Sami Evangelista
sami.evangelista@lipn.univ-paris13.fr

This form is a summary description of the model entitled "Simple load balancing system" proposed for the Model Checking Contest @ Petri Nets. Models can be given in several instances parameterized by scaling parameters. Colored nets can be accompanied by one or many equivalent, unfolded P/T nets. Models are given together with property files (possibly, one per model instance) giving a set of properties to be checked on the model.

Description

This net models a simple load balancing system composed of a set of clients, two servers, and between these, a load balancer process (called lb process thereafter).

The clients Their role is to send requests to servers (transition client_send), wait for an answer and get it (transition client_receive). Note that requests are not directly sent to servers but to the lb process so that this one routes it to the appropriate server (i.e., tokens modeling requests are put in place client_request before being put in server_request).

The servers Each of the two servers waits for requests (i.e., tokens in place server_request). When such a request arrives it processes it and send a reply to the client (transition server_process). The server then notifies the lb process (transition server_notify) so that this one possibly rebalances requests between servers. Once the lb process has acknowledged this notification, the server can go back to the idle state (transition server_endloop).

The lb process The lb process has the most complex task.

First, in the idle state, it can receive a request from a client (transition lb_receive_client). It then routes this request either to server 1 (transition lb_route_to_1) either to server 2 (transition lb_route_to_2) depending on the loads of these two servers. Place lb_load is a key element of the net as it used by the lb process to guide its actions. This place always has two tokens $(1, l1)$ and $(2, l2)$ where $l1$ (resp. $l2$) is the number of requests assigned to server 1 (resp. 2).

Second, when a server notifies the lb process that it has completed a request, the lb process records this information by modifying the content of place lb_load and then goes to the "balancing" state (transition lb_idle_receive_notification) in order to possibly reassign requests between servers. In this state (place lb_balancing), the lb process can perform four actions. The first one is to go back to the idle state if loads are already balanced (transition lb_no_balance). Transition lb_balance_to_1 (resp. lb_balance_to_2) models the situation where server 2 (resp. server 1) has more requests to handle than server 1 (resp. server 2). The lb process then reroutes one request from one server to the other. In these first three situations (transitions lb_no_balance, lb_balance_to_1 and lb_balance_to_2), the lb process goes back to the idle state. At last, in the "balancing" state, the server must treat server notifications (transition lb_balancing_receive_notification) in order to keep an up-to-date information on the loads of servers and correctly rebalances.

References

Model borrowed from the Helena tool distribution (slightly modified).

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
N	Number of clients	2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20

Size of the colored net model

number of places:	14
number of transitions:	13
number of arcs:	57

Sami Evangelista
sami.evangelista@lipn.univ-paris13.fr

Size of the derived P/T model instances

Parameter	Number of places	Number of transitions	Number of arcs
$N = 2$	32	45	252
$N = 5$	59	180	1158
$N = 10$	104	605	4148
$N = 15$	149	1280	8988
$N = 20$	194	2205	15678

Structural properties

ordinary — *all arcs have multiplicity one* ✔
simple free choice — *all transitions sharing a common input place have no other input place* ✗ [a]
extended free choice — *all transitions sharing a common input place have the same input places* ✗ [b]
state machine — *every transition has exactly one input place and exactly one output place* ✗ [c]
marked graph — *every place has exactly one input transition and exactly one output transition* ✗ [d]
connected — *there is an undirected path between every two nodes (places or transitions)* ✔ [e]
strongly connected — *there is a directed path between every two nodes (places or transitions)* ✔ [f]
source place(s) — *one or more places have no input transitions* ✗ [g]
sink place(s) — *one or more places have no output transitions* ✗ [h]
source transition(s) — *one or more transitions have no input places* ✗ [i]
sink transitions(s) — *one or more transitions have no output places* ✗ [j]
loop-free — *no transition has an input place that is also an output place* ✗ [k]
conservative — *for each transition, the number of input arcs equals the number of output arcs* ✗ [l]
subconservative — *for each transition, the number of input arcs equals or exceeds the number of output arcs* ✗ [m]
nested units — *places are structured into hierarchically nested sequential units* [n] ✗

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place* ✔ [o]
deadlock — *there exists a reachable marking from which no transition can be fired* ✗ [p]
reversible — *from every reachable marking, there is a transition path going back to the initial marking* ✗
quasi-live — *for every transition t, there exists a reachable marking in which t can fire* ? [q]
live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire* ? [r]

(a) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(b) transitions “T-server_process_1” and “T-server_process_2” share a common input place “P-server_idle_1”, but only the former transition has input place “P-server_request_1_1”.
(c) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(d) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(e) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(f) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(g) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(h) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(i) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(j) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(k) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(l) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(m) stated by CÆSAR.BDD version 1.7 on all 5 instances (2, 5, 10, 15, and 20).
(n) the definition of Nested-Unit Petri Nets (NUPN) is available from `http://mcc.lip6.fr/nupn.php`
(o) stated by CÆSAR.BDD version 2.0 to be true on 3 instance(s) out of 5, and unknown on the remaining 2 instance(s).
(p) stated by CÆSAR.BDD version 2.0 to be false on 3 instance(s) out of 5, and unknown on the remaining 2 instance(s); confirmed at MCC’2014 by Helena on 3 colored instances ($N = 2$, $N = 5$, and $N = 10$), and by GreatSPN, Lola, and Tapaal on the 3 corresponding P/T instances.
(q) found to be false by CÆSAR.BDD version 1.9 on unfolded net instances 2, 5, and 10.
(r) found to be false by CÆSAR.BDD version 1.9 on unfolded net instances 2, 5, and 10, which are not quasi-live.

Sami Evangelista
sami.evangelista@lipn.univ-paris13.fr

Size of the marking graphs

Parameter	Number of reachable markings	Number of transition firings	Max. number of tokens per place	Max. number of tokens per marking
$N = 2$	832 [s]	2650 [t]	1 [u]	11 [v]
$N = 5$	116,176 [w]	566,332 [x]	1 [y]	17 [z]
$N = 10$	4.06034×10^8 [aa]	3.05120×10^9 [ab]	1 [ac]	27 [ad]
$N = 15$	1.374×10^{12} [ae]	?	?	≥ 20 [af]
$N = 20$	4.583×10^{15} [ag]	?	?	≥ 25

(s) computed by Alpina, ITS-Tools, Marcie, Neco, and PNXDD at MCC'2013; confirmed by CÆSAR.BDD version 1.8; confirmed by GreatSPN, PNMC, Stratagem, and Tapaal at MCC'2014.
(t) computed by Marcie at MCC'2014.
(u) computed by GreatSPN, PNMC, and Tapaal at MCC'2014.
(v) computed by GreatSPN, PNMC, and Tapaal at MCC'2014.
(w) computed by ITS-Tools, Marcie, Neco, and PNXDD at MCC'2013; confirmed by CÆSAR.BDD version 1.8; confirmed by GreatSPN, PNMC, Stratagem, and Tapaal at MCC'2014.
(x) computed by Helena and Marcie at MCC'2014.
(y) computed by GreatSPN, PNMC, Marcie, and Tapaal at MCC'2014.
(z) computed by GreatSPN, PNMC, Marcie, and Tapaal at MCC'2014.
(aa) computed by ITS-Tools, Marcie, and PNXDD at MCC'2013; confirmed by CÆSAR.BDD version 1.8; confirmed by GreatSPN and PNMC at MCC'2014; exact value: 406,034,376.
(ab) computed by Marcie at MCC'2014; exact value: 3,051,203,628.
(ac) computed by GreatSPN, PNMC, and Marcie at MCC'2014.
(ad) computed by GreatSPN, PNMC, and Marcie at MCC'2014.
(ae) computed by PNXDD at MCC'2013.
(af) lower bound given by the number of initial tokens.
(ag) computed by PNXDD at MCC'2013.