Grid Engine Advance Reservation

6.x advance reservation solutions

Andreas Haas

Software Engineering

Sun Grid Engine

Sun microsystems

We make the net work.

6.x solutions roadmap

1. Flexible dispatch priority scheme combining new resource request based policy with existing policies
2. advance reservation to solve resource sharing job starvation problem
3. fix deficiencies with time/date based resource disposition
4. round-off new resource sharing capabilities with timely and aimed job preemption
5. propagate job net dispatch priorities

Flexible priority scheme (RRDP)

- Normalized static urgency combining time dependent factors and RRDP
  - Per resource weighting factor
  - deadline weighting factor
  - Waiting time weighting factor

  } **NSU**
- Normalized Ticket Amount combines Enterprise Edition Ticket concepts
  - share tree based policy
  - functional policy
  - override policy

  } **NTA**

- NSU/NTA linear combination finally used as priority value

Flexible priority scheme

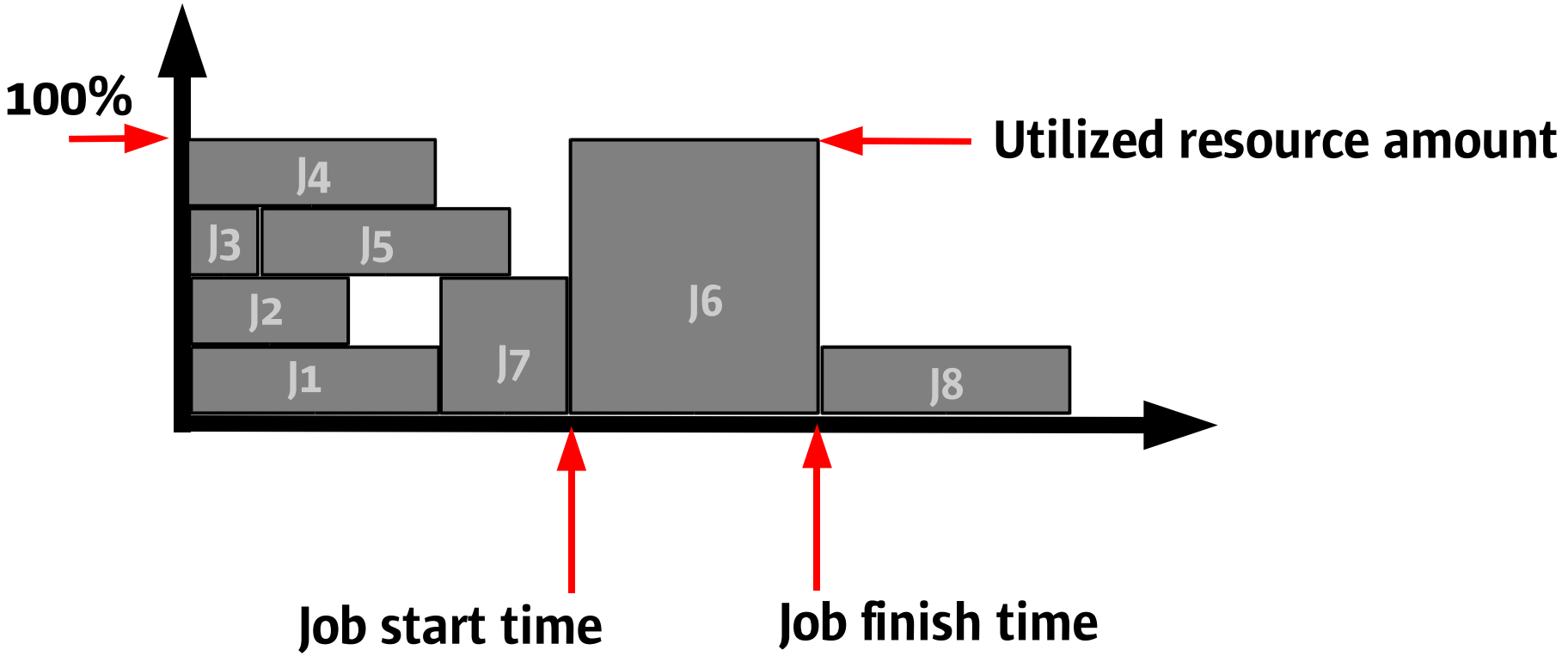
Example: A 2 time parallel job with -l h_vmem=4G,mylic=0.5 resource requests that waits since 1 hour 25 minutes and must have been started after one hour to meet it's deadline. Only functional ticket policy is in use.

```
Slots=1    (implicit)    (2 times, weight 10000)      =  20000.00
h_vmem=4G                (2 times, weight 0.000005)   =  42949.00
mylic=0.5                (2 times, weight 100000)     = 100000.00
waiting 1:25:10s         (weight 11.50)               =  58765.00
start latest in 1:0:0s (weight 180000000)             =  50000.00
static urgency resulting                              = 271714.00
normalized static urgency (NSU)                    -->      0.22
          0 stix +      133675 ftix +         0 otix =    133675
normalized ticket priorty (NTA)                    -->      0.51
NSU 0.22 (weight 0.75) + NTA 0.51 (weight 0.25)    =    0.2925
```

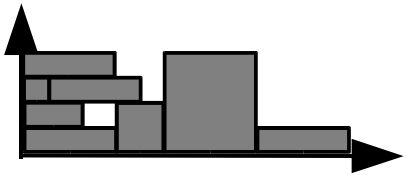
Automated control of job dispatch order based on resource request

Advance Reservation/Backfilling: Resource utilization diagram

1. Register each assignment in a per resource utilization diagram
2. Use diagram information to decide about further assignments

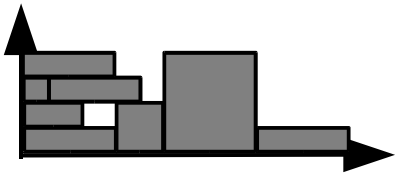


Advance Reservation/Backfilling: Resource Utilization Tree (RUT)

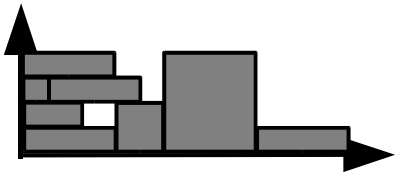


Cluster

Cluster-wide schedule

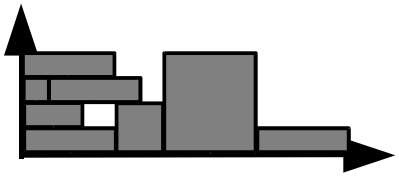


Saruman.sun.com

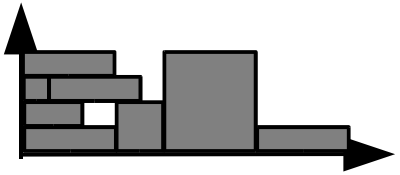


Gandalf.sun.com

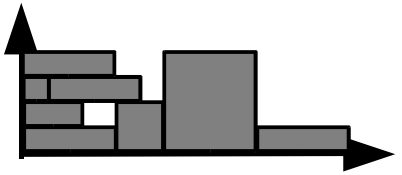
Per host schedules



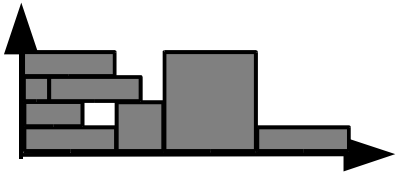
High@saruman



Low@saruman



High@gandalf



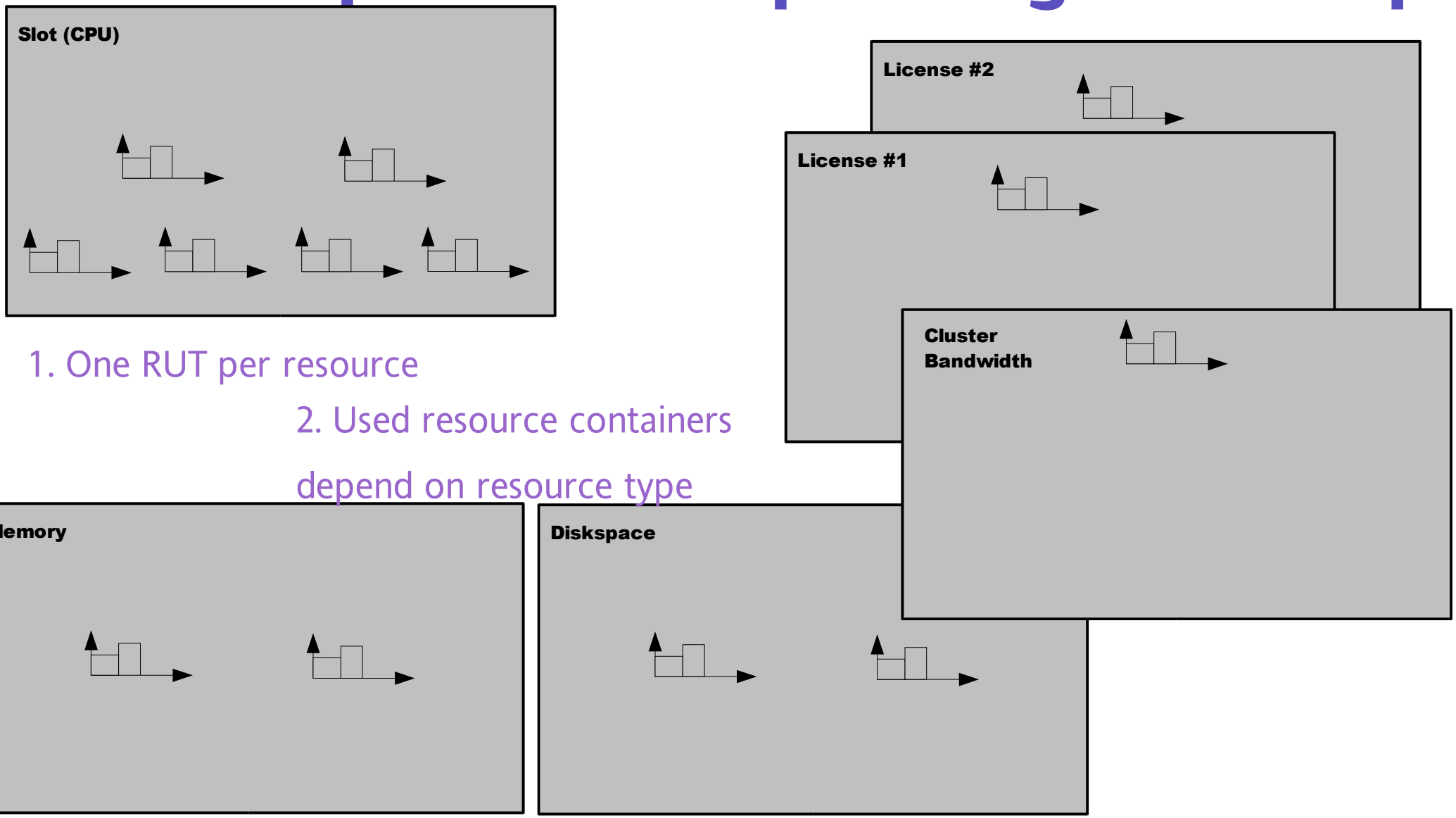
Low@gandalf

Per queue instance schedules

1. Going the path down to the leafes allows verifying resource availability for any possible assignment

2. Complete RUT used only if resource were configured in all resource containers

Advance Reservation/Backfilling Multiple RUTs depending on set-up

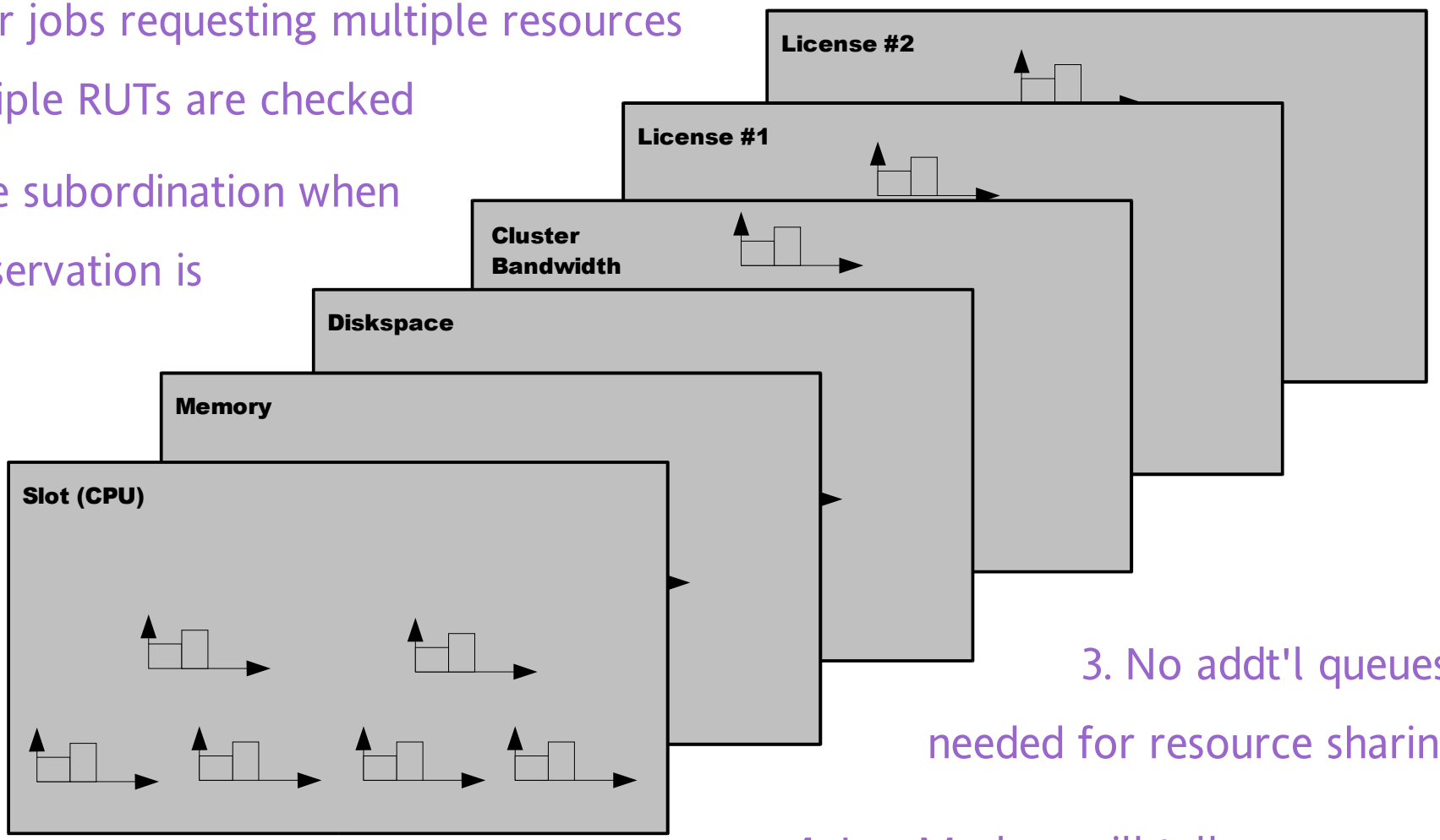


1. One RUT per resource
2. Used resource containers depend on resource type

Advance Reservation/Backfilling

1. For jobs requesting multiple resources multiple RUTs are checked

2. No queue subordination when advance reservation is enabled



3. No addt'l queues needed for resource sharing

4. Lev Markov will tell you more about the planning algorithm

Solution for time/date based resource disposition

- Jobs will no longer be dispatched in queues that close by calendar if job can't finish
- When queue is closed resource reservation is done for the time after

Timely and aimed job preemption

- Algorithm considers a job X be preempted by job Y, only if
  - job X blocks resources required by Y
  - X preemption is prerequisite to start Y earlier than w/o preemption
  - Job X would not finish soon
  - further conditions
- The preempting job Y is bound during certain time to the preempting assignment
- Preemption algorithm enhances base advance reservation algorithm (Lev)

Priority propagation in job nets

- For predecessor jobs the maximum priority of all successors jobs will be used as dispatch priority
- Priority propagation affects only dispatch order, but not priority used when priorities are compared to decide about preemption