

# Handling Signals in the Job Wrapper

*EGEE 07 Conference  
Budapest, 1-5 October 2007*

*Francesco Giacomini, Marco Cecchi – INFN*

- **Jobs may be terminated by the batch system, in particular when they exceed the allocated time slot**
  - The BS does so sending one or more signals, eventually SIGKILL
  - If this situation is not handled appropriately
    - The information about what really happened to the job will be little or none
    - The job will *probably* end in “Done Failed” and will be resubmitted

- **Wait to be signalled**
  - The batch system usually tries to kill the job gently with catchable signals (e.g. SIGTERM) before using the uncatchable SIGKILL
  - Catch the *soft* signals and try to do quickly something sensible
- **Anticipate the termination**
  - Does it make any sense?
  - It could be useful if the things to do cannot be done quickly enough
  - Need to know from the batch system how long the time slot is

- **bkill command**
  - *Sends signals to kill, suspend, or resume unfinished jobs*
  - *On UNIX, SIGINT and SIGTERM are sent to give the job a chance to clean up before termination, then SIGKILL is sent to kill the job. The time interval between sending each signal is defined by the JOB\_TERMINATE\_INTERVAL parameter in lsb.params(5).*
- **The job is killed when some resource usage limit, if set, is exceeded**
  - CPU, memory, threads, stack, swap, ...
- **If the CPU limit is exceeded, SIGXCPU is sent first**

- **qdel command**
  - *A batch job being deleted by a server will be sent a SIGTERM signal following by a SIGKILL signal. The time delay between the two signals is an attribute of the execution queue from which the job was run (settable by the administrator).*
- **kill\_delay**
  - *The time delay between the sending of SIGTERM and SIGKILL when a qdel command is issued against a running job (default 2)*
- **For maui:**
  - *The parameter RESOURCELIMITPOLICY controls which resources are limited, what limit policy is enforced per resource, and what actions the scheduler should take in the event of a policy violation.*

- **Something is already in place for the job wrapper that comes with the WMS 3.1**
  - SIGTERM (only) is trapped
    - Common denominator between LSF, PBS and probably every BS under UNIX
  - The handler does the following:
    1. An appropriate message is written in the jw standard output  
“Job has been terminated by the batch system”
    2. Done Failed is logged with the same message
    3. Maradona file is transferred (a copy of the jw stdout)
    4. The workspace is cleaned up
- **Not satisfactory**
  - e.g. It doesn't work for LSF

- **Trap a more complete set of signals**
  - Differentiate by batch system? How?
- **Refine the handler**
  - Include the received signal in the log message
  - Do the operations in parallel where applicable
    - They are I/O bound
- **Questions**
  - Can we avoid to cleanup the workspace?
  - Should the OSB be transferred?
  - If so, should the job be waited for or even be killed -9?
  - What should the status of the job be?
  - Should the behavior be configurable? By whom?