Hugues Evrard
hugues.evrard@inria.fr

This form is a summary description of the model entitled "Raft" 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

Consensus algorithms ensure the replication of a log among a set of faulty servers, thus enabling the implementation of fault-tolerant services replicated on several machines. Nowadays, such algorithms are crucial for developpers of cloud services that must survive data center outages.

The standard consensus algorithm, the famous Paxos designed by L. Lamport, is notoriously complex and thus difficult to implement correctly. Therefore, a new consensus algorithm named Raft has been proposed recently [OO14]. Raft aims at providing the same guarantees as Paxos while being simpler to grasp and thus easier to implement. The rapid adoption of Raft by several companies (Facebook, Hashicorp, CoreOS) illustrates the need for a "simpler Paxos".

In Raft, time is divided in terms and at most one server can be elected as leader for a given term. The leader is in charge of commiting client requests on a majority of servers, and server crashes or network partitions can trigger a leader election for a new term.

The present P/T nets are derived from the formal specification in LNT of the Raft consensus protocol, where servers can crash and restart at any moment. There are several instances for various numbers of Raft servers, ranging from 2 to 10. Each instance models two client requests and allows up to two crashes per server.

This LNT specification was used as a case study for the DLC (*Distributed LNT Compiler*) tool [EL15]. DLC is able to automatically generate for Raft a distributed implementation in C, which can be deployed on a cluster of machines communicating via sockets.

The LNT specification was translated to LOTOS, and then to an interpreted Petri net using the CADP toolbox. Finally, the present P/T net was obtained by stripping out all dataflow-related information (variables, types, assignments, guards, etc.) from the interpreted Petri net, leading to a NUPN (*Nested-Unit Petri Net*) model translated to PNML using the CÆSAR.BDD tool.

References

[OO14] Diego Ongaro and John Ousterhout. *In Search of an Understandable Consensus Algorithm.* Proceedinsg of the USENIX Annual Technical Conference (USENIX ATC 14), Philadelphia, PA, USA, June 2014, pages 305–319. USENIX Association. http://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro

[EL15] Hugues Evrard and Frédéric Lang. *Automatic Distributed Code Generation from Formal Models of Asynchronous Concurrent Processes.* Proceedings of the 23rd Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP'15), Turku, Finland. IEEE, 2015. http://hal.inria.fr/hal-01086522

See also:

- *Raft web site:* http://raftconsensus.github.io
- *Verification based on CADP found a missing transition in a Raft specification:* http://groups.google.com/forum/#!topic/raft-dev/yu-wOUx-gnA

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
N	N: number of Raft servers	2, 3, 4, 5, 6, 7, 8, 9, and 10

Hugues Evrard
hugues.evrard@inria.fr

Size of the model

Parameter	Number of places	Number of transitions	Number of arcs	Number of units	HWB code
$N = 2$	28	52	159	7	2–6–18
$N = 3$	53	98	328	12	2–11–33
$N = 4$	88	160	563	19	2–18–54
$N = 5$	133	238	864	28	2–27–81
$N = 6$	188	332	1231	39	2–38–114
$N = 7$	253	442	1664	52	2–51–153
$N = 8$	328	568	2163	67	2–66–198
$N = 9$	413	710	2728	84	2–83–249
$N = 10$	508	868	3359	103	2–102–306

Structural properties

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

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place* ..... ✔ [(p)]
deadlock — *there exists a reachable marking from which no transition can be fired* ..... ? [(q)]
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* ..... ? [(r)]

(a) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(b) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(c) stated by CÆSAR.BDD version 2.6 on all 9 instances.
(d) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(e) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(f) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(g) from place 1 one cannot reach place 0.
(h) place 0 is a source place.
(i) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(j) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(k) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(l) stated by CÆSAR.BDD version 2.2 on all 9 instances.
(m) 3 transitions are not conservative, e.g., transition 0.
(n) transition 0 is not subconservative.
(o) the definition of Nested-Unit Petri Nets (NUPN) is available from http://mcc.lip6.fr/nupn.php
(p) safe by construction – stated by the CÆSAR compiler.
(q) stated by CÆSAR.BDD version 2.2 to be false on 5 instance(s) out of 9, and unknown on the remaining 4 instance(s).
(r) stated by CÆSAR.BDD version 2.2 to be true on 5 instance(s) out of 9, and unknown on the remaining 4 instance(s).

Hugues Evrard
hugues.evrard@inria.fr

live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire* ..... ?

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
$N = 2$	7381 [(s)]	?	1	6 [(t)]
$N = 3$	3.38196e+07 [(u)]	?	1	11 [(v)]
$N = 4$	2.96586e+12 [(w)]	?	1	18 [(x)]
$N = 5$	5.93795e+18 [(y)]	?	1	27 [(z)]
$N = 6$	2.90944e+26 [(aa)]	?	1	38 [(ab)]
$N = 7$	$\geq$ 2.46637e+35 [(ac)]	?	1	51 [(ad)]
$N = 8$	$\geq$ 4.87881e+45 [(ae)]	?	1	66 [(af)]
$N = 9$	$\geq$ 4.83981e+52 [(ag)]	?	1	83 [(ah)]
$N = 10$	$\geq$ 1.84049e+66 [(ai)]	?	1	102 [(aj)]

(s) stated by CÆSAR.BDD version 2.2.
(t) stated by CÆSAR.BDD version 2.2.
(u) stated by CÆSAR.BDD version 2.2.
(v) stated by CÆSAR.BDD version 2.2.
(w) stated by CÆSAR.BDD version 2.2.
(x) stated by CÆSAR.BDD version 2.2.
(y) stated by CÆSAR.BDD version 2.2.
(z) stated by CÆSAR.BDD version 2.2.
(aa) stated by CÆSAR.BDD version 2.2.
(ab) stated by CÆSAR.BDD version 2.2.
(ac) stated by CÆSAR.BDD version 2.2.
(ad) stated by CÆSAR.BDD version 2.6.
(ae) stated by CÆSAR.BDD version 2.2.
(af) stated by CÆSAR.BDD version 2.6.
(ag) stated by CÆSAR.BDD version 2.2.
(ah) stated by the CÆSAR compiler.
(ai) stated by CÆSAR.BDD version 2.2.
(aj) stated by the CÆSAR compiler.