

Raytheon

HTCCondor Week 2017

University of Wisconsin — Madison, Wisconsin — May 2-5, 2017

Detecting & Managing Job Events & Progress with an HTCCondor Update Job Info Hook

Integrated Defense Systems

Michael V. Pelletier

May 2017

This document does not contain technology or Technical Data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations. eTPCR IDS-10494

Copyright © 2017 Raytheon Company. All rights reserved.
Third-party content rights as indicated.

About Michael V. Pelletier

Raytheon

- Joined Raytheon IT on March 2, 2009
- 2009-2015: Sysadmin at Raytheon Missile Defense Center
- 2015-present: Program Support & Delivery – Architecture
- Highlights
 - 2009 Excellence in Information Solutions Award
 - 2010 Authors & Inventors Award
 - 2011-2013 Technical Honors Peer Recognition
 - 2013 Excellence in Engineering & Technology Award
 - In recognition of HTCondor implementation
 - 2016 Excellence in Information Technology Award
- Prior Roles
 - Nortel: 2000-2008
 - Taos: 1998-2000
 - TechTeam Global: 1992-1998
 - University of Michigan Engineering: 1988-1992



Photo: Raytheon Advanced Media, March 2017

About Raytheon

■ Raytheon Company

- Founded in 1922
- \$24.1 billion in 2016 sales
- 63,000 employees worldwide

■ Integrated Defense Systems

- Headquartered in Tewksbury, Mass.
- Broad portfolio of weapons, sensors, and integration systems across multiple mission areas including **air and missile defense radars**; early warning radars; naval ship operating systems; command, control, communications; air traffic systems

■ Information Technology

- “Today’s extensive High Performance Computing needs require us to continuously push the envelope to create leading-edge technical processing solutions for complex systems and program challenges.”

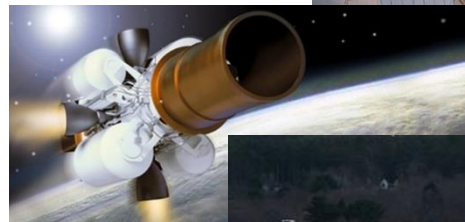
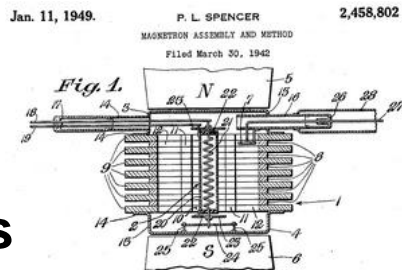


Photo: Michael Pelletier mvpel@flickr

Nearing a century of technology and innovation

Raytheon's Vision



Clockwise from top left: SM-3 launch: US Navy Pub Dom; SBX-1: © 2011 Michael Pelletier (Author); Patriot radar set, AN/TPY-2 AEU: © Raytheon (by Dan Plumpton – Raytheon Advanced Media)

What's At Stake

BOSTON
BUSINESS JOURNAL

INDUSTRY NEWS > TECHNOLOGY

Raytheon signs \$1.1B radar deal with Qatar

Feb 27, 2017, 1:20pm EST

SAUDI ARABIA | MIDDLE EAST | WORLD

SAUDI ARABIA

Saudi forces shoot down 4 Houthi ballistic missiles

Arab News | Published — Tuesday 28 March 2017

Paula Hancocks & Joshua Berlinger, March 7, 2017

CNN

World » Missile defense system that China opposes arrives in South Korea

Story highlights

THAAD is designed to take down North Korean missiles

NEW: China says it will take measure to preserve its interests

Seoul (CNN) — The first pieces of a US-built missile defense system designed to mitigate the threat of North Korean missiles arrived at the Osan Air Base in South Korea Monday night, according to the US military.

The announcement comes just a day after North Korea test-fired four ballistic missiles into the Sea of Japan,

Aegis Ashore Site in Romania Declared Operational

By: **Sam LaGrone**
May 12, 2016 12:57 PM



US MDA Photo – Public Domain

US Naval Institute News

Modeling and Simulation



*Raytheon has been deeply involved with **all phases of modeling and simulation**, from the design and development of models and simulation systems to their use in real-world, mission-critical applications.*

*...**simulations** play a central role in optimizing the value of what are **typically expensive field tests**, and in some cases simulations actually reduce the amount of field testing needed.*



Mark Russell
Vice President of
Engineering, Technology
& Mission Assurance

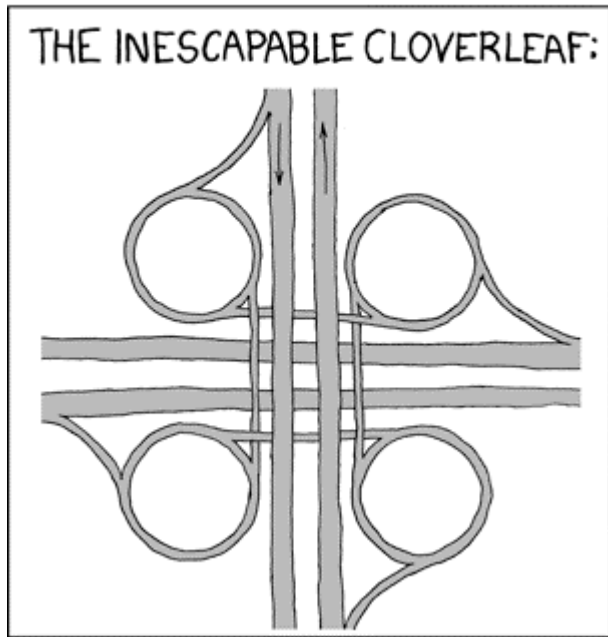
***This was
published in
January 2013...***

***It's still relevant
today, but even
more so.***

Software Peril: Hangs

HIGHWAY ENGINEER PRANKS:

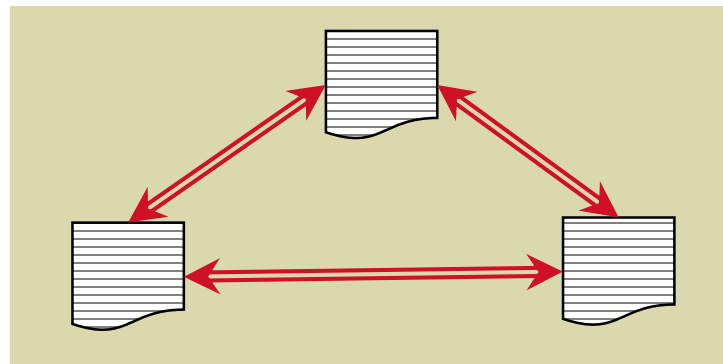
THE INESCAPABLE CLOVERLEAF:



CC-BY-NC 2.5 – <https://xkcd.com/253/>

- Hang cause is not always obvious

```
static bool retry = true;
while( retry = true ) { ... }
```
- MIL-STD-498, EIA/IEEE J-STD-016
 - Calls for entity-based software design, with defined communication interfaces between entities



Hangs in HTCondor Slots

```
condor1$ condor_q
```

```
-- Schedd: condor1 : <10.1.1.10:25399?...
```

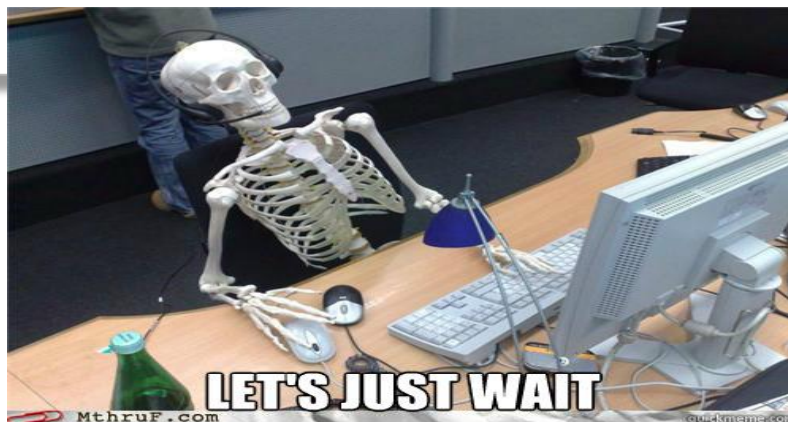
ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
611.38	joeuser	4/2 20:50	5+13:27:15	R	0	23544	(Scenario 39)

```
1 jobs; 0 completed, 0 removed, 0 idle, 1 running, 0 held, 0 suspended
```

```
condor1$
```

True Story!

- The job was only supposed to take three or four hours, and it hung in the first 10 minutes
- Wasted slot time == badput



Mrs Vivian Bennett – Little Rock – MemeSuper.com

The Usual Workarounds

▪ Runtime Cap / Deadline

```
periodic_hold = ( CurrentTime - JobCurrentStartDate > 5 * $(HOUR) )
```



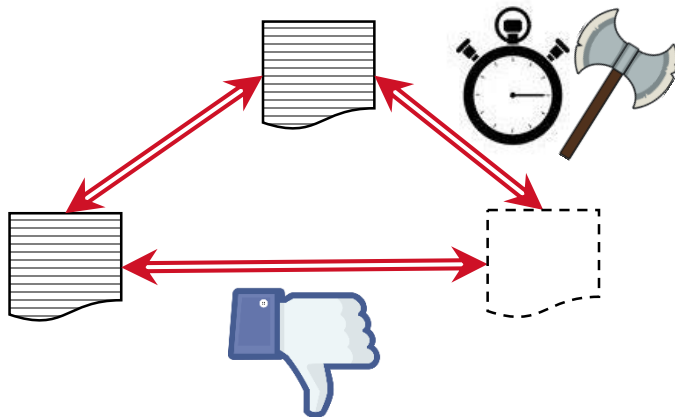
- What if an important off-nominal run really needs six hours instead of five?
 - It will be killed 5 hours into a good run
- Cap has to be configured for each type of job, and use the longest expected runtime
- What if the job hangs after five minutes?
 - It will occupy the slot until the full cap elapses four hours and 55 minutes later
- Automatic restart of intermittently-hanging jobs is impractical with long deadlines



Mrs Annette Wallace of Murietta, Georgia – ClipartFest.com

The Usual Workarounds

▪ Software-internal abort timeout



- Some of our software will run on hardware destined for destruction, so “reconnect or die trying” is the rule
- “Test like you fly” – words to live by
- Could pose security risks if used

THAAD Flight Test FTO-01, September 2013



US Army Photo – CC-BY 2.0

Solution: the “Update Job Info” Hook

▪ <Keyword>_HOOK_UPDATE_JOB_INFO

HTCondor V8.6.1 Manual Section 4.4, p.559

4.4 Hooks

A *hook* is an external program or script invoked by HTCondor.

...

The Daemon ClassAd hooks permit the *condor_startd* and *condor_schedd* daemons to **execute** hooks once or **on a periodic basis**.

HTCondor V8.6.1 Manual Section 3.5.33, p.352

For the fetch work hooks, the full path to **the program invoked by the *condor_starter* periodically as the job runs**, allowing the *condor_starter* to present an updated and augmented job ClassAd to the program.

Update Job Info Hook Operation

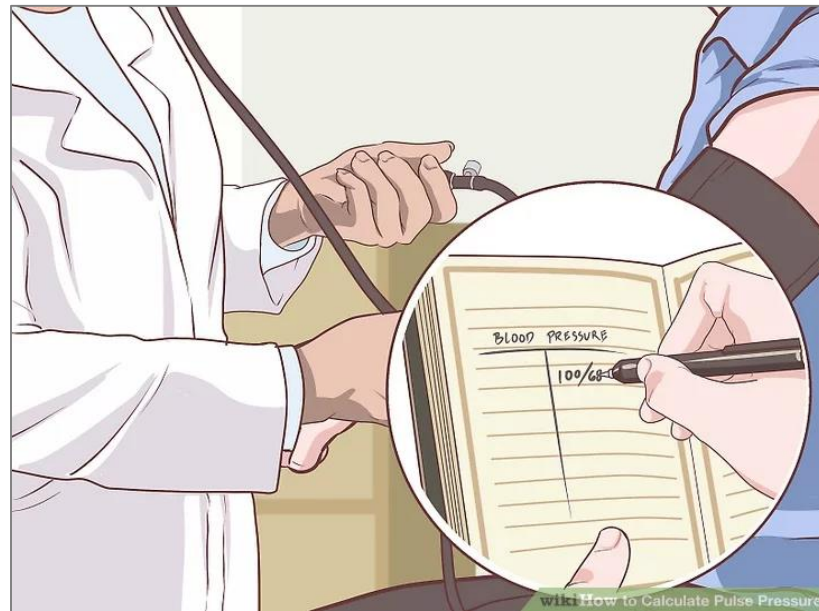


US Agricultural Research Service – Public Domain

- Invoked eight seconds after job starts and then once every five minutes
- The current job ClassAd is provided on standard input
- Runs in job's workspace as job owner
- Output and exit status are ignored
 - Updates to the job's ClassAd require *Chirp*
- Write it to run quickly and efficiently

The “Checkfile” Hook for Hang Detection

- Probe the job
 - Last modification time of output file
 - *If it stops updating, job is likely hung*
 - Pattern matching in the output file
 - *If the software generates known output that indicates a hang or other failure*
- Update the job attributes
 - `CheckfileLastModifiedTime = 1491190343`
 - `CheckfileRegexFound = False`
- Use job attributes to take action (not the hook itself)



CC BY-NC-SA – wikiHow.com

```
periodic_hold = ( CurrentTime - CheckfileLastModifiedTime > 600 )
```


Setup and Use

- Install the hook script
- Configure the pool to offer a hook keyword for it

```
CHECKFILE_HOOK_UPDATE_JOB_INFO = $(LIBEXEC)/hook_checkfile
```

- The STARTER_INITIAL_UPDATE_INTERVAL and UPDATE_INTERVAL configuration parameters control the time of the first run after job start, and the interval for subsequent runs, but there's no need to change these for typical use

- Call the hook in a submit description

```
+HookKeyword = "Checkfile"
```

```
+WantIOProxy = True
```

```
+CheckfileName = "job.out"
```

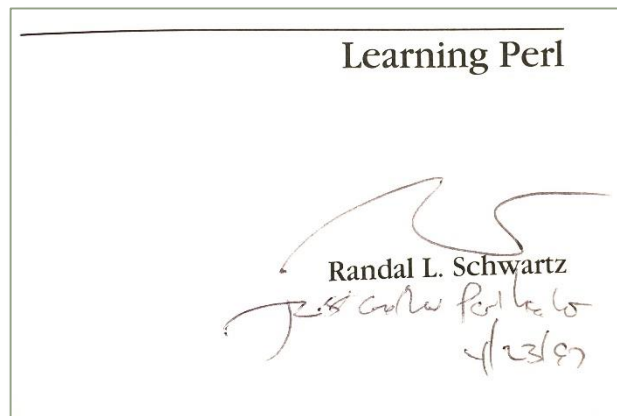
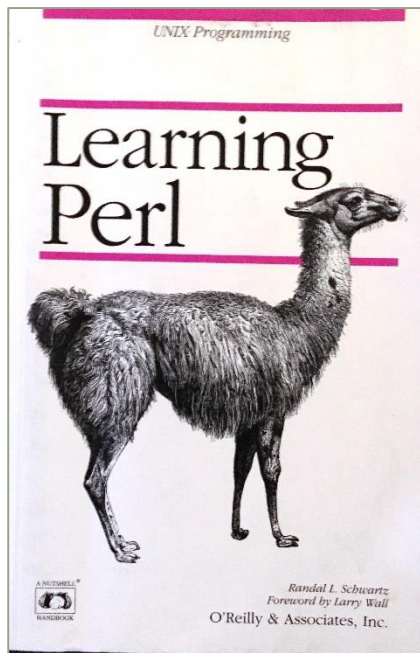
Checkfile: Step 1

- Reading the job ClassAd into a Perl hash
 - Script receives simple key-value pairs, just like *condor_q -long* output
 - All hash keys are standardized to lower case
 - No parsing is done, so ClassAd strings retain their quotes

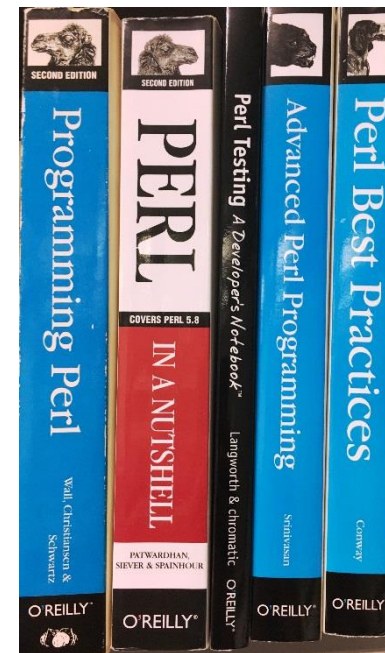
```
# Read the job ClassAd from stdin into a hash
sub read_stdin_classad() {
    my %ad = undef;
    while(<STDIN>) {
        chomp;
        my ($key, $val) = split(m{\s=\s}, $_, 2);
        $ad{lc($key)} = $val;
    }
    return %ad;
}
```

You May Be Wondering, “Why Perl?”

- As a novice USENET sysadmin at the University of Michigan, I noticed Larry Wall’s posts in comp.sources in 1988, and...



*But if you prefer
mandatory whitespace,
feel free to use
something else.*



Checkfile: Step 2

■ Canonicalize the CheckfileName attribute

- If undefined, use the job's "output" submit description file, or `_condor_stdout`
- Insure the checkfile is a full pathname, using either the scratch directory or the job's "Iwd" attribute, the "initialdir"
- `MATCH_EXP_CheckfileName` is set to the identified full path

```
sub setup_checkfile {
    my ($cf, $iwd, $out) = @_;

    # Use either _condor_stdout or Out if checkfilename is not defined
    if (length($cf) == 0 || $cf eq 'undefined') {
        $cf = "$ENV{CONDOR_SCRATCH_DIR}/_condor_stdout";
        if (! -e $cf) {
            $cf = $out;
        }
    }

    # Generate full path for checkfilename - look in scratch first,
    # then fall back to the submitter's InitialDir
    if ($cf !~ m{^/}) {
        $cf = "$ENV{CONDOR_SCRATCH_DIR}/$cf";
        if (! -e $cf) {
            $cf = "$iwd/$cf";
        }
    }
    return $cf;
}
```

Checkfile: Step 3

- Run a `stat()` on the `CheckfileName`
 - The `get_info()` function caches the result of the `stat()` call in a `BEGIN` block, to avoid making multiple `stat()` calls on the same file
 - The `+CheckfileName` which was set in the submit description is the `stat()` target
 - Items of interest are the last modification time, last access time, and file size

```
# Gather the file stats
my $mtime = get_info($ad{checkfilename}, "mtime");
my $atime = get_info($ad{checkfilename}, "atime");
my $size = get_info($ad{checkfilename}, "size");
```


Checkfile: Step 4

- Use *condor_chirp* to update the job ClassAd via `set_job_attr`
 - Be sure to avoid updating previously-set values you failed to find in this run
 - Non-delayed, so it requires the `WantIOProxy` attribute to be set to `True`
 - Since a non-delayed “`set_job_attr`” call is more labor-intensive, we don’t send an update unless the value has changed by comparing to STDIN job ClassAd
 - Delayed updates could take up to 15 minutes to be recognized

```
# Update the job ClassAd (only chirp if changed)
($mtime != $ad{checkfilelastmodifiedtime}) &&
    system($condor_chirp, "set_job_attr", "CheckfileLastModifiedTime", $mtime);
($atime != $ad{checkfilelastaccessedtime}) &&
    system($condor_chirp, "set_job_attr", "CheckfileLastAccessedTime", $atime);
($size != $ad{checkfilesize}) &&
    system($condor_chirp, "set_job_attr", "CheckfileSize", $size);
```

Implementing a Trailing Stop Order

- Set the file-activity timeout policy in the submit description

```
TIMEOUT = 10 * $(MINUTE)
LOG_AGE = ifThenElse( \
    ( ! isUndefined(CheckfileLastModifiedTime ) , \
    ( CurrentTime - CheckfileLastModifiedTime ), 0)
periodic_hold = JobStatus == 2 \
    && ( $(LOG_AGE) > $(TIMEOUT) )
periodic_hold_reason = "Log not updated for 10 minutes"
periodic_hold_subcode = 54321
```

- No hard deadlines, no risky app-induced abort timeout
 - A four-hour job hung after 10 minutes stops in 15 minutes instead of five hours
 - The six hour job can run to completion as long as it keeps generating output

Adjustments Needed for Automatic Restart

- The ClassAd and Checkfile will have old data upon restart
 - Need to adjust `periodic_hold` to avoid immediately holding restarted jobs
 - Insure the job has been running at least as long as the timeout before holding

```
TIMEOUT = 600
SUSP = ifThenElse(isUndefined(CumulativeSuspensionTime), \
                  0, CumulativeSuspensionTime )
MOD = ifThenElse(isUndefined(CheckfileLastModifiedTime), \
                 0, CurrentTime - CheckfileLastModifiedTime )
periodic_hold = JobStatus == 2 \
                && ( CurrentTime - $(SUSP) - $(MOD) >= $(TIMEOUT) ) \
                && ( CurrentTime - JobCurrentStartDate >= $(TIMEOUT) )
periodic_hold_subcode = 54321
```

Automatic Restart with Periodic Release

- Release the job if it was held due to a hang
 - The software might only hang under certain circumstances, and might run to completion if restarted.

```
MAX_TRIES = 3
periodic_release = ( HoldReasonCode == 3 \
    && HoldReasonSubcode == 54321 \
    && NumJobStarts < $(MAX_TRIES) )
```

- By matching against the subcode set for the periodic_hold, this expression will not auto-release a job that was put on hold by the user or as a result of an error
- The job will not be restarted once it reaches three attempts

So Far, So Good

So what can we check besides the file's size and timestamps?

Anything we can code!



geralt @ Pixabay – Public Domain

Flagging an Error in the File

- Set Boolean job attribute CheckfileRegexpFound

- Submit description invocation:

```
+CheckfileName = "interceptor_entity.out"
```

```
+CheckfileRegexp = "^error:"
```

```
+CheckfileRegexpModifier = "i"
```

```
+CheckfileRegexpChunkSize = 1024
```

- Translation: if the file contains a case-insensitive match to the string “error:” at the beginning of a line in the last 1024 bytes of the file, CheckfileRegexpFound is set to True.

- Take action using periodic_hold:

```
periodic_hold = ( CheckfileRegexpFound == True )
```

```
periodic_hold_reason = "Error detected in interceptor log"
```

Capture File Contents as Job Attributes

- Extract data values from the Checkfile using regexp matches

```
+CheckfileDataRegexp = "^Percent Complete: (\d+)"
```

```
+CheckfileName = "rcs_analysis.out"
```

```
+CheckfileDataAttributeName = "JobPercentComplete"
```

- Translation: look for the last line in the default chunk size beginning with “Percent Complete” and pull out the numeric value following the colon, and set the JobPercentComplete attribute accordingly.

```
description = Progress: $$ (JobPercentComplete) %
```

```
PREEMPTION_REQUIREMENTS = ( isUndefined(JobPercentComplete) \
    || JobPercentComplete < 50 )
```

Future Development

- Use string lists to allow multiple checks to be performed

```
+CheckfileName = { "interceptor.log", "c2bmc.log" }
```

**Command & Control, Battle Management, & Communications
<https://www.mda.mil/system/c2bmc.html>*

```
+CheckfileRegExp = { "^error:", "^exception:" }
```

```
CheckfileRegExpFound = False || True
```

- Translation: an exception line was found in the c2bmc.log file

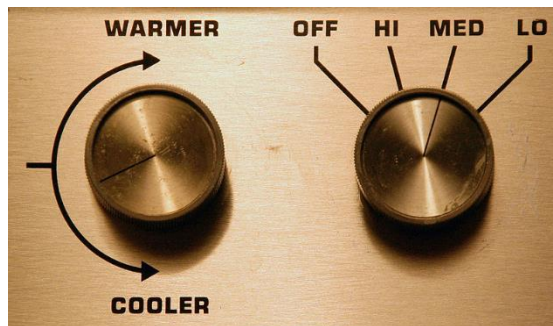
- Limited only by your mad coding skillz



Feature Request

- Non-hook periodic program invocation in submit description
 - Read the job ClassAd updates from hook standard output
 - Allow a user-specified interval with a reasonable minimum limit

“THERE OUGHTA BE A KNOB FOR THAT!”



Tysto @ Wikimedia – Public Domain

Questions and Comments

