# Research and Exploit of Resources Sharing Strategy at IHEP

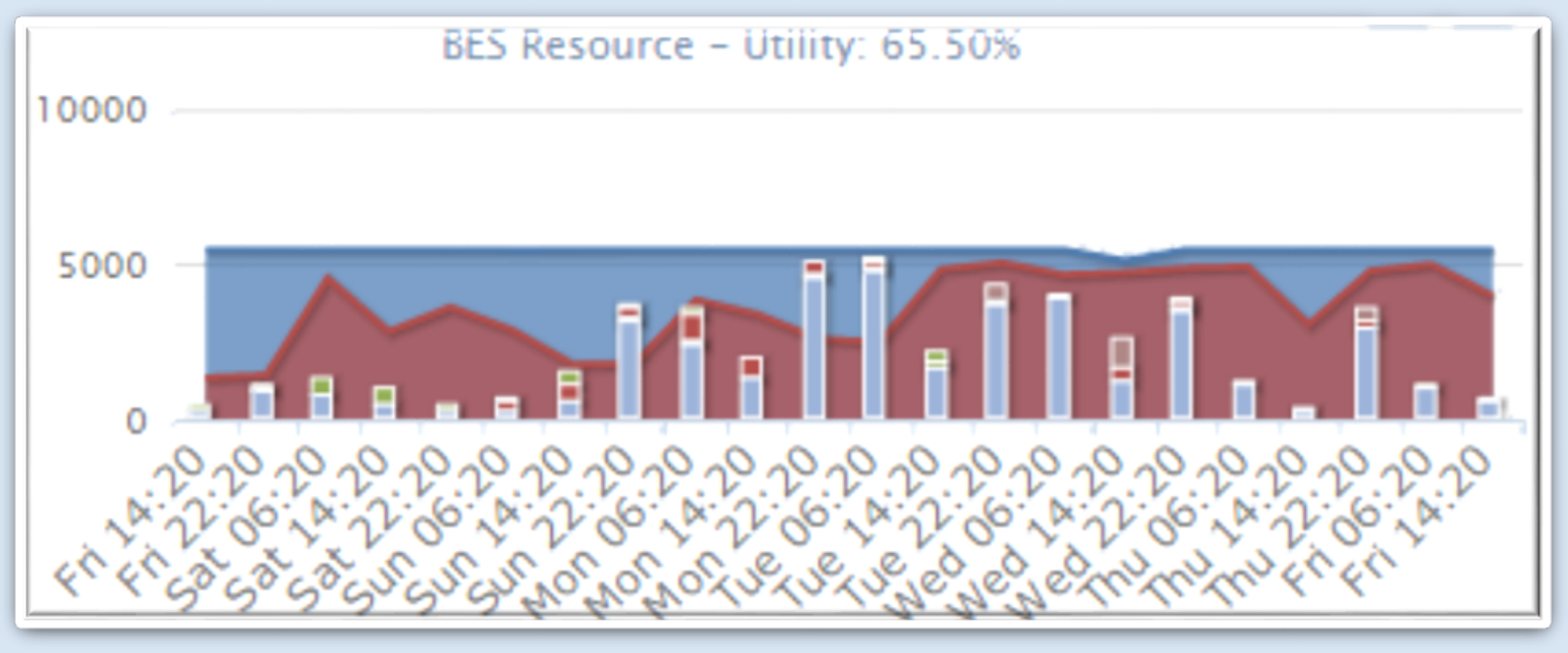
Xiaowei Jiang, Jingyan Shi, Jiaheng Zou, Qingbao Hu, Ran Du
jiangxw@ihep.ac.cn, shijy@ihep.ac.cn, zoujh@ihep.ac.cn, huqb@ihep.ac.cn, duran@ihep.ac.cn

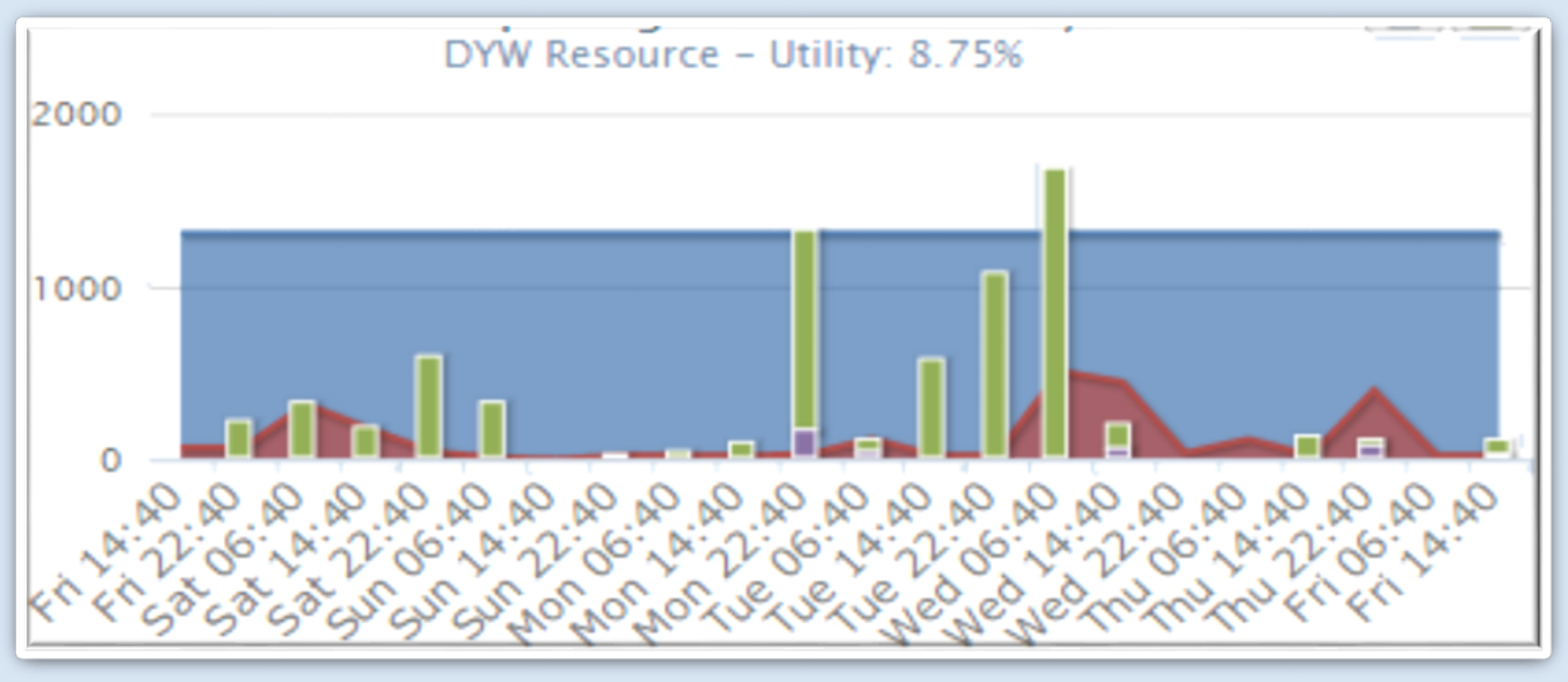
## Background

At IHEP, computing resources are contributed by different experiments including BES, JUNO, DYW, HXMT, etc. The resources were divided into different partitions to satisfy the dedicated experiment data processing requirements. Utilizing Torque&Maui, IHEP had maintenanced a local cluster with 50 queues serving for above 10 experiments for more than 10 years.

## Problems

The separated resource partitions leaded to imbalance resource load. As shown in the two following diagrams, BES resource partition is quite busy on some time points, even with lots of jobs in queue; Oppositely, DYW resources stay idle for a long time. However, sometimes the situations are contrary.



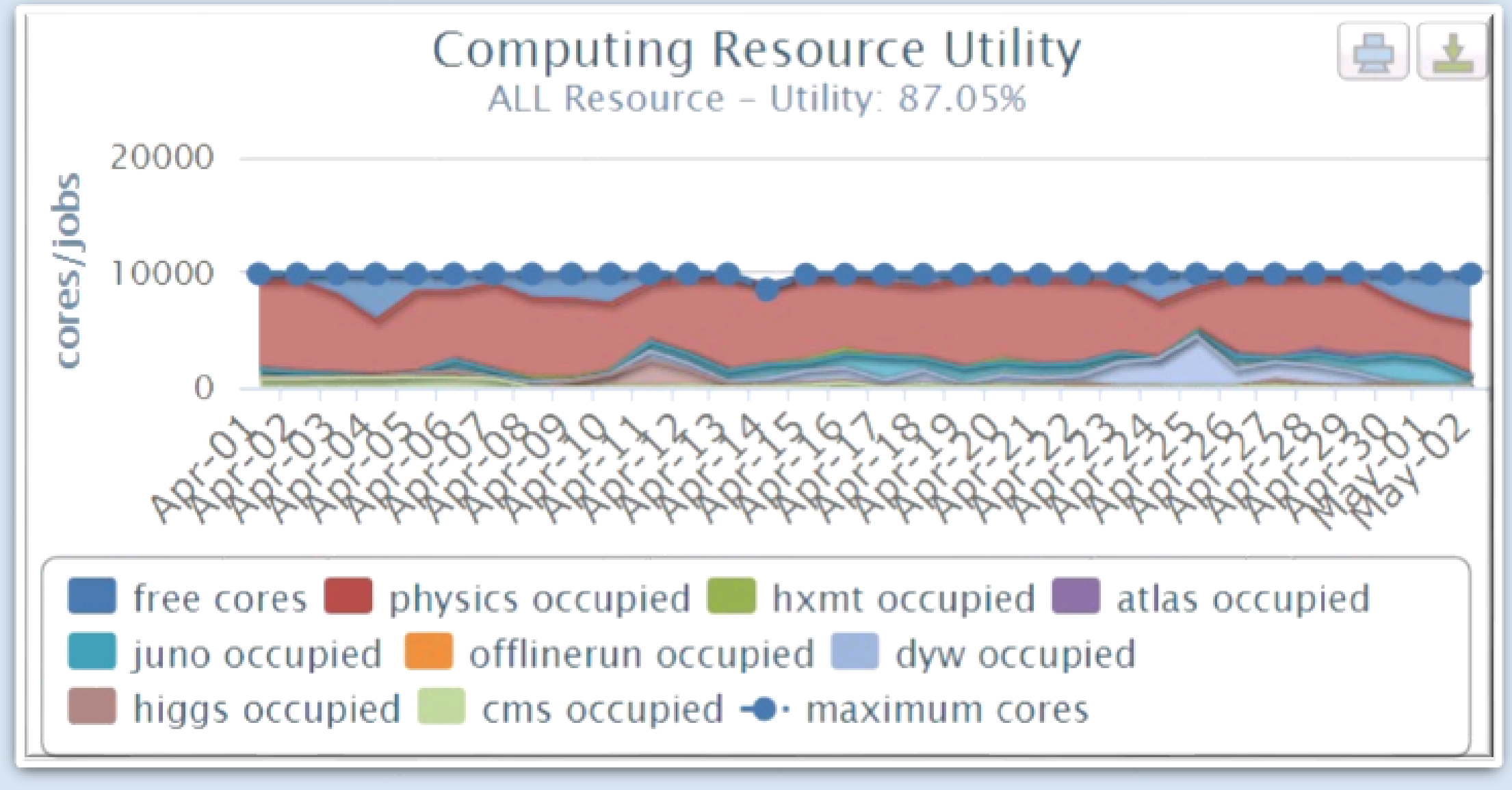


## Basic Method

After migrating resources from Torque&Maui to HTCondor in 2016, job scheduling efficiency has been improved a lot. But resource usage can not increase more due to the separated resources among all experiments. To aim at breaking resource isolation, an efficient sharing strategy was presented to improve the overall resource usage. This strategy consists of two core components: sharing policy and Central Controller. **Sharing policy** dynamically defines the sharing quota for each experiment group. **Central Controller** manages the sharing information which is published to worker nodes automatically.

## Conclusion

With sharing strategy, overall resource usage of IHEP computing cluster has dramatically increased from around 50% to around 90%. The total wall-time without sharing strategy in 2016 is 40,645,124 CPU hours, while it's 73,341,585 CPU hours with sharing strategy in 2017, increasing by 80.44%. The results indicate sharing strategy is efficient and integrally promotes experiment data processing.



## Sharing Policy

In this sharing policy, all resources are collected in an unique pool which is shared by all experiment group. Resources of each experiment are divided into two parts: dedicated resource and sharing resource. The slots in dedicated resource only run jobs of own experiment group, and the slots in sharing resource are shared by jobs of all experiment groups. $N_{all}$(number of total resources) and $Ng_i$(number of resources for group i) are constrained by the following conditions:

$$N_{all} = \sum_{i=group_0}^{group_n} Ng_i \ , \ (i = physics, juno, dyw, hxmt, ...)$$

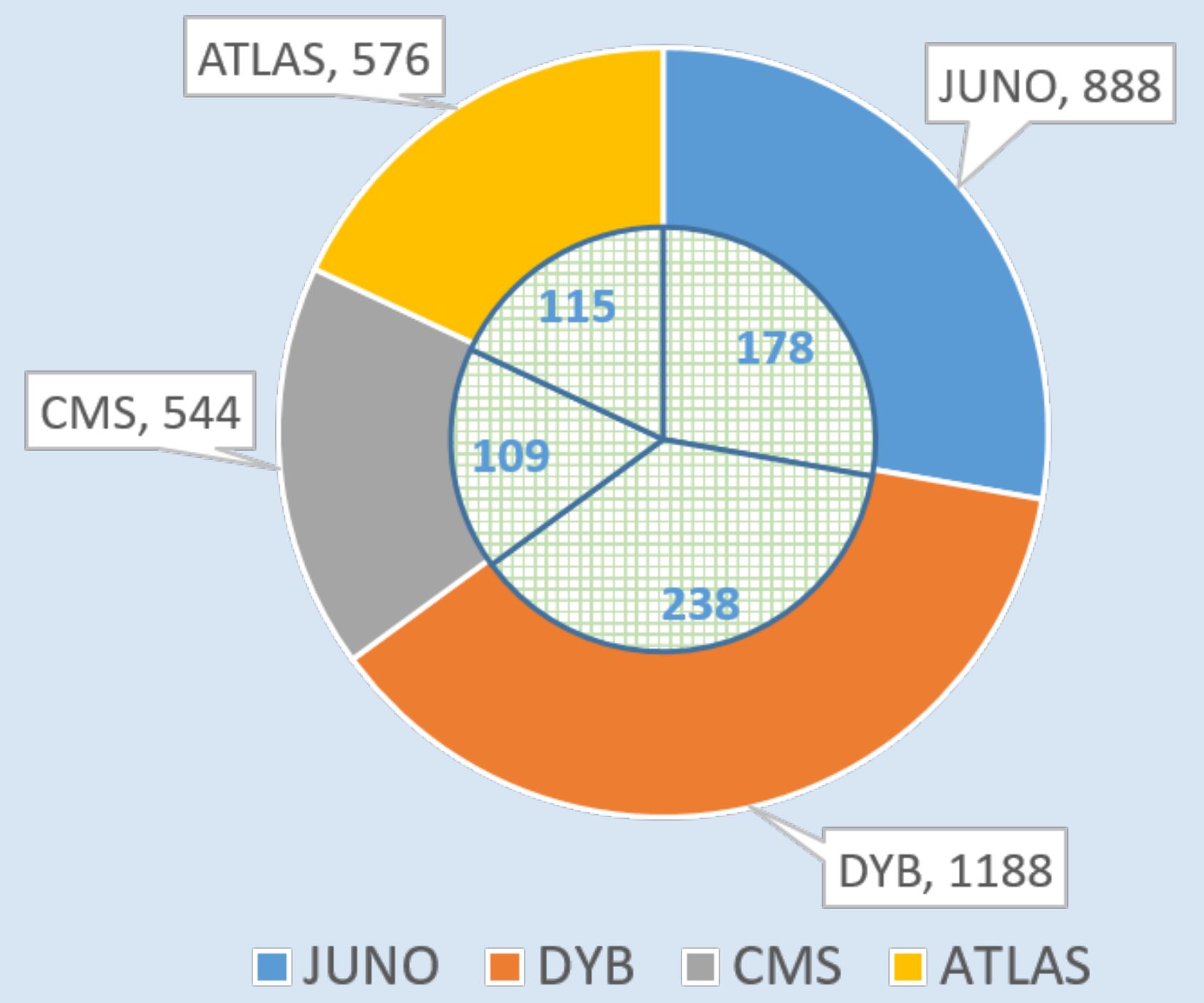
$$Ng = Ng_{dedicated} + Ng_{sharing}$$

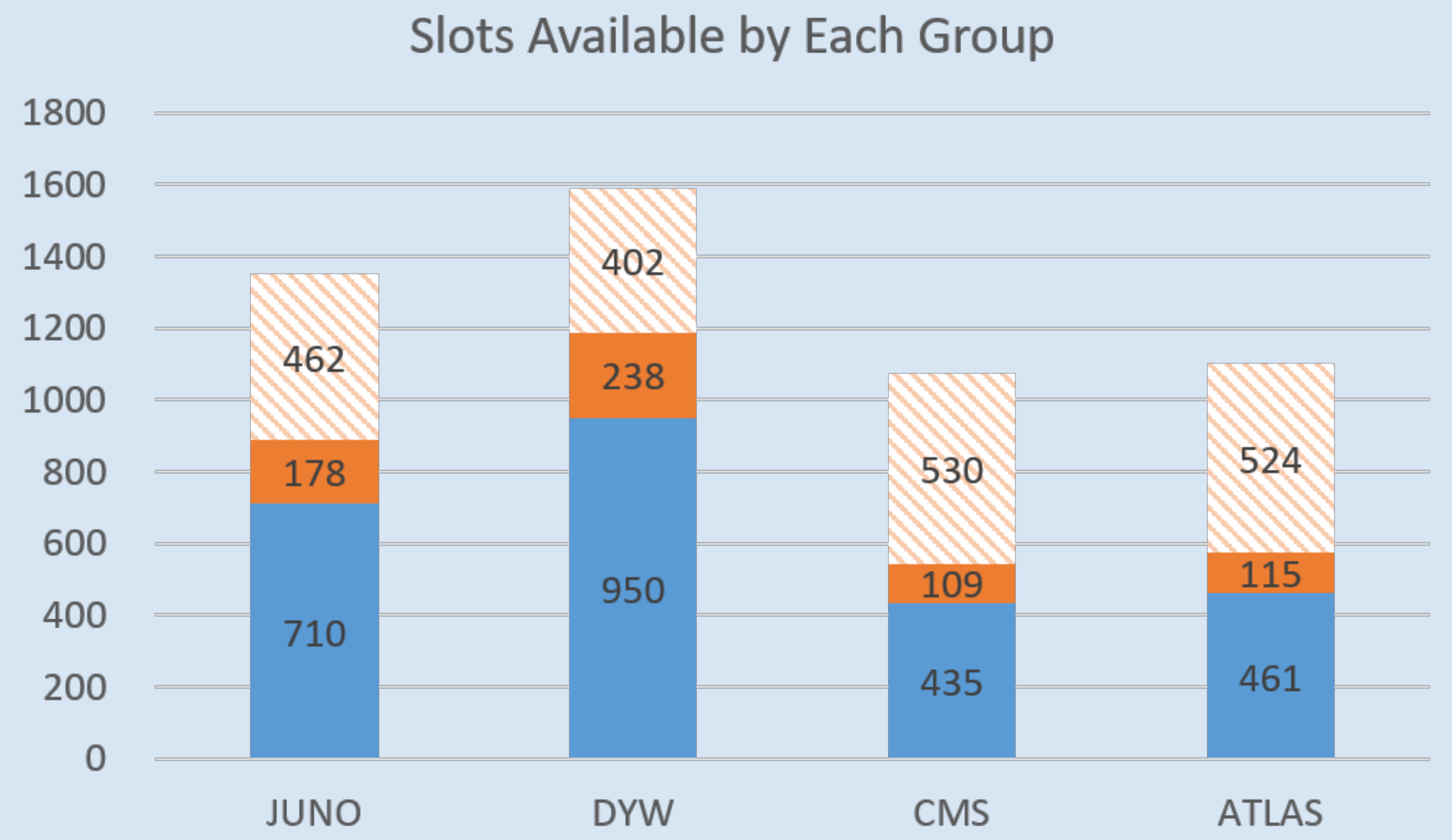
$Rate_{sharing}$ is defined for evaluating the number of sharing resources contributed by group(each group has its own sharing rate), So $Ng_{sharing}$(number of sharing resources) and $Ng_{dedicated}$(number of dedicated resources) are evaluating as the simple expressions below.

$$Ng_{sharing} = Rate_{sharing} * Ng$$

$$Ng_{dedicated} = (1 - Rate_{sharing}) * Ng$$

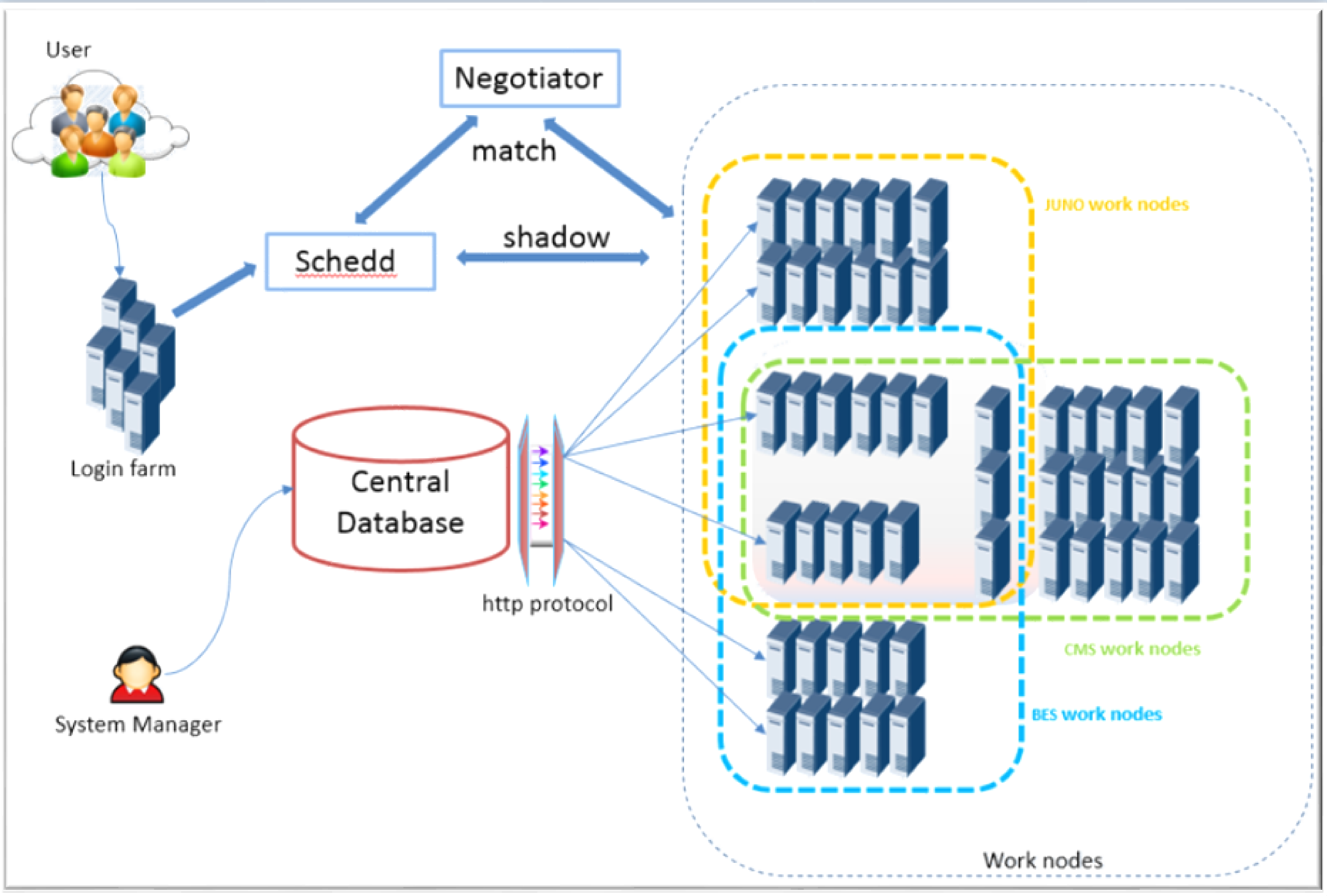
Default sharing rate is 0.2. To maximize sharing effect, the ratio is dynamically adjusted between 0 and 1 based on amount of jobs from each experiment group. The following diagrams show a case of sharing and dedicated resources, which is with the resource status in a historic moment.





## Central Controller

Central controller system was developed to allocate resources for each experiment group. This system consists of two sides: server and client. A management database is built at server side, which is storing resource, group and experiment information. Once the sharing ratio needs to be adjusted, resource group will be modified and updated into database. The resource group information is published to the server buffer in real time. The Client periodically pulls resource group information from server buffer via https protocol. And resource scheduling conditions at client side is corrected based on the dynamic resource group information. By this process, share ratio can be regulated dynamically.


User
Negotiator
match
Schedd
shadow
Login farm
Central Database
http protocol
System Manager
JUNO work nodes
CMS work nodes
BES work nodes
Work nodes