SAND2022-6742C

Sandia National Laboratories

Resource extension using commercial cloud techniques at Sandia

Paul Wolfenbarger

Current capability and future hopes

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

High Performance Application Testing Requirements

One Node

- This confirms that a given toolset and code change
  - Compiles / links
  - Runs and outputs correct answers
  - Uses SHM transport and low parallelization levels

Small scale (2-4 nodes)

- This proves that MPI or other communication mechanism functions
  - Tests for effects of non-SHM transport
  - Larger test cases

Full scale (N nodes)

- Mostly triggered by hand
- Proves scaling effects match expectations

High Performance Application Testing Requirements

One Node

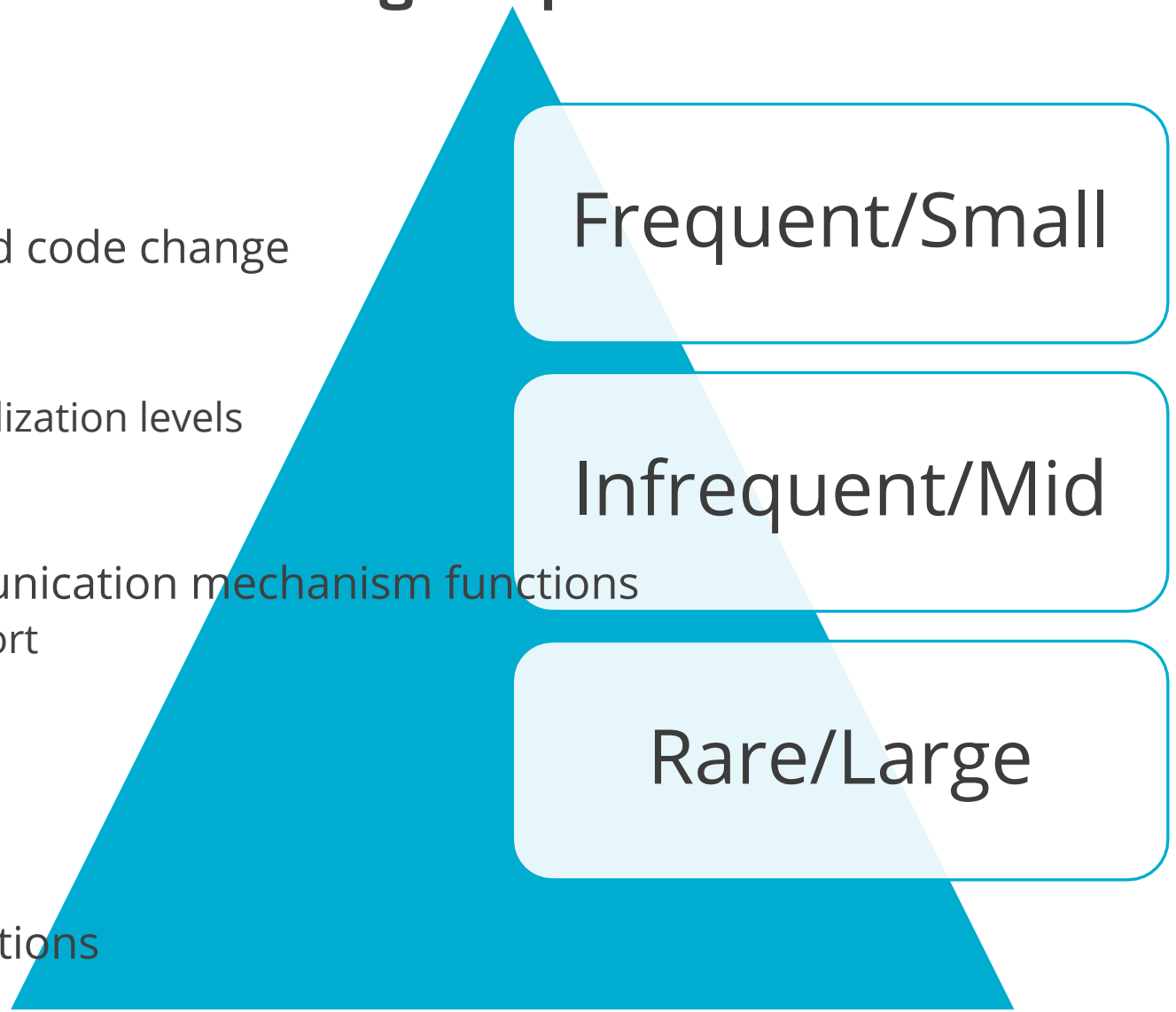
- This confirms that a given toolset and code change
  - Compiles / links
  - Runs and outputs correct answers
  - Uses SHM transport and low parallelization levels

Small scale (2-4 nodes)

- This proves that MPI or other communication mechanism functions
  - Tests for effects of non-SHM transport
  - Larger test cases

Full scale (N nodes)

- Mostly triggered by hand
- Proves scaling effects match expectations



High Performance Application Testing Requirements

One Node

- This confirms that a given toolset and code change
  - Compiles / links
  - Runs and outputs correct answers
  - Uses SHM transport and low parallelization levels

Small scale (2-4 nodes)

- This proves that MPI or other communication mechanism functions
  - Tests for effects of non-SHM transport
  - Larger test cases

Full scale (N nodes)

- Mostly triggered by hand
- Proves scaling effects match expectations

Most tests

Some tests

Few tests

Available Hardware

Queued HPC systems

- Maintained by corporate – matches production environment
- Testing competes with Analysts
- Programming Environments from
  - Vendor
  - Admins
  - SEMS
  - CDE

Dedicated test machines

- High purchase and maintenance costs
- Frequently the best way to get early or dedicated access to advanced architectures
- No load leveling
- Programming Environments from
  - DIY
  - SEMS
  - CDE

Cloud as an alternative

Cloud systems

- Load leveling handled by administrators
- Designed for web services (lighter than most single nodes)
- Configurable via scripting
- Cinder (disk) space is limited
- Network setup is significant, but a one-time cost.
- Programming Environments are all DIY

Starting: CEE services through NILE

- https://wiki.sandia.gov/display/CEE9/Cloud+OpenStack

Tenant Types

- DevOps Tenant with sudo Access:
- Enclave Tenant:
  - Designed for Mission support.
  - Cross Domain Solution limits SRN access.
  - Allows proxy access.
  - Allows customer base image uploads.
  - Allows sudo access.

Network setup

- https://wiki.sandia.gov/display/CEE9/CEE+Cloud+Network+Setup

Simplified Sample init script

```
#!/bin/bash

mkdir /scratch

# set up gitlab-runner

curl -L "https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.rpm.sh" | sudo bash

export GITLAB_RUNNER_DISABLE_SKEL=true

yum -y install gitlab-runner

# register and start a gitlab-runner

gitlab-runner register \
        --non-interactive \
        -u https://cee-gitlab.sandia.gov \
        -r d6icnDdd3R9SVyByEBxz \
        --tag-list "cee-cloud-16,cee-cloud-32,cee-cloud-4,cee-cloud-8" \
        --executor "shell" \
        --shell-user "cth-entity" \
        --shell-host "localhost"

gitlab-runner start
```

How to get a PE (Programming Environment)

Copy from CEELAN

- By hand via rsync - painful
- Scripted via crontab – breaks when an instance ip address changes
- Scripted via Jenkins – same as crontabs but easy to manually launch

Install using “yum” at initialization

- All script commands run as root
- Many packages are very old
  - As an example docker from yum is unusable

Build a docker with PE already installed

- Resolves the PE problem altogether
- Newer version of docker required
- Certificate and proxy settings required
- I count 17 steps to get docker functional and 5 of those are currently not automated.
  - Addition of yum channel for newer docker (or manual retrieval)
  - Retrieval/install of sandia_root_ca.cer

Things do not remain static

The cost and flexibility make the ceecloud a useful addition, but the difficulties of network setup, PE synchronization, and docker use are significant.

The openstack system is changing

- Addition of a CEE Tenant
  - CEE LAN access to home and project space in virtual systems.
  - CEE SRN Account(s) required.
  - No sudo access.

CEE is working with openshift and rhel 8, which (if available) will

- Bring in podman/newer docker
- Allow instances to be launched on a per-job basis
- Add additional capacity