# HTCondor Integration with the Linux Out-Of-Memory Killer

Brian Bockelman

## Overview

The integration of Condor with Linux kernel cgroups HTCondor with a mechanism to measure the amount of RAM the kernel has allocated to the job. Prior mechanisms in the procd were estimates of the RAM usage of varying precision (often quite poor in the case of complex jobs); cgroups provides to-the-byte precision, as this is the number provided by the kernel.

When a job’s memory usage violates the slot’s memory limit, Condor has three mechanisms:

1. **Rely on administrator-specified policies**. The sysadmin can set a policy, periodically evaluated, that may decide whether to kill the job.
2. **Have the kernel enforce a soft-limit**. The kernel will invoke the out-of-memory killer process when the system has run out of memory and the job is over its memory limit. If the job is over its memory limit but there is available system memory, the job would continue to run.
3. **Have the kernel enforce a hard limit**. The kernel will invoke the out-of-memory killer when the job first goes over its memory limit.

The downside of (1) is it often fails: by time the periodic policy is evaluated, the system may have already failed or the OOM killer decides to kill Condor instead of the job.

The downside of (2) and (3) is Condor is not currently informed when the kernel invokes the OOM killer; it cannot distinguish between normal job termination and when the kernel killed the job for violating policy. Further, the OOM killer will kill the smallest number of processes in the job necessary for freeing up memory. This may leave the job in a half-dead state.

The kernel can be instructed to notify an external process when a cgroup is out of memory instead of invoking the OOM-killer. The kernel will not run any of the processes in the job until the external processes manages to free memory in the cgroup.

I propose a new behavior for when the hard- and soft-limit mechanism triggers. The condor\_starter will register to be notified when the job runs out of memory. When this occurs, it will kill the job’s processes and put the job on hold with a new “Out of job resources” hold state.

## Proposed Solution

### Implementation Description

When cgroups are enabled and a memory limit is set, the vanilla starter will invoke a new routine to configure the OOM notification event. This notification is setup as described in <http://www.kernel.org/doc/Documentation/cgroups/memory.txt>. The notifications occur via an event FD; this FD will be given to DaemonCore to be monitored in the starter’s event loop.

When an OOM event occurs, the kernel will inform DaemonCore the file descriptor can be read. DaemonCore will invoke a second new routine in the vanilla starter which will request the procd kill the job and put the job on hold with a new hold code - CONDOR\_HOLD\_CODE\_JobOutOfResources. It will create an appropriate hold reason:

Error from slot1@hcc-briantest.unl.edu: Job has gone over memory limit of 100 megabytes.

There are no new configuration knobs. The backward compatibility issues are minimal; a new hold state is added, and sites will see the new behavior of jobs being put on hold instead of inexplicably killed. This is arguably the ‘correct’ behavior when enforcing memory limits – a relatively unused mode in no stable-series release - so there is no need to provide a compatibility mode with the old behavior.

### Verification

This can be tested with a job that purposely goes over its memory request and configuring the starter to use the “hard” memory limit. The job should not run to completion; instead, it should be put in the hold state with a message along the lines of the one listed above.

The proposed patch has been run at Nebraska for approximately 6 months.

## Development Plan

The implementation and verification described above has already been performed. If there are no changes to the initial version of the design document, there is no additional development needed.

The changes are available in the “oom” branch of [git@github.com:bbockelm/htcondor.git](mailto:git@github.com:bbockelm/htcondor.git).