Consiglio Nazionale
delle Ricerche

Report for Stochastic Model Checking

Academic Year 2016/2017

Tommaso Papini
tommaso.papini@unifi.it
DT21263

PRISM Tutorial Part 3: Dynamic power management

In Section will be described the modelization through PRISM of a *DPM* (*Dynamic Power Management*) system, following the PRISM tutorial found at [1]. DPMs are used to apply different power usages to some computing device, according to a predefined strategy that takes into account the current state of the device. This kind of systems have been studied largely in literature, for example in [2] where a DPM for a Fujitsu disk drive has been studied.

A generic DPM system is made of three distinct components:

- *Service Queue* (*SQ*): holds the requests that the Service Provider will have to serve, in an ordered fashion, and can have finite queue capacity;
- *Service Provider* (*SP*): serves, one at a time, the requests stored in the Service Queue, serving each time the request at the head of the queue;
- *Power Manager* (*PM*): can change the power state of the Service Provider according to certain policies.

The *SP* could be anything that is a computing device that serves requests, such as a disk drive as in [2], but also a CPU or a Web Server.

At any given time, the *SP* is in one of three possible power states, each of which:

- *sleep*: the *SP* is in a low-power consumption mode and is unable to serve any request unless explicitly awaken by the *PM*;
- *idle*: the *SP* is awake but currently not serving any request, so any newly arriving request will be served immediately by the *SP*;

- *busy*: the *SP* is currently serving a request and will be available to serve the next in queue as soon as it's finished.

Ideally, when in the *sleep* state the *SP* will be requiring little to none power, when in the *idle* state it will require more, as it is awake and ready to serve requests, while when *busy* it will require even more, as the *SP* in that case is actively working on a request. The *PM* is charged with employing a power consumption strategy by switching the *SP*'s power state, in order to maximise the availability of the service while minimising the overall power consumption.

A first PRISM model for a DPM based on [2] is proposed in Code 1, as seen in [1].

```
// Simple dynamic power management (DPM) model
// Based on:
// Qinru Qiu, Qing Wu and Massoud Pedram
// Stochastic modeling of a power-managed system: Construction and optimization
// Proc. International Symposium on Low Power Electronics and Design, pages 194--199, ACM Press,
    1999

ctmc

//-----------------------------------------------------------------------------

// Service Queue (SQ)
// Stores requests which arrive into the system to be processed.

// Maximum queue size
const int q_max = 20;

// Request arrival rate
const double rate_arrive = 1/0.72; // (mean inter-arrival time is 0.72 seconds)

module SQ

    // q = number of requests currently in queue
    q : [0..q_max] init 0;

    // A request arrives
    [request] true -> rate_arrive : (q'=min(q+1,q_max));
    // A request is served
    [serve] q>1 -> (q'=q-1);
    // Last request is served
    [serve_last] q=1 -> (q'=q-1);

endmodule

//-----------------------------------------------------------------------------

// Service Provider (SP)
// Processes requests from service queue.
// The SP has 3 power states: sleep, idle and busy

// Rate of service (average service time = 0.008s)
const double rate_serve = 1/0.008;

module SP

    // Power state of SP: 0=sleep, 1=idle, 2=busy
    sp : [0..2] init 1;
```

```
47
48      // Synchronise with service queue (SQ):
49
50      // If in the idle state, switch to busy when a request arrives in the queue
51      [request] sp=1 -> (sp'=2);
52      // If in other power states when a request arrives, do nothing
53      // (need to add this explicitly because otherwise SP blocks SQ from moving)
54      [request] sp!=1 -> (sp'=sp);
55
56      // Serve a request from the queue
57      [serve] sp=2 -> rate_serve : (sp'=2);
58      [serve_last] sp=2 -> rate_serve : (sp'=1);
59
60  endmodule
61
62  //-------------------------------------------------------------------
63
64  // Reward structures
65
66  rewards "queue_size"
67      true : q;
68  endrewards
```

Code 1: PRISM code for the model of a DPM based on [2]. Source [1].

References

[1] PRISM Team. PRISM Tutorial Part 3: Dynamic power management. `http://www.prismmodelchecker.org/tutorial/power.php`. [Online; accessed 18-September-2017].

[2] Qiu, Q., Qu, Q., and Pedram, M. Stochastic modeling of a power-managed system-construction and optimization. *IEEE Transactions on computer-aided design of integrated circuits and systems 20*, 10 (2001), 1200–1217.