



Troubleshooting



New Zealand eScience Infrastructure



We will cover

1. Examples of some common errors on NeSI.
2. Troubleshooting Methodology.
3. How to get the best help when you do get stuck.

Define 'Troubleshooting'

A logical, systematic search for the source of a problem in order to solve it, and make the product or process operational again.

Applications Support



Fig 1. Important People



Common NeSI Errors

Common Errors

Resource Errors

<i>TIMEOUT</i> <i>CANCELLED DUE TO TIME LIMIT</i>	Not enough walltime
<i>oom-kill</i> <i>OOM</i> <i>OUT_OF_MEMO+</i> <i>Out of Memory</i>	Not enough memory.
<i>Segmentation Fault</i> <i>SIGSEGV</i> <i>SEGFAULT</i> <i>Creates .core files.</i>	Code is doing something inappropriate with memory.
<i>SIGBUS</i> <i>Bus error</i>	Code is referring to nonexistent memory (bad memory address).

Common Errors

Input Errors

<i>No input file specified</i>	Path error.
^M <i>bad interpreter</i> \r\n	Batch script contains DOS line breaks (\r\n) <code>dos2unix <file></code>
<i>Permission Denied</i>	Wrong file/directory permissions. <code>chmod u+rwX <file></code>
The input works on my local machine!	Validate with checksums. Develop a simple input as a test. What is broken: the software, your specific input file, or both?

Common Errors

Queue Errors

My job doesn't start!	Check <code>squeue -u <username></code> under 'reason'
<i>SBATCH</i> error when submitting	Trying to run 2 debug jobs? Asking for more time/memory/cpus than a node has? Using an expired project? Typos?
There are no outputs!	Check --output in slurm script. Room in filesystem? <code>nn_check_storage_quota</code> Permissions Did the job start?

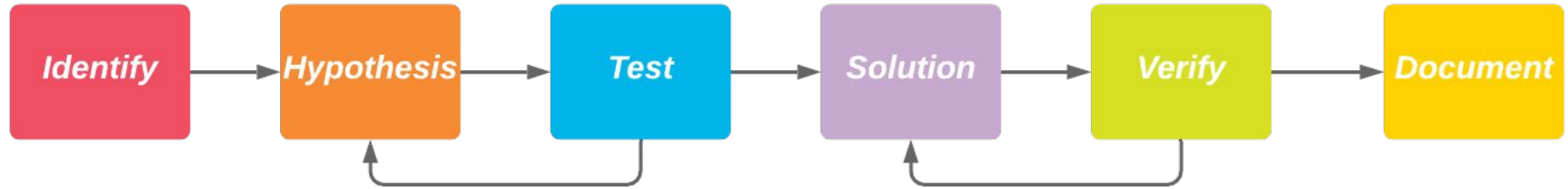
Common Errors

Program Errors

<i>The following module(s) are unknown: "SOFTWARE"</i>	Error with module load
I loaded the module and nothing happened!	Loading module doesn't do anything by itself.
<i>command not found.</i>	Not loaded modules.
It's not working the same as on my desktop.	Try a different version. Confirm with same input.



Methodology



Troubleshooting is an *iterative* process, often using a *process of elimination*.

Identify

What is not working as it should be?

- Aim is to identify *root cause* failure. As opposed to downstream errors (errors caused by errors)
- Information gathering.
 - SLURM
 - Output logs

On NeSI - SLURM

`sacct` - provides info on all of your jobs submitted/finished that day.

JobID	JobName	Timelimit	Submit		Start		Elapsed	TotalCPU	AI	NT	ReqMem	MaxRSS	State	NodeList	Partition
10027373	problem_1	00:30:00	Jan 21 15:13	Jan 21 15:14	Jan 21 15:14	15:14	00:07:24	00:02.629	6		512Mc		OUT_OF_MEMO+	wbn094	large
10027373.ba+	batch		Jan 21 15:14	Jan 21 15:14	Jan 21 15:14	15:14	00:07:24	00:02.628	6	1	512Mc	3000768K	OUT_OF_MEMO+	wbn094	
10027373.ex+	extern		Jan 21 15:14	Jan 21 15:14	Jan 21 15:14	15:14	00:07:24	00:00:00	6	1	512Mc	0	COMPLETED	wbn094	

`squeue -u username` - provides info on all of your PENDING, RUNNING jobs.

JOBID	PARTITION	PRIORITY	NAME	USER	ST	START_TIME	NOD	CPUS	FEATURES	REASON	TIME_LEFT
10074260	large	0	problem_2	cwal219	PD		N/A	2 16	(null)	JobHeldUser	15:00

`nn_seff jobid` - provides info on a FINISHED job.

```
Job ID: 10078413
Cluster: mahuika
User/Group: cwal219/cwal219
State: TIMEOUT (exit code 0)
Cores: 16
Tasks: 1
Nodes: 16
Job Wall-time: 100.07% 10:00:25 of 10:00:00 time limit
CPU Efficiency: 12.49% 19:59:32 of 6-16:06:40 core-walltime
Mem Efficiency: 0.28% 135.82 MB of 46.88 GB
```

Identify

What is not working as it should be?

On NeSI - Output Logs

Logs for *stdout* and *stderr* are specified in the SLURM header

```
#SBATCH --output=<path>
```

```
#SBATCH --error=<path>
```

If no *stdout* is specified it will go into a file named `slurm-<jobid>.out`.

If no *stderr* specified it will print to the same file as *stdout* (recommended).

The paths can be *relative* or *absolute*.

If output paths are invalid, the **job will run but your outputs will be lost**.

Many applications make their own output. *stdout* is just the output that would be printed to terminal if running interactively.

Don't delete your error logs >:(

Identify

What is not working as it should be?

On NeSI - Output Logs

- Step back through log, from failure until you reach the cause.
- Stack trace / traceback
- May need to run again with a verbose (`-v`) flag or extra print statements.
- `tail` and `grep`

 **Warning:**

Something has gone wrong but is recoverable.

 **Error:**

Something has gone wrong fatally.

Hypothesis

What is your best guess?

- What has changed since it last worked?
- *Should* something have changed?
- External / Internal?
- If the problem is intermittent, is there a pattern?
- Look for common factors.
- Ask a friend/Google.

Question the obvious and avoid making assumptions.



Fig 2. Bus error

Test

Isolate variable.

Confirm/reject hypothesis.

- Reproduce the problem in a controlled environment.
- Control independent variables.
- Are there variables that couldn't be controlled?
- Positive or Negative control.
- Consider there *may be multiple factors*.

Design tests in a way that can be repeated quickly. You may have to do this step multiple times.

- Run interactively (login node)
- Short walltime - Low memory - Few CPUs
- `--qos=debug` or `--partition=prepost`

Test

Isolate variable.

Confirm/reject hypothesis.

The test disproves your hypothesis...

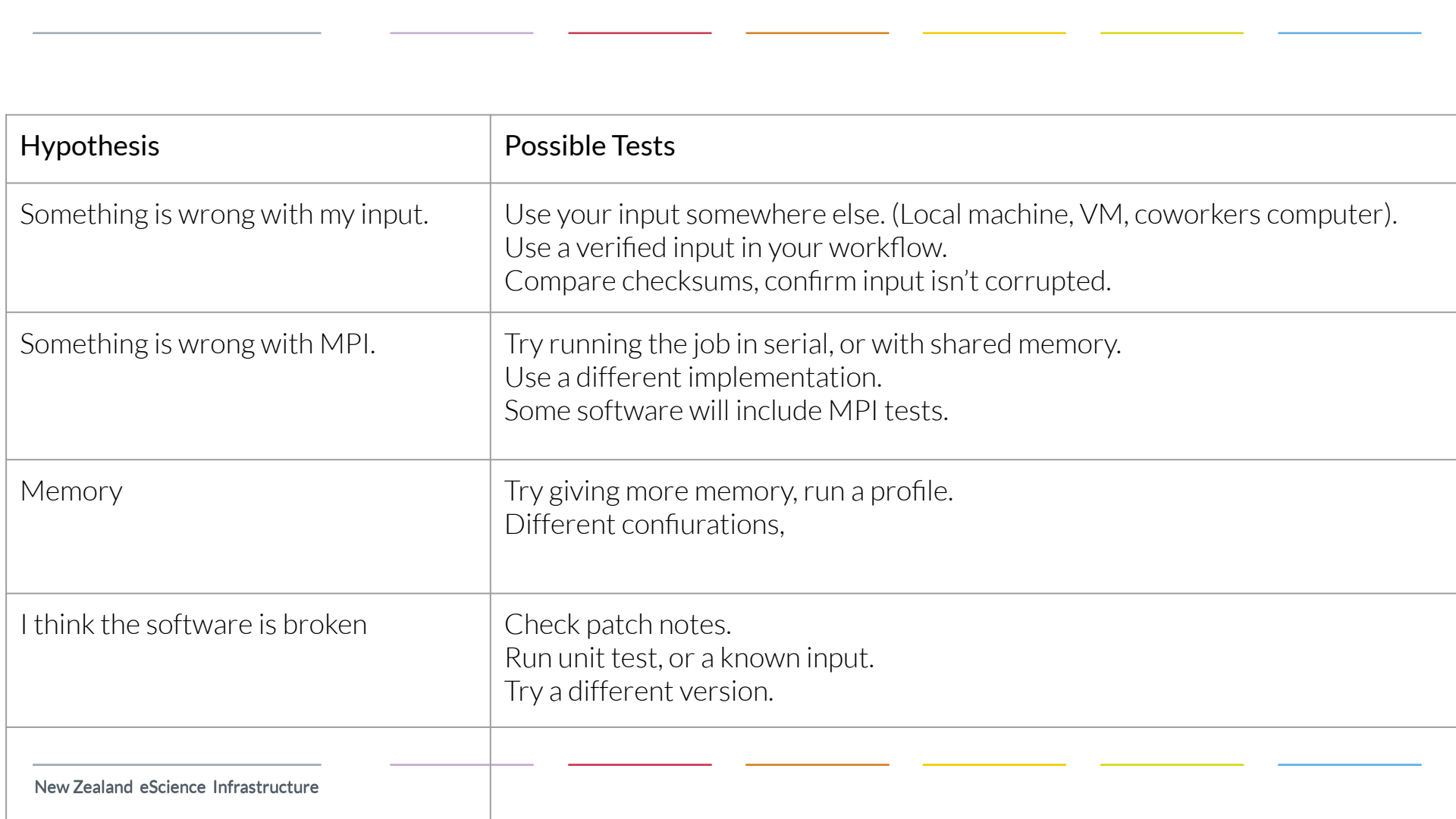
- Cross it off the list and come up with a new hypothesis.

The test confirms your hypothesis...

- Celebrate! Then move on to developing a solution.

It was inconclusive....

- Bad test. Try again.
-



Hypothesis	Possible Tests
Something is wrong with my input.	Use your input somewhere else. (Local machine, VM, coworkers computer). Use a verified input in your workflow. Compare checksums, confirm input isn't corrupted.
Something is wrong with MPI.	Try running the job in serial, or with shared memory. Use a different implementation. Some software will include MPI tests.
Memory	Try giving more memory, run a profile. Different configurations,
I think the software is broken	Check patch notes. Run unit test, or a known input. Try a different version.
New Zealand eScience Infrastructure	

Solution

Plan and implement solution.

- Check if a fix already exists.
 - Patch / Updates
 - May have to wait on developer.
 - Workaround / Hack
 - Keep a copy of the pre-solution version.
-

Verify

Confirm fix.

Don't assume the problem is fixed because your solution worked in a test environment!

- Can you reproduce the original error?
- Verify fixed component. (Unit testing)
- Verify fixed component fits into whole. (Integration testing)
- Verify whole (System testing)

If the problem persists...

- Go back and try another solution.

If you run into a *new* problem...

- Identify your new problem.
 - Did you correctly assess the root error or are there multiple independent problems?
 - Did the job get further than last time, or did it fail at the same point (or earlier)?
 - Don't be disheartened, getting a new error is progress!

Everything works!

- Go back and try another solution.

Document

Record your process.

- Useful for future you if the problem happens again.
- People will like you.
- If the software is open-source, consider submitting a patch to the developers.



Getting Help

Getting Help

If you are stuck, there is no shame in asking for help!

<http://support.nesi.org.nz>

Before you do though:

- Make some effort solving issue yourself.
- Check the NeSI support documentation to see if your issue has already been solved.
- Google it.
- Ask a friend/co-worker.
- Have you attended our introductory workshop?
- ~~The error log doesn't explicitly tell you how to fix the problem.~~

How to Google

- Don't just copy and paste the entire text of your error message, especially if it includes context-specific details like usernames and file paths.
- Include software name.
- Use the asterix (*) wild card, especially in place of context-specific text.
- Use “quotes” when pasting error messages into Google search to search for specific phrases.
- Quality resources
 - StackOverflow.
 - Github
 - Developers documentation / community forums.
 - NeSI documentation.

Bad Google:

[error] Project:Sim1, Design:HFSSDesign1-Initial (DrivenTerminal), Failed to check out license 'hfss_solve'

Better Google:

ANSYS EM “Failed to check out license 'hfss_solve'”

Google

what is a bus



All

Images

Maps

News

Videos

More

Settings

Tools



electric buses en route to Auckland ...
ourauckland.aucklandcouncil.govt.nz



Auckland bus drivers vote to continue ...
staff.co.nz



bus routes for Whau and central suburbs ...
ourauckland.aucklandcouncil.govt.nz



NeSI
New Zealand eScience
Infrastructure

Fin