SimGreen 101

Energize your publications with SimGrid

Power-up (or down) your application with SimGrid

Da SimGrid Team

June 11, 2015

Energy in SimGrid

Available Now (in v3.12 or higher)

- Power consumption depends on the CPU utilization
- Dynamic Voltage Frequency Scaling
  - changing the frequency of the CPU
  - power consumption and computing power change accordingly
- Switching on/off
  - OFF hosts have a fixed power consumption (boot on LAN)
  - Boot up / shut down can be given a duration and an energy consumption
  - Turning off hosts kills processes; Turning on restarts some of them

```
<process host="host1" function="master" on_failure="restart"/>
MSG_process_auto_restart_set(proc, 1);
```

TBD sooner or later

- Power consumption depending on disk, network usage
- Power models for virtual machines

Probably not adapted to SimGrid granularity

- Energy models of the L2 vs. L3 cache hits

Outline

- Dynamic Voltage Frequency Scaling
- Switching on and off hosts
- More information on energy in SimGrid

Outline

- Dynamic Voltage Frequency Scaling
- Switching on and off hosts
- More information on energy in SimGrid

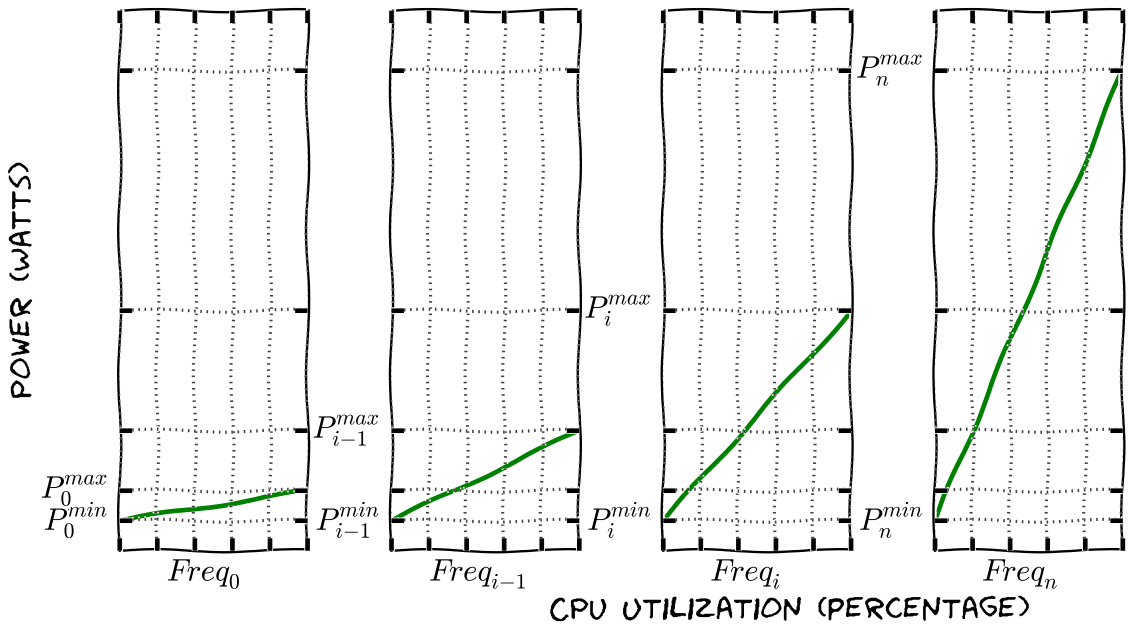
DVFS and Energy Model

DVFS: Dynamic Voltage and Frequency Scaling

- Every modern CPU can reduce its computation speed to save energy
- pstate: levels of performance (CPU frequency). *Governors* pick the right one
- In SimGrid: you manually switch between pstates, which change the flop rate

Energy Model

- For a given pstate, power consumption is a linear function of the CPU use
- Classically accepted model in the literature, rarely challenged



Basic Energy Model Instanciation

```
<host id="MyHost2" power="100.0Mf" >
   <prop id="watt_per_state" value="100.0:200.0" />
   <prop id="watt_off" value="10" />
</host>
```

watt_off power consumption when the host is switched off $\Longrightarrow$ 10 Watts

watt_per_state power consumption interval [min:max]

- Idling host $\Longrightarrow$ 100 Watts
- Fully loaded host (`100.0Mf`=100 MFlops/s) $\Longrightarrow$ 200 Watts
- Linear model in between: CPU loaded at 50% $\Longrightarrow$ 150 Watts

DVFS Energy Model Instanciation

```
<host id="MyHost1" power="100.0Mf,50.0Mf,20.0Mf" pstate="0" >
   <prop id="watt_per_state" value="95.0:200.0, 93.0:170.0,
         90.0:150.0" />
   <prop id="watt_off" value="10" />
</host>
```

power 3 pstates (starting at pstate 0): 100 Mflops/s, 50 Mflops/s, 20 Mflops/s

pstate Starting pstate of that host (here, pstate=0, ie. 100 Mflops/s)

watt_per_state two power values (min:max as before) for each pstate

- Here, CPU loaded at 50% in pstate 2 consumes 120 Watts.
- Remember, pstates are numbered from 0: pstate 2 is 20 Mflops/s peak

watt_off as before

User-side API

Initialization

- `sg_energy_plugin_init();` $\longrightarrow$ ⚠ call it before `MSG_init()`

DVFS and switching pstates

- Get total number of pstates on the given host:
  `int MSG_host_get_nb_pstates (msg_host_t host)`
- Switch the pstate:
  `void MSG_host_set_pstate (msg_host_t host, int pstate)`
- Get the current pstate:
  `int MSG_host_get_pstate (msg_host_t host)`
- Get current speed (in flop/s):
  `double MSG_host_get_current_power_peak(msg_host_t host)`
- Get the speed (in flop/s) for a given pstate:
  `double MSG_host_get_power_peak_at(msg_host_t host, int pstate)`

User-side API (2/2)

Tracking (and predicting) Energy Consumption

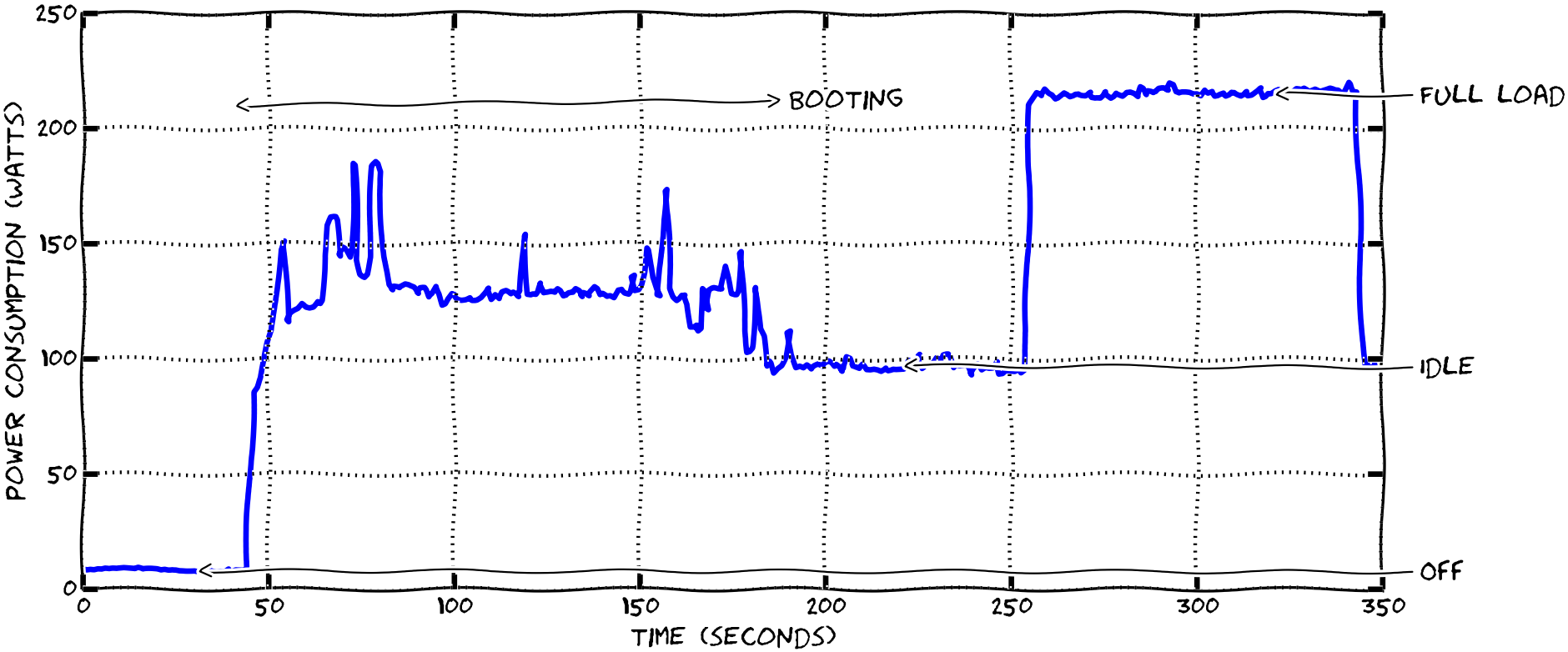
- Get total energy consumed so far:
  `double MSG_host_get_consumed_energy (msg_host_t host)`
- Get the max power value (in Watts) for a given pstate:
  `double MSG_host_get_wattmax_at(msg_host_t host, int pstate)`
- Get the min power value (in Watts) for a given pstate:
  `double MSG_host_get_wattmin_at(msg_host_t host, int pstate)`

Outline

- Dynamic Voltage Frequency Scaling
- Switching on and off hosts
- More information on energy in SimGrid

On/off energy model

- Switching between on and off takes time (seconds) and energy (Joules).



Many ways to do it

- No easy model of the noisy phenomenon: everybody wants something specific
- So SimGrid provides basic mechanisms, and you have to help yourself
- Switching on/off with `MSG_host_on()` and `MSG_host_off()` is instantaneous

All you need is pstates

Proposal: Declare *virtual pstates* in your xml to encode booting, etc

- For a boot taking 150 seconds and 18 000 Joules, create new pstate 3 with:
  - Computing speed: 1 flop / 150 seconds = 0.006666667f
  - Energy power: 18 000 Joules / 150 seconds = 120 Watts
- For a shut down taking 7 seconds and 770 Joules, create new pstate 4 with:
  - Computing speed: 1 flop / 7 seconds = 0.1429f
  - Energy power: 770 Joules / 7 seconds = 110 Watts

```
<host id="MyHost1"  pstate="0"
      power="100.0Mf,50.0Mf,20.0Mf, 0.006666667f,0.1429f" >
  <prop id="watt_per_state"
        value="95.0:200.0,93.0:170.0,
               90.0:150.0,120:120,110:110" />
</host>
```

Useful API

- Switch off a host directly: `void MSG_host_off(msg_host_t host)`
- Switch on a host directly: `void MSG_host_on(msg_host_t host)`
- Test if a host is up: `int MSG_host_is_on(msg_host_t host)`

Actually switching ON a host

```
void simulate_bootup(msg_host_t host) {

  int previous_pstate = MSG_host_get_pstate(host);
  XBT_INFO("Switch to virtual pstate 3, that encodes the shutting
           down state in the XML file of that example");
  MSG_host_set_pstate(host,3);

  XBT_INFO("Actually start the host");
  MSG_host_on(host);

  XBT_INFO("Simulate the boot up by executing one flop on that host");
  msg_host_t host_list[1] = {host};
  double flops_amount[1] = {1};
  double bytes_amount[1] = {0};
  msg_task_t bootup = MSG_parallel_task_create("boot up", 1, host_list,
                      flops_amount, bytes_amount, NULL);
  MSG_task_execute(bootup);
  MSG_task_destroy(bootup);

  XBT_INFO("Switch back to previously selected pstate %d", previous_pstate);
  MSG_host_set_pstate(host, previous_pstate);
}
```

Feel the power of doing your own model

Actually switching OFF a host

```
void simulate_shutdown(msg_host_t host) {

  int previous_pstate = MSG_host_get_pstate(host);

  XBT_INFO("Switch to virtual pstate 4, that encodes the shutting
           down state in the XML file of that example");
  MSG_host_set_pstate(host,4);

  XBT_INFO("Simulate the shutdown by executing one flop on that remote
           host (using a parallel task)");
  msg_host_t host_list[1] = {host};
  double flops_amount[1] = {1};
  double bytes_amount[1] = {0};
  msg_task_t shutdown = MSG_parallel_task_create("shutdown", 1, host_list,
                        flops_amount, bytes_amount, NULL);
  MSG_task_execute(shutdown);
  MSG_task_destroy(shutdown);

  XBT_INFO("Switch back to previously selected pstate %d", previous_pstate);
  MSG_host_set_pstate(host, previous_pstate);

  XBT_INFO("Actually shutdown the host");
  MSG_host_off(host);
}
```

Feel the power of doing your own model

Outline

- Dynamic Voltage Frequency Scaling
- Switching on and off hosts
- More information on energy in SimGrid

Useful pointers

Documentation

- The SURF Energy Plugin is actually documented!
  `http://simgrid.org/simgrid/3.12/doc/group__SURF__plugin__energy.html`
- MSG host management functions
  `http://simgrid.org/simgrid/3.12/doc/group__m__host__management.html`

Examples in the archive (3.12 and higher)

- Platform file: *platforms/energy_platform.xml*
- DVFS: *msg/energy/pstate/pstate.c*
- DVFS and direct switch off: *msg/energy/consumption/energy_consumption.c*
- Model of on/off with switching duration and power consumption: *msg/energy/onoff/onoff.c*

Publication

- Currently underway, TBD for GreenCom 2015?