



Embedded Nimrod

Straightforward HTC in HPC environments

October 22, 2019

Zane van Iperen David Green
Hoang Nguyen David Abramson

Research Computing Centre
University of Queensland

High Throughput Computing (HTC)

- HTC – High Throughput Computing
 - Large quantities, small-footprint, loosely-coupled
- HPC – High Performance Computing
 - Longer walltimes, tightly-coupled (MPI), etc.
- Significant overlap
- Classic HTC Example: Parameter Sweeps
 - Single task
 - Run many times with different parameter combinations

HTC on HPC

- Swarms of jobs overloading the scheduler (PBSPro)
- Were mostly HTC-style jobs
- Job Arrays weren't enough.
- Run-on effects
 - Resource fragmentation
- Scheduling and setup overhead can be longer than the job itself

Embedded Nimrod

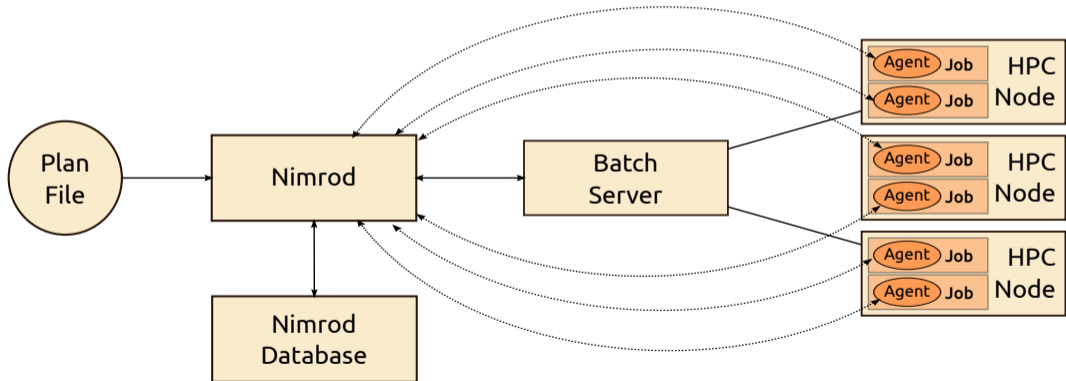
- Leverages the parameter sweep and execution engine of Nimrod/G,
- Bridges the gap between HTC and HPC,
- Can run millions of jobs in one fell swoop,
- Optimises job placement,
- Has a minimal learning curve.

Nimrod/G

- Nimrod – distributing computing toolkit.
- Nimrod/G – the “grid” scheduler.
- Can dispatch work over resources such as HPC clusters and cloud infrastructure.
- Uses agent-based execution model.



Infrastructure – Traditional Nimrod/G

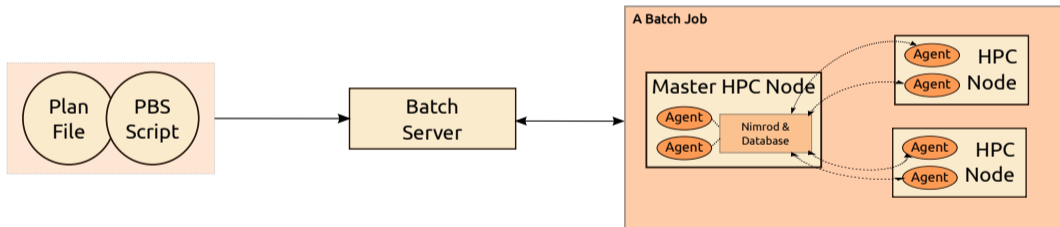


Infrastructure – Traditional Nimrod/G

Problems:

- Significant amount of setup and configuration required:
 - Database (PostgreSQL/SQLite3)
 - Message Queue (RabbitMQ/Apache Qpid)
 - The cluster itself
- Need to convert the job script to a Nimrod Planfile
- Not the best use of Researchers' time

Infrastructure – Embedded Nimrod



Infrastructure – Embedded Nimrod

Pros:

- Handles all setup and configuration behind-the-scenes
- Is (almost) a drop-in replacement for job arrays.

Cons:

- Assumes that nodes have a shared filesystem
 - Assumes the submission directory is writable from all nodes
- A large chunk of resources may take time to become available

An example job script (PBSPro)

```
#!/usr/bin/env nixec  
#PBS -lselect=4:ncpus=4:ompthreads=2:mem=16gb  
#PBS -lwalltime=10:00:00  
  
#NIM shebang /bin/bash  
#NIM parameter x integer range from 1 to 100 step 1  
#NIM parameter y integer range from 1 to 100 step 1  
  
expr ${NIMROD_VAR_x} \* ${NIMROD_VAR_y}
```

An example job script (PBSPro)

```
#!/usr/bin/env nixec
#PBS -lselect=4:ncpus=4:ompthreads=2:mem=16gb
#PBS -lwalltime=10:00:00

#NIM shebang /usr/bin/env python3
#NIM parameter x integer range from 1 to 100 step 1
#NIM parameter y integer range from 1 to 100 step 1

import os
x = int(os.getenv('NIMROD_VAR_x'))
y = int(os.getenv('NIMROD_VAR_y'))
print(x * y)
```

Recap

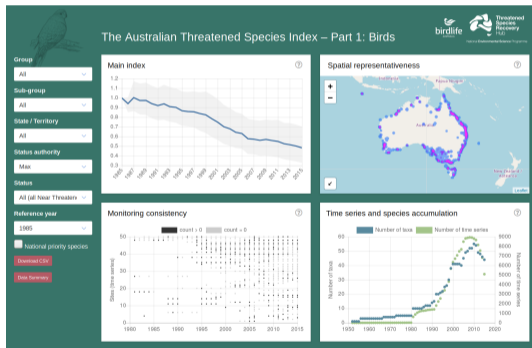
- #NIM shebang defines the script interpreter.
 - Can be /bin/bash, /usr/bin/python, etc.
 - Defaults to /bin/sh if not specified
- #NIM parameter defines the job parameters.
- Parameter values are passed via NIMROD_VAR_ environment variables.
- $nAgents = select \times \frac{ncpus}{ompthreads}$ (PBSPro)

Use Cases

- Threatened Species Index
- Inland Drayage Research

Use Case: Threatened Species Index

- National index of threatened bird species
- Interactive data explorer
- 60 data sources and counting
- 2018 – Bird, 2019 – Mammals, 2020 – Plants



(<https://tsx.org.au/tsx/>)

Use Case: Threatened Species Index

6 parameters, ~40,000 combinations:

- Group – Terrestrial, Wetland, Marine, etc.
- Subgroup – Grassland, Rainforest, etc.
- State/Territory – QLD, NSW, etc.
- Status authority – BirdLife Australia, EPBC, ICUN
- Status – Vulnerable, Endangered
- Reference Year

6 hours, with 32 cores on Tinaroo (Intel Xeon E5-2680 v3)

Use Case: Inland Drayage Research

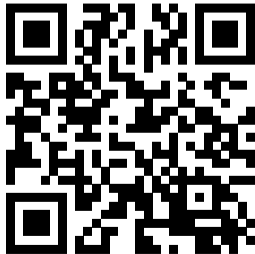
- Optimising cargo transport routes at the Port of Brisbane.
 - Time & separation modes,
 - fleet composition & truck size,
 - coupling & precedence principles.

Use Case: Inland Drayage Research

- 600 jobs
- Walltimes from seconds to hours
- One parameter – the file name
- ~2 days total walltime, down from a week
- 48 cores total, 6 nodes, 3 agents/node, 8 cores/process

What's next?

- <https://github.com/UQ-RCC/nimrod-embedded>
- Free Software: Apache 2.0 License
- Runs on:
 - Tinaroo, Awoonga, Flashlite (UQ, PBSPro)
 - Wiener (UQ, SLURM)
 - NordIII (BSC, Spectrum LSF)






THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA


CREATE CHANGE

Thank you

Zane van Iperen
Research Computing Centre
z.vaniperen@uq.edu.au

 facebook.com/uniofqld

 instagram.com/uniofqld

 twitter.com/rccuq

CRICOS code 00025B