**Requirements**

* objective: run a benchmark while collecting T, power, fan speed, TEC current, TEC voltage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | P [W] | T [C] | Measurement3 | ... |
| Δt |  |  |  |  |

* controlled variables: DVFS, fan speed/pwm, executed program, cores, runtime, sampling rate, print interval
* settings: # cores, # multimeters, # power supplies, output file, settings file location, data to collect,
* criteria
  + software: extensible (more data sources, new hardware – cpu, instruments), modular, error reporting, testable + debug, documented + source control, secure, safety features, well-designed external interface (MATLAB, bundled executable), portable (other Linux systems), automated (parse setup files, generate reports, compute statistics)
  + functionality: accurate (timing, data collection), high performance (speed, space), batch/bulk experiments, low overhead (non-intrusive)

**Detailed Information**

* motivation
  + lightweight; portable + supported
  + kernel issues + lack of driver knowledge: patching linux 3.0 with perfmon fails (perfmon is kernel version specific); updating kernel 2.6.28 with 3.0 coretemp.c fails with compiler errors; updating 2.6.28 with module compiled against either 2.6.28 or 3.0 fails; building kernel-headers with updates to just msr-index.h and processor-flags.h fails
* tools: perl, libpfm, Ryan's code, alarms, c, libraries, lm-sensors
* stakeholders: sreda, sjayakum, lab members
* problems: overdesign,
* comparison
  + Ryan's solution: -: performance monitoring, +: easier interface
* examples
  + ir camera data collection: instruments fire events, data collector records, GUI + interface, multithreaded, post-process data, low overhead, 2 machine setup
* future: more instruments (ir camera), multiple benchmarks, rerun trials, queue up trials, integrate performance monitors, control algorithm (MATLAB)

**Future Features**

* linux perf integration, GUI, MATLAB integration for control algorithm,

**Key Design Challenges**

* code: execute tasks periodically with low overhead, terminate data collection and benchmark simultaneously, timing accuracy, power overhead of monitoring
* launch benchmark, multiple benchmarks with core affinities, time setting
  + fork
  + start data collector, start benchmarks, benchmark terminates, terminate data collector
* data collection
  + 1 process, data collection tasks queued up (real-time or post-process data)
  + multiple processes, 1 per data collection task (post-process data)
  + threads (real-time or post-process data)
* timing: alarms, sleep

**pfmon**

* perl: @, %, fork, exec, join, die, system, pipes, pid
* -: no documentation, no source control, outdated code, no long term support
* settings: output, debug, time, print interval, # cores, file headers, events, keywords for benchmarks,
* functionality: launch + kill benchmarks, MATLAB, multithreaded

**Design**

* multithreaded
* launch benchmark internally
* basic requirements: power, temperature, TEC V, TEC I