NAME

oddmanout − Run a command on each node of a cluster and report which nodes produce anomalous output

SYNOPSIS

oddmanout [**—verbose**] **—help**

oddmanout **—version**

oddmanout [**—verbose**]... [**—param**=*string*] [**—dump**] [**—diff**[=*program*]] [**—majority**=*number*] [**—roundnums**] [**—script**=*template*] [**—nocleanup**] [**—awk**=*program*] [**—launch**=*template*] [**—run**=*command*]... **—nodes**=*number* [*filename*]...

DESCRIPTION

Homogeneity is a desirable attribute for workstation clusters used for high-performance computing. However, it can be difficult to ensure that all nodes in the cluster are *exactly* the same. Nodes may have hung processes, filesystems that failed to mount, modules that failed to load, etc. On very large clusters, there may even be nodes with different CPU speeds or amounts of memory. **oddmanout** helps find nodes that are different from the rest. The idea is to run a command (or set of commands) on every node of a cluster, find the most common output across all nodes, and report those nodes whose output is different from that.

Writing script files

In addition to using the command line, one can tell **oddmanout** what command(s) to run by using a set of script files. **oddmanout** script files are fairly simple. Blank lines and lines beginning with # are ignored. The remaining lines must be of the form "*operation*: *value*". *operation* must be one of the following (case-insensitive):

LAUNCH *value* is a command template to use to launch a script on a number of nodes. The actual command is formed from the template by replacing `NODES` with the number of nodes, `PARAM` with a given run-time parameter, and `SCRIPT` with the name of a Bourne-shell script. (These replacement terms are all case-sensitive.) If `SCRIPT` is not found in the template, it will be appended automatically.

RUN *value* is a command or command pipeline to run on the cluster. It must use Bourne-shell syntax and it must write its output to the standard output device.

`LAUNCH` and `RUN` can be specified multiple times per script file. Each `RUN` operation is performed in sequence using the most recently specified `LAUNCH` command template.

The following is an example of an **oddmanout** script file that uses **mpirun** to launch the **uptime** command on multiple nodes of a cluster. **uptime**'s output is piped through an **awk** command that outputs all fields except the actual uptime (i.e., the current time, the number of users, and the {1, 5, 15}−minute load fields). After running **uptime**, the script examines each node's */proc/cpuinfo* file to ensure that the CPUs in all nodes are identical.

```
# Example of an oddmanout script file
# By Scott Pakin <pakin@lanl.gov>

LAUNCH: mpirun -np NODES
RUN: uptime | \
       awk '{                              \
             printf "%s", $1;              \
             for (i=NF-6; i<=NF; i++) {    \
               printf " %s", $i            \
             };                            \
             printf "\n"                   \
           }'
RUN: cat /proc/cpuinfo
```

Note that a backslash can be used within the `RUN` operation to indicate that the command to execute continues on the subsequent line.

OPTIONS

−h, **—help**
Output a summary of **oddmanout**'s command-line options. When used with `--verbose`, `--help` outputs a complete Unix man page.

−v, **—verbose**
Make **oddmanout** output a description of each step it takes. Additional uses of `--verbose` further increase the amount of information that **oddmanout** outputs.

−V, **—version**
Output **oddmanout**'s version number and exit the program.

−n *number*, **—nodes**=*number*
Specify the number of nodes (i.e., unique hosts) on which to run commands. This argument is mandatory; **oddmanout** needs to know how much output to expect.

−p *number*, **—param**=*string*
Specify an additional parameter to pass to the launcher. See the description of `--launch` for more information.

−D, **—dump**
Display the baseline output for each command that is run on the cluster.

−d [*program*], **—diff**[=*program*]
Display differences between each node's output and the baseline. Differences are prefixed with one of `a` for ''added'', `d` for ''deleted'', or `c` for ''changed''. If *program* is specified, then that program is used instead of **diff**. (It must produce the same output as **diff**, however.)

−M *number*, **—majority**=*number*
Define the number or fraction of similar outputs that constitute a majority. If *number* is an integer greater than or equal to 1, then that number of nodes constitutes a majority. If *number* is a fraction between 0 and 1, then that fraction of the total number of nodes constitutes a majority. If not specified, `--majority` defaults to 2.

−R, **—roundnums**
Round all numerical values in the output to the nearest single-digit factor of a power of 10. For example, the floating-point number −876.54e+1 would be rounded to 9000. The intention is to filter out insignificant numerical differences that might show up as false positives in the output. Note that `--roundnums` rounds *every* number in the output, while may lead to some unexpected results. For example, `--roundnums` would cause the time 12:38 to be rounded to 10:40. (12 gets rounded down to 10 and 38 separately gets rounded up to 40.)

−s *template*, **—script**=*template*
Specify a filename template to use for temporary script (and other) files generated by **oddmanout** in the course of its execution. The name must end with four or more `X`s, which will be replaced by random alphanumerics whenever a filename is generated. The default template is `./omocmd-XXXXXX`.

—nocleanup
Keep temporary files around after **oddmanout** terminates. Normally, the program deletes temporary files before exiting.

−a *program*, **—awk**=*program*
Specify a program to use instead of **awk** for postprocessing output. *program* must behave exactly like **awk**, though.

−l *template*, **—launch**=*template*
Specify a template for a command that should be used to run a script on the cluster. Any occurences of `NODES` will be replaced with the number of nodes; any occurences of `PARAM` will be replaced with a given run-time parameter; and, any occurences of `SCRIPT` will be replaced with the name of a Bourne-shell script. (These replacement terms are all case-sensitive.) If `SCRIPT` is not found in the template, it will be appended automatically.

−r *command*, **—run**=*command*

Specify a command to run on the cluster. `--run` can be specified multiple times on the command line, enabling a number of commands to run in sequence.

In addition to the preceding options, **oddmanout** accepts the names of zero or more files containing commands for it to execute. See the section on "Writing script files" for more information. The order of operation is the `--launch` option, the `--run` options, and then the specified script files. If none of those are used, **oddmanout** outputs a usage message.

DIAGNOSTICS

Warning messages regarding script files are preceded with the script-file name and line number in which the error occurred.

missing command

A line in the script file contains something other than a comment or a line of the form "*operation*: *value*".

file ended prematurely

A `RUN` command ended with \, indicating a continuation line, but the script file ended before one was found.

we need a LAUNCH command before we can RUN

At least one `LAUNCH` command (or `--launch` command-line option) must be given before a program can be run on the cluster.

ignoring unrecognized command

A script file contained a command not listed in the section on "Writing script files".

EXAMPLES

Ensure that all nodes mount exactly the same filesystems, using **prun** to launch a **mount** command on 256 nodes and specifying that */usr/bin/gawk* be used internally as the **awk** command:

```
oddmanout --nodes=256 --launch="prun -N NODES" --run=mount
  --dump --diff --awk=/usr/bin/gawk
```

Using **pdsh** as the launcher, see if any spurious process are still running somewhere on a 512−node cluster in which machines are named `foobar0`, `foobar1`, ..., `foobar511`:

```
oddmanout --diff --launch='pdsh -w foobar[0-'expr NODES - 1']'
  --run="ps -eo user,command | sort" --nodes=512
```

Note that in the preceding example, we used `--diff` but omitted `--dump` because we know that there are a lot of processes running on every node. We really don't need to see the complete list.

CAVEATS

The following are some caveats of which the **oddmanout** user must be wary:

Shell expansions

Commands specified on the command line using `--run` are shell-expanded *twice*: once by the shell before being passed to **oddmanout** and once by the shell invoked by the launcher. As a result, commands that themselves contain quotes may not be expressible via the command line. Consider, for example, the following `RUN` command which verifies (on Linux) that all nodes report the same amount of total memory:

```
RUN: cat /proc/meminfo | awk '/Mem:/ {print $1,$2}'
```

That command cannot be expressed as follows on the command line:

```
--run="cat /proc/meminfo | awk '/Mem:/ {print $1,$2}'"
```

The problem is that the first shell does not honor the single quotes and therefore expands `$1` and `$2`. Swapping the single and double quotes only shifts the problem to the second shell. In some shells (e.g.,

`bash`, but not `tcsh`) the dollar signs can be backslashed to achieve the desired purpose:

```
--run="cat /proc/meminfo | awk '/Mem:/ {print \$1,\$2}'"
```

Optional arguments

The `--diff` option takes an optional argument. Because of the way that **perl**'s *Getopt::Long* module parses arguments, the optional argument can be separated from `--diff` by a space instead of an equals sign. The result is that `--diff` may unexpectedly "swallow" a filename that immediately follows:

```
oddmanout --nodes=32 --diff commands.omo
```

The following possibilities work around the problem:

```
oddmanout --nodes=32 --diff -- commands.omo

oddmanout --diff --nodes=32 commands.omo
```

In the former, `--` specifies that no more options follow. In the latter, `--diff` is moved earlier in the command line so that it sees `--nodes=32`, which it knows is a new command-line option.

Interpreting differences

When **oddmanout** reports differences between one node's output and the majority output, a few things may be unclear. First, because **oddmanout** discards line numbers you may see differences like the following:

```
Host abc123's differences from the baseline output:
  D: foo bar baz quux
  A: foo bar baz quux
```

Saying that a line was both deleted (`D`) and added (`A`) implies that, in fact, the line merely appeared at a different location in the output.

Second, although **oddmanout** ignores differences in lines for which there is no majority line, when the program finds a node that differs from the majority, it outputs *all* differences, including those for which there is no majority. For example, running "`cat /proc/cpuinfo`" on every node of an Alpha/Linux cluster will output the `cpu serial number` for each node that has *some* output which is different from the majority's, even though there's obviously no majority serial number (as each serial number is guaranteed to be unique). Similarly, a `ps` command may show an **oddmanout**−introduced **awk** command which accepts a node-specific `HOSTNAME` variable on its command line and therefore differs from node to node.

LEGAL NOTICE

SEE ALSO

awk (1), *diff* (1), *prun* (1), *mpirun* (1), *pdsh* (1)

AUTHOR

Scott Pakin, *pakin@lanl.gov*