Claude Girault
Claude.Girault@lip6.fr

Description

Let be a set C of client, a set S of servers, a set M of managers and a pool of U resource units. A client may send a request a resource to the server set. Any server may transmit the request to the manager set and waits for a grant. Any manager may allocate a resource unit from the pool and returns a grant. A waiting server transmits this grant to the client. After using the resource the client frees it by sending a message to the server set. Any server may notify the manager set and waits for an acknowledge that it transmits it to the client.

An idle client (place **Ci**) sends (transition **csR**) a client request (place **CR**) to the server set and waits (place **CwG**) until it receives (transition **crG**) a grant (place **CG**) from the server allowing access to a resource unit.

- An idle server (place **Si**) may receive (transition **srR**) a request from a client, treats it (place **StR**) and sends (transition **ssR**) a server request (place **SR**) to the manager set. Then, it waits (place **SwG**) for a grant (place **SG**) from a manager.
- An idle manager (place **Mi**) may receive (transition **mrR**) a server request, it waits (place **MwU**) for a free ressource unit (place **Uf**) that it allocates (transition **maU**). Then, it prepares a grant (place **MpG**) that it sends (transition **msG**) to the server and becomes again idle.
- When a waiting server receives (transition **srG**) a grant (place **SG**) from the manager, it prepares (place **SpG**) a grant (place **CG**) for the client, sends it (transition **ssG**), and becomes again in state idle.

When a waiting client receives (transition **crG**) a grant (place **CG**) from a server, it becomes busy (place **Cb**) using the resource unit. After finishing it sends (transition **csF**) a free message (place **CF**) to a server and waits (place **CwA**) for an acknowledge.

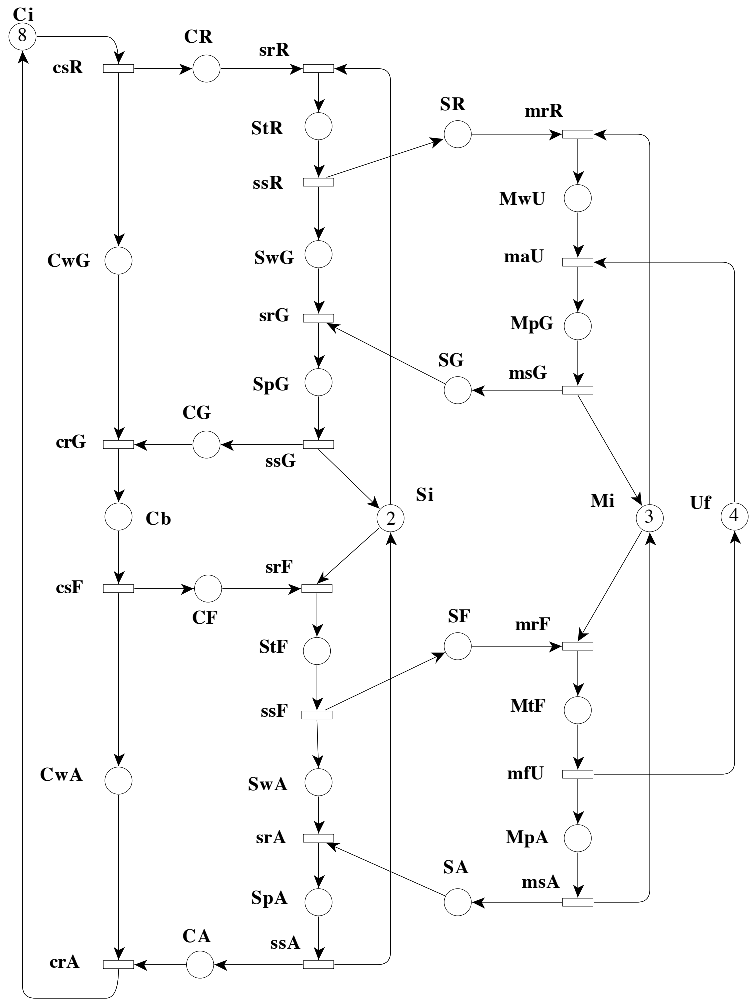
- When an idle server (place **Si**) receives (transition **srF**) a free message (place **CF**) from a client, it treats it (place **StF**) and sends (transition **ssF**) a server free message (place **SF**) to the manager set. Then, it waits (place **Sw**) for an acknowledge (place **SA**).
- When an idle manager receives (transition **mrF**) a free message (place **SF**) from a server, it treats it (place **MtF**) and releases (transition **mfree**) the resource. Then, it prepares (place **MpA**) an acknowledge (place **SA**) for the server, sends it (transition **msA**), and becomes again in state idle.
- When a waiting server (place **SwA**) receives (transition **srA**) an acknowledge (place **SA**) from the manager. It prepares (place **SpA**) an acknowledge (place **CA**) for the client, sends it (transition **ssA**), and becomes again in state idle.

When a waiting client (place **CwA**) receives (transition **crA**) this acknowledge, it becomes again in state idle.

In its most general form, the model has four parameters: the number $|C|$ of clients, the number $|S|$ of servers, the number $|M|$ of managers, and the number $|U|$ of resources units.

We consider here a simplified form, in which the model is parameterized by two natural numbers N and P (with $N > P$) such that

- the number of clients is set to $C = 8N$
- the number of servers is set to $S = 2N$
- the number of managers is set to $M = 3N - 3P$
- the number of resource units is set to $U = 4N + 8P$.



Graphical representation for $N = 1$ and $P = 0$

Scaling parameter

Parameter name	Parameter description	Chosen parameter values
N and P, two linear factors	The scaling parameters only change the initial marking (see description)	(1,0), (2,0), (2,1), (5,0), (5,1), (10,0), (10,1), (10,2), (20,0), (20,1), (20,2), (20,3), (20,4), (50,0), (100,0), (200,0), (500,0), (1000,0), (2000,0), (5000,0)

Size of the model

Although the model is parameterized, its size does not depend on parameter values.

number of places:	25
number of transitions:	18
number of arcs:	54

Structural properties

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

Behavioural properties

safe — *in every reachable marking, there is no more than one token on a place*	no [o]
dead place(s) — *one or more places have no token in any reachable marking*	no [p]
dead transition(s) — *one or more transitions cannot fire from any reachable marking*	no [q]
deadlock — *there exists a reachable marking from which no transition can be fired*	? [r]
reversible — *from every reachable marking, there is a transition path going back to the initial marking*	?
live — *for every transition t, from every reachable marking, one can reach a marking in which t can fire*	?

[a] 4 arcs are not simple free choice, e.g., the arc from place "Mi" (which has 2 outgoing transitions) to transition "mrF" (which has 2 input places).
[b] transitions "mrF" and "mrR" share a common input place "Mi", but only the former transition has input place "SF".
[c] 18 transitions are not of a state machine, e.g., transition "ssF".
[d] 2 places are not of a marked graph, e.g., place "Mi".
[e] confirmed by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[f] stated by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[g] confirmed by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[h] confirmed by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[i] confirmed by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[j] confirmed by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[k] stated by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[l] 18 transitions are not conservative, e.g., transition "ssF".
[m] 9 transitions are not subconservative, e.g., transition "ssF".
[n] the definition of Nested-Unit Petri Nets (NUPN) is available from `http://mcc.lip6.fr/nupn.php`
[o] in the initial marking, some places have several tokens; confirmed by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[p] stated by CÆSAR.BDD version 3.3 on all 20 instances (see aforementioned parameter values).
[q] stated by CÆSAR.BDD version 2.7 on all 20 instances (see aforementioned parameter values).
[r] depends on the initial marking.

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
$N = 1, P = 0$	27 576[s]	113 316[t]	?	≥ 17[u]
$N = 2, P = 0$	7 081 638[v]	44 030 250[w]	?	≥ 34[x]
$N = 2, P = 1$	12 462 173[y]	77 859 168[z]	?	≥ 39[aa]
$N = 5, P = 0$	$>$ 39 919 315[ab]	$>$ 240 893 998[ac]	?	≥ 85[ad]
$N = 5, P = 1$	$>$ 24 676 885[ae]	$>$ 161 268 212[af]	?	≥ 90[ag]
$N = 10, P = 0$	?	?	?	≥ 170[ah]
$N = 10, P = 1$	?	?	?	≥ 175[ai]
$N = 10, P = 2$	?	?	?	≥ 180[aj]
$N = 20, P = 0$	?	?	?	≥ 340[ak]
$N = 20, P = 1$	?	?	?	≥ 345[al]
$N = 20, P = 2$	?	?	?	≥ 350[am]
$N = 20, P = 3$	?	?	?	≥ 355[an]
$N = 20, P = 4$	?	?	?	≥ 360[ao]
$N = 50, P = 0$	?	?	?	≥ 850[ap]
$N = 100, P = 0$	?	?	?	≥ 1700[aq]
$N = 200, P = 0$	?	?	?	≥ 3400[ar]
$N = 500, P = 0$	?	?	?	≥ 8500[as]
$N = 1000, P = 0$	?	?	?	≥ 17000[at]
$N = 2000, P = 0$	?	?	?	≥ 34000[au]
$N = 5000, P = 0$	?	?	?	≥ 85000[av]

Other properties

There are deadlocks iff $|C| \geq |S| + |U|$ and $|C| \geq |M| + |U|$.

The types of deadlock also depend whether $|S| \geq |M|$ and whether $|S| \geq |M| + |U|$.

[s] stated by prod in April 2017.
[t] stated by prod in April 2017.
[u] lower bound given by the number of initial tokens.
[v] stated by prod in April 2017.
[w] stated by prod in April 2017.
[x] lower bound given by the number of initial tokens.
[y] stated by prod in April 2017.
[z] stated by prod in April 2017.
[aa] lower bound given by the number of initial tokens.
[ab] stated by prod in April 2017.
[ac] stated by prod in April 2017.
[ad] lower bound given by the number of initial tokens.
[ae] stated by prod in April 2017.
[af] stated by prod in April 2017.
[ag] lower bound given by the number of initial tokens.
[ah] lower bound given by the number of initial tokens.
[ai] lower bound given by the number of initial tokens.
[aj] lower bound given by the number of initial tokens.
[ak] lower bound given by the number of initial tokens.
[al] lower bound given by the number of initial tokens.
[am] lower bound given by the number of initial tokens.
[an] lower bound given by the number of initial tokens.
[ao] lower bound given by the number of initial tokens.
[ap] lower bound given by the number of initial tokens.
[aq] lower bound given by the number of initial tokens.
[ar] lower bound given by the number of initial tokens.
[as] lower bound given by the number of initial tokens.
[at] lower bound given by the number of initial tokens.
[au] lower bound given by the number of initial tokens.
[av] lower bound given by the number of initial tokens.