Outline

Why we need partitionable slots
How they’ve worked since 7.2
What’s new in 7.8
What’s still left to do
What’s the Problem?

Queue, Uncore & I/O

Shared L3 Cache

Core

Core

Core

Core

Core

Memory Controller
Example Machine:

- 8 cores
- 8 Gigabytes memory
- 2 disks
The old way
(Still the default)

$ condor_status

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
<th>ActvtyTime</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:slot1@chevre.cs.wi">slot1@chevre.cs.wi</a> LINUX</td>
<td>X86_64 Unclaimed Idle</td>
<td>0.110</td>
<td>1024</td>
<td>0+00:45:04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot2@chevre.cs.wi">slot2@chevre.cs.wi</a> LINUX</td>
<td>X86_64 Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+00:45:05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot3@chevre.cs.wi">slot3@chevre.cs.wi</a> LINUX</td>
<td>X86_64 Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+00:45:06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot4@chevre.cs.wi">slot4@chevre.cs.wi</a> LINUX</td>
<td>X86_64 Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+00:45:07</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot5@chevre.cs.wi">slot5@chevre.cs.wi</a> LINUX</td>
<td>X86_64 Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+00:45:08</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot6@chevre.cs.wi">slot6@chevre.cs.wi</a> LINUX</td>
<td>X86_64 Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+00:45:09</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot7@chevre.cs.wi">slot7@chevre.cs.wi</a> LINUX</td>
<td>X86_64 Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+00:45:10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot8@chevre.cs.wi">slot8@chevre.cs.wi</a> LINUX</td>
<td>X86_64 Unclaimed Idle</td>
<td>0.000</td>
<td>1024</td>
<td>0+00:45:03</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total Owner Claimed Unclaimed Matched Preempting Backfill

| X86_64/LINUX | 8 | 0 | 0 | 8 | 0 | 0 | 0 |
| Total        | 8 | 0 | 0 | 8 | 0 | 0 | 0 |
Problem: Job Image Size

Image sizes

Count

Image Size in Mb
Simple solution: Static non-uniform memory

# condor_config
NUM_SLOTS_TYPE_1 = 1
NUM_SLOTS_TYPE_2 = 7
SLOT_TYPE_1 = mem=4096
SLOT_TYPE_2 = mem=auto
Result is

$ condor_status

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:slot1@chevre.cs.wi">slot1@chevre.cs.wi</a></td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.150</td>
<td>4096</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot2@chevre.cs.wi">slot2@chevre.cs.wi</a></td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>585</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot3@chevre.cs.wi">slot3@chevre.cs.wi</a></td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>585</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot4@chevre.cs.wi">slot4@chevre.cs.wi</a></td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>585</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot5@chevre.cs.wi">slot5@chevre.cs.wi</a></td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>585</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot6@chevre.cs.wi">slot6@chevre.cs.wi</a></td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>585</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot7@chevre.cs.wi">slot7@chevre.cs.wi</a></td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>585</td>
<td></td>
</tr>
<tr>
<td><a href="mailto:slot8@chevre.cs.wi">slot8@chevre.cs.wi</a></td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed Idle</td>
<td>0.000</td>
<td>585</td>
<td></td>
</tr>
</tbody>
</table>

Total Owner Claimed Unclaimed Matched Preempting

Backfill

X86_64/LINUX 8 0 0 8 0
Better, still not good

- Job requirements:
  - Requirements = memory > 2048
- How to steer small jobs to small slot?
  - Trivia question in classads?
- How to pick correct sizes?
- Changes require startd restarts
8 Gb machine partitioned into 5 slots

4Gb Slot | 1Gb | 1Gb | 1Gb | 1Gb

4 Gb Job | 1Gb | 1Gb | 1Gb | 1Gb
8 Gb machine partitioned into 5 slots

4Gb Slot
1Gb
1Gb
1Gb
1Gb
8 Gb machine partitioned into 5 slots

4Gb Slot  1Gb  1Gb  1Gb  1Gb

1Gb

4 Gb Job

7 Gb free, but idle job
New: Partitionable slots

- Work in progress
- First landed in 7.2
- More work in 7.8
- Even more goodness to come...
- But very usable now
The big idea

› One “partionable” slot
› From which “dynamic” slots are made
› When dynamic slot exit, merged back into “partionable”
› Split happens at claim time
(cont)

- Partionable slots split on
  - Cpu
  - Disk
  - Memory
  - (Maybe more later)
- When you are out of one, you’re out of slots
3 types of slots

› Static (e.g. the usual kind)
› Partitionable (e.g. leftovers)
› Dynamic (usableable ones)
  • Dynamically created
  • But once created, static
8 Gb Partitionable slot

1Gb 1Gb 1Gb 1Gb 4Gb
8 Gb Partitionable slot

1Gb  1Gb  1Gb  5Gb
How to configure

NUM_SLOTS = 1
NUM_SLOTS_TYPE_1 = 1
SLOT_TYPE_1 = cpus=100%
SLOT_TYPE_1_PARTITIONABLE = true
Looks like

$ condor_status

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
<th>ActvtyTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot1@c</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.110</td>
<td>8192</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Total</th>
<th>Owner</th>
<th>Claimed</th>
<th>Unclaimed</th>
<th>Matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>X86_64/LINUX</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
When running

$ condor_status

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot1@c</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.110</td>
</tr>
<tr>
<td>slot1_1@c</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
</tr>
<tr>
<td>slot1_2@c</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
</tr>
<tr>
<td>slot1_3@c</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
</tr>
</tbody>
</table>
All this in 7.2

What are the problems?

› Slow matching
› Broken for parallel universe
› Dedicated slots users broken
› Fragmentation
› Selection of dynamic slots sizes tricky
Fixed in 7.8

› Matching faster
  • CLAIM_PARTITIONABLE_LEFTOVERS
    • = false to make slow again..

› Parallel universe fixed
› Dedicated slot users fixed
› Support for defragging!
Answer to trivia question

› Requirements = (Memory > 1024)
  • How can startd parse?

› It Can’t!
THIS IS BIG!

› Memory requirements deprecated
› Don’t do
   • Requirements = memory > 1024
› Generates a warning now:
   › condor_submit submit7
   › Submitting job(s)...
   › WARNING: your Requirements expression refers to TARGET.Memory. This is obsolete. Set request_memory and condor_submit will modify the Requirements expression as needed.
Instead, use

- `request_memory = 2048 # mbytes`
- Same is true for disk, cpus
  - `request_disk = 16384 # kbytes`
  - `request_cpus = 1`
- Requirements automatically fixed
I have to change all my submit files?

► There’s a knob for that ™
► JOB_DEFAULT_REQUESTMEMORY
► JOB_DEFAULT_REQUESTDISK
► JOB_DEFAULT_REQUESTCPUS
► submit side defaults
► Can be expressions...
I don’t want to change the config file

JOBDefaultRequestMemory
  ifthenelse(MemoryUsage != UNDEF, MemoryUsage, 1)

JOBDefaultRequestCPUs
  1

JOBDefaultRequestDisk
  DiskUsage
What about the startd side?

- Startd has a say, too:
  ```c
  MODIFY_REQUEST_EXPR_REQUESTCPUS
  quantize(RequestCpus, \{1\})
  $$\text{MODIFY\_REQUEST\_EXPR\_REQUESTMEMORY}$$
  quantize(RequestMemory, \{\text{TotalSlotMem}/\text{TotalSlotCpus} / 4\})
  $$\text{MODIFY\_REQUEST\_EXPR\_REQUESTDISK}$$
  quantize(RequestDisk, \{1024\})
  ```
Why quantize?

- Allow slot reuse

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot1@c LINUX</td>
<td>X86_64 Unclaimed</td>
<td>Idle</td>
<td>0.110 4096</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot1_1@c LINUX</td>
<td>X86_64 Claimed</td>
<td>Busy</td>
<td>0.000 2048</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot1_2@c LINUX</td>
<td>X86_64 Claimed</td>
<td>Busy</td>
<td>0.000 1024</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>slot1_3@c LINUX</td>
<td>X86_64 Claimed</td>
<td>Busy</td>
<td>0.000 1024</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Also holds for cpus

- Much easier way to do whole machine
- Basically same as memory requirements
- Easier to set up than Parallel universe
## Fragmentation

<table>
<thead>
<tr>
<th>Name</th>
<th>OpSys</th>
<th>Arch</th>
<th>State</th>
<th>Activity</th>
<th>LoadAv</th>
<th>Mem</th>
</tr>
</thead>
<tbody>
<tr>
<td>slot1@c LINUX</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Unclaimed</td>
<td>Idle</td>
<td>0.110</td>
<td>4096</td>
</tr>
<tr>
<td>slot1_1@c LINUX</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
<td>2048</td>
</tr>
<tr>
<td>slot1_2@c LINUX</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
<td>1024</td>
</tr>
<tr>
<td>slot1_3@c LINUX</td>
<td>LINUX</td>
<td>X86_64</td>
<td>Claimed</td>
<td>Busy</td>
<td>0.000</td>
<td>1024</td>
</tr>
</tbody>
</table>

Now I submit a job that needs 8G – what happens?
Solution: New Daemon

- condor-defrag (new in 7.8)
  - One daemon defrags whole pool
    - Central manager good place to run
- Scan pool, try to fully defrag some startds
- Only looks at partitionable machines
- Admin picks some % of pool that can be “whole”
Oh, we got knobs...

DEFRAG_DRAINING_MACHINES_PER_HOUR
default is 0
DEFRAG_MAX_WHOLE_MACHINES
default is -1
DEFRAG_SCHEDULE
  • graceful (obey MaxJobRetirementTime, default)
  • quick (obey MachineMaxVacateTime)
  • fast (hard-killed immediately)
Defrag vs. Preemption

- Defrag can be general purpose
  - Looks only at startds, not at demand
  - Can also preempt non-partitionable slots
    - (if so configured)

- Negotiator preemption looks at 2 jobs
Advanced Topics

› More than one Partitionable slot
› Mix and match partitionable slots
› Overcommitting partitionable slots
› Parallel universe
Future work

› Claiming partitionable slots
  • So RANK based preemption works
› condor_q -analyze
› More knobs!
Thank you