CollectI Intro and Demo

Abhinav Thota
SciAPT
Indiana University

RESEARCH TECHNOLOGIES
INDIANA UNIVERSITY
University Information Technology Services

PERVASIVE TECHNOLOGY INSTITUTE
INDIANA UNIVERSITY

What is Collectl?

- Collectl is a systems monitoring tool
- It can monitor a broad set of subsytems including:
  - buddyinfo, cpu, disk, inodes, infiniband, lustre, memory, network, nfs, processes, quadrics, slabs, sockets and tcp.
- The collected data can be stored in compressed or uncompressed data files
  - which themselves can be in either raw format or in a space-delineated format that enables plotting using gnuplot or Microsoft Excel.
- A KB article is available: https://kb.iu.edu/d/bedc

Indirect monitoring of applications

- This sounds like a monitoring tool for system admins
- But, we are using this to monitor some of the subsystems to indirectly monitor/evaluate our application (s) running on the system
  - Instead of profiling or instrumenting the code
- By looking at memory consumption on a compute node, we can tell what the application is consuming
- By looking at CPU utilization, we can again tell what the application is doing
- This can be useful, even if multiple users are using a node
  - We can filter processes by who owns them, for example

On Karst -

- module load gnuplot
  - This is so that we can generate plots at the end of a collectl run
- module load collectl
- Feel free to run "collectl" to get a feel for how this works

Commonly used options

- -Z would be the most obvious option
  - Includes CPU, memory and IO data

Option	Description
`-F`	Flushes the output buffers at the specified interval (in seconds); `-F0` (zero) causes a flush at each sample interval
`-i`	Indicates how frequently a sample occurs (in seconds) The value preceding the colon (`:`) is the primary sampling interval. The value following the colon is the rate at which subsystem, subprocess (`-sZ`), and slab (`-sY`) data is collected, and must be a multiple of the value before the colon.
`-s`	Determines what subsystem data (summary or detail) is collected; the default is `cdn`, which stands for CPU (`c`), disk (`d`), and network (`n`) summary data; lowercase characters indicate summary data collection, uppercase characters indicate detail data collection; to get both, include lowercase and uppercase letters (e.g., `-s Zl`). Options for summary and detailed subsystem data include: **Summary subsystems** `b` - memory fragmentation `c` - CPU `d` - disk `j` - interrupts `l` - Lustre `m` - memory `n` - networks `s` - sockets `y` - slabs (system object caches) **Detail subsystems** `C` - CPU `D` - disk `J` - interrupts `L` - Lustre `M` - memory node data (or numa data) `N` - networks `Y` - slabs (system object caches) `Z` - processes **Note:** In the above example script, `-s Zl` directs `collectl` to collect detail data for processes (`Z`) and summary data for Lustre I/O (`l`).
`--procfilt`	Tells `collectl` to get data only for the processes that are specified by the filter parameters In the above example, the `--procfilt` option indicates that only processes for the `$UID` user ID (`u$UID`) should be monitored. Other filters and options for specifying what data to monitor and how to record them include `--diskfilt`, `--memopts`, `--lustopts`, and `--procopts`.
`-f`	Sets the directory for `collectl` output For jobs on Big Red II, Karst, or Mason, set `COLLECTLDIRECTORY` to a temporary directory in your Data Capacitor II scratch space (e.g., `/N/dc2/scratch/username/temp/` ; replace `username` with your IU Network ID username).

As a background process in a batch job

```
module load collectl
cd /N/dc2/scratch/username/temp
SAMPLEINTERVAL=10
COLLECTLDIRECTORY=/N/dc2/scratch/username/temp/
collectl -F1 -i$SAMPLEINTERVAL:$SAMPLEINTERVAL -sZ --procfilt
u$UID -f $COLLECTLDIRECTORY &
./my_binary
collectl_stop.sh
```

- You should adjust SAMPLEINTERVAL according to the expected runtime for your application. If your job will run for less than a day, UITS recommends setting SAMPLEINTERVAL=10; if your job will run for multiple days, UITS recommends setting SAMPLEINTERVAL=30 or SAMPLEINTERVAL=60.

CAFÉ

- CAFÉ is the application we will use today to demonstrate Collectl and other tools
- Used to analyze changes in gene family size and provide a statistical foundation for evolutionary inferences.
  - https://hahnlab.github.io/CAFE/
- Please copy this directory to your $HOME:
  - /N/dc2/scratch/athota/incoming/cafe
  - Unzip v4.0.zip and run "make" to build the binary
  - The binary will be built in the CAFE-4.0/release directory
  - From the top level directory, to run cafe:
    - CAFE-4.0/release/cafe caferun.sh

Karst Desktop Beta

- Linux remote desktop service based on VNC
- We will be using this service for the hands-on part of the presentation
- The KB article: https://kb.iu.edu/d/bfwp
- TL;DL:
  - https://www.cendio.com/thinlinc/download
  - Server: test-desktop.karst.uits.iu.edu
  - Password: your IU CAS password
  - Jobs can be submitted to Karst from the desktop nodes

Running CAFÉ with Collectl

```
#!/bin/bash
cd /N/u/athota/Karst/perf-tuning/cafe
collectl -F1 -i5:5 -sZ --procfilt u1224111 -f collectl_log/ &
CAFE-4.0/release/cafe caferun.sh
killall collectl
sleep 10
killall perl
sleep 10
```

- This writes a log file to the collectl_log directory in this format:
  - hostname-YYYYMMDD-HHMMSS.raw.gz

Plotting the data

- Can be done with the collectl_plot.sh script that is included on IU systems
- From the collectl_plot directory, run:
  - collectl_plot.sh hostname-YYYYMMDD-HHMMSS.raw.gz .
  - Don't forget the "dot" at the end :)
- There is a lot more data that we are not plotting in the log file
  - Can adjust gnuplot scripts to change the plots