


PROFILING WITH INTEL VTUNE

Useful Tips

August 9, 2023 | Dr. Martin Errenst

- Useful for defining profiles in a script
- `vtune -help` is your friend!
 - E.g. `vtune -help collect`
- `vtune -collect hotspot -knob <option>=<value> <application>` to configure collection
 - `vtune -help collect hotspots` to find list of available options
 - Similar for other analysis types

Configure Analysis 

WHERE

 Local Host 

WHAT

 Launch Application 

Specify and configure your analysis target: an application or a script to execute.

Application:**Application parameters:** Use application directory as**Advanced**

HOW

 Performance Snapshot 

Get a quick snapshot of your application performance and identify next steps for deeper analysis.
[Learn More](#)

Copy Command Line to Clipboard**Command line:**

```
/beegfs/Tools/easybuild/stacks/rome/2021a/software/VTune/2021.9.0/vtune/2021.9.0/bin64/vtune -collect performance-snapshot -app-working-dir /tmp/errenst_workdir/smalltests/build/src --app-working-dir=/tmp/errenst_workdir/smalltests/build/src -- /tmp/errenst_workdir/smalltests/build/src/vectoradd simd
```

Copy

Close

- Consider and minimize “non-deterministic” effects
 - Store results in `/tmp`, if you only have a remote filesystem
 - Consider other influences (e.g. network, hdd vs. ssd)
 - Remember statistical nature of VTunes (sampling) results
 - ⇒ Don't take resulting numbers too literally
 - Separate active processes competing for the same resources

- Profile results grow large for long running test cases
 - Minimize examples when iterating on problem
 - Collect only required data (do you need call stacks?)
 - Limit data collection to relevant code sections

To steer data collection

- Consider using the Intel ITT API
- Steer the data collection in your code
- Provides much more than just resume and pause
- Can group and mark separate code sections

```
#include <ittnotify.h>

int main(int argc, char* argv[]){
    // Boring initialization

    __itt_resume();
    // relevant code section
    __itt_pause();

    // Boring cleanup

    return 0;
}
```

- github.com/jrfonseca/gprof2dot

- Script to extract call graph

- Works with various profiler formats

- Export vtune result with

```
vtune -report gprof-cc -result-dir <path> -format text  
-report-output output.txt
```

- Then create dotfile and graph

- VTune can import Linux Perf Traces
- Collected with `perf record`
- To analyze existing measurements or when data collection with VTune is not possible

- github.com/brendangregg/FlameGraph
 - Create flame graph from various profiler results
 - **But:** VTune ships its own flame graph visualization in recent versions!

- [Intel VTune Profiler User Guide](#)
- [Intel VTune Profiler Performance Analysis Cookbook](#)
- [Tutorial: Analyze Common Performance Bottlenecks](#)
- [Tutorial: Analyzing an OpenMP* and MPI Application](#)
- [Intel 64 and IA-32 Architectures Optimization Reference Manual](#)
- [Intel 64 and IA-32 Architectures Software Developer Manuals](#)