Workstation Cluster

AVACS S3
Phase 2

July 28, 2011

1 Description of the Model

In this test case, we consider the *dependability of a fault-tolerant workstation cluster* [5]. Notably, we consider a finite-state model leading to finite CTMCs. Taking into account different aspects of the model, we can consider it either as a Continuous-Time Markov Chain (CTMC) [5] or as a Continuous-Time Markov Decision Process (CTMDP) [7]. We can handle the CTMC version by our tool INFAMY [4] whereas the CTMDP version can be treated by our modified version of the probabilistic model checker MRMC [3, 7]

Figure 1 depicts a sketch of the aforementioned dependable cluster of workstations. The cluster consists of two sub-clusters, which, in turn, contain N workstations connected via a central switch. The two switches are connected via a backbone. Each component of the system can break down (fail) and is then fixed by a single repair unit responsible for the entire system (not depicted in the figure). Hereby, the quality of

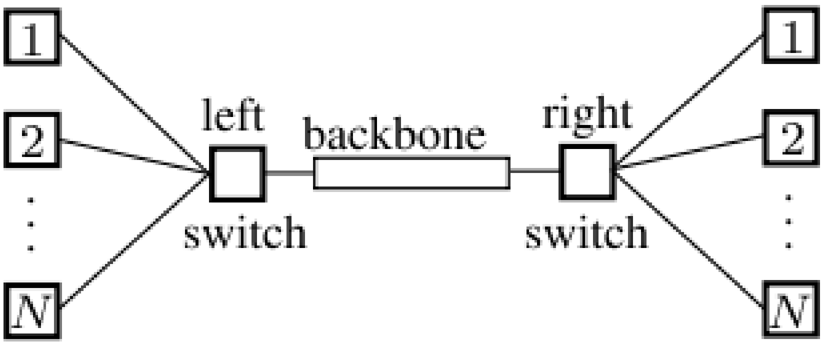

Figure 1: Dependable Cluster of Workstations

service (QoS) constraint *Minimum* requires at least k ($k < N$) workstations to be operational where $k = \lfloor \frac{3}{4} \cdot N \rfloor$. Workstations have to be connected via switches. If in each sub-cluster the number of operational workstations is smaller than k the backbone is required to be operational to provide the required service.

We say that our system provides *premium* service whenever at least N workstations are operational. These workstations have to be connected by each other via operational switches. When the number of operational workstations in one sub-cluster is below N, premium quality can be ensured by an operational backbone under the condition that there are at least N operational workstations in total. We consider the following properties:

1. $P_1 : P_{=?}(F^{[0,1]}\neg Minimum)$: The propability that the QoS drops below minimum quality within one time unit,

2. P_2 : The expected number of repairs by time point one. The corresponding *PRISM* property is $R_{=?}[C \leq T]$,

3. P_3 : The probability to reach non-premium service within time T: $P_{\leq x}(F^{[0,T]}\neg premium)$,

4. P_4 : The steady-state probability of having non-premium service: $S_{\leq x}(\neg premium)$ **and**

5. P_5 : The steady-state probability of being in a state where the probability to reach non-premium service within time T is above $\frac{1}{2}$:$S_{\leq x}(\neg P_{<\frac{1}{2}}(F^{[0,T]}\neg premium))$.

Properties P_1 and P_2 are used for performance measures of *INFAMY* whereas the others are tested with our modified version of *MRMC*. In the remainder of this case study, results are presented for both tools.

2 Results for *INFAMY*

In order to compare our tool *INFAMY* [4] against a model checker which does not employ truncation, we use *PRISM* [6] in version 3.2, which was latest when handling this case study. All results were obtained on a Linux machine with an AMD Athlon XP 2600+ processor at 2 GHz equipped with 2 GB of RAM. A comparison of *PRISM* and two configurations of *INFAMY*, namely Layered-chain and FSP, for a various number of workstations is given in Table 1. The Uniform chain configuration is omitted, as it is always dominated by the Layered-chain configuration. *PRISM* implements three different engines: a sparse-matrix and two symbolic engines. We used the sparse-matrix engine as it was the fastest one.

The probability and the expected number of repairs respectively, are depicted in Table 2. Since the resulting probabilites are very small in some cases, we use a precision of 10^{-12} for the computation of the truncation point. The time columns "tm (s)" of Table 1 have the format $t_1/t_2/t_3$ where the first number t_1 denotes the time needed for model construction, t_2 represents the time to check property 1 and t_3 the time for property 2. While the number of states and transitions for *PRISM* increases dramatically with parameter N, *INFAMY* scales much better in both configurations. The reason is apparent from the column displaying the depth: the depth of the full model is approximately linear in N, while the depth needed is sub-linear in N. For $N \geq 2048$ and $N \geq 1024$ respectively, *PRISM* cannot model check these properties as it runs out of memory and

N	*PRISM*			Layered			FSP		
	dep	tm (s)	n	dep	tm (s)	n	dep	tm (s)	n
512	1029	12/ 197/952	9466K	105	2/5/5	193K	20	2/0/0	6K
1024	2053	48/1132/-	37806K	109	2/5/5	208K	26	3/0/0	11K
2048	-	-	-	116	2/6/6	236K	36	6/1/1	22K
4096	-	-	-	129	3/7/8	239K	53	19/1/1	48K

Table 1: Comparison of *PRISM* and *INFAMY* for different N

N	prob.	exp.
512	5.96E-08	0.72
1024	6.01E-08	1.08
2048	.07E-08	1.38
4096	6.11E-08	1.56

Table 2: Propabilities and Expected Number of Repairs for various N

crashes; this is denoted by "-". As N increases, the truncation depth grows only slowly in *INFAMY* and hence, only a small fraction of the state space needs to be explored in the truncation construction. Although the FSP configuration is not very efficient in terms of toal time, thereby spending the largest part of the running time in the first phase, it shows a lot of potential in terms of depth and number of states. In Table 3, we give additional performance measures for a model of $N = 512$ workstations for property 1 using larger time bounds. Results are given for *PRISM* (sparse engine), FSP and FSP exponential respectively. The state space explored by *PRISM* has depth 1029 and includes 9.4 million states. Up to $t = 20$, *INFAMY* with FSP is faster than *PRISM*. However, for larger time bounds, the model construction dominates, as for each layer exploration the error estimate is recomputed. The FSP variant with exponential layer explorations is suitable for this case and is consistently the fastest method for $t \leq 50$, as shown in the last column of Table 4.

3 Results for modified *MRMC*

Time bounded reachability analysis for CTMDPs was thus far restricted to time-abstract policies [7], using a dediated algorithm for uniform CTMDPs [1]. In a uniform CTMDP (including the one studied here) rate sums are identical across states and nondeterministic choices, which can be exploited in the algorithm.

Results and statistics are reported in Table 5, 6 and 7. For P_1, we also give numbers for time-abstract policy-based computation exploiting model uniformity [1]. We chose $\epsilon = 10^{-6}$ and $K_{max} = 70$. As we see, the probabilities obtained for P_1 using time-abstract and general policies agree up to ϵ, thus time-abstract policies seem sufficient to obtain maximal reachability probabilities for this model and property. Our

t	*PRISM* time (s)	FSP depth	time (s)	n	FSP exponential depth	time (s)	n
10.0	11.9/ 359.5	52	47.5/ 3.0	46K	64	6.8/ 4.7	71K
20.0	11.7/ 593.8	80	318.1/ 13.5	111K	128	44.4/ 35.6	289K
30.0	11.7/ 802.2	104	1016.9/ 32.5	190K	128	62.5/ 50.2	289K
50.0	11.7/1171.8	147	4660.4/103.7	382K	256	404.9/323.0	1167K

Table 3: Cluster Performance Statistics for Different Time Bounds

t	prob.
10.0	3.79E-06
20.0	1.01E-05
30.0	1.68E-05
50.0	3.04E-05

Table 4: Results of Property 1 for Different Time Bounds

runtime requirements are higher than what is needed for the time-abstract policy class, if exploiting uniformity [1]. Without uniformity exploitation [2], the time-abstract computations are worse by a factor of 100 to 100000 compared to our analysis (yielding the same probability result, not shown in the table). However, even for the largest models and time bounds considered, we were able to obtain precise results within reasonable time, which shows the practical applicability of the method. Long-run properties P_2 and nested variation P_3 can be handled in a similiar amount of time, compared to P_1. In Appendix A we give a comparison of different versions of *MRMC* and *ETMCC*.

References

[1] C. Baier, H. Hermanns, J. P. Katoen, and B. R. Haverkort. Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov Decision Processes. *Theor. Comput. Sci.*, 345(1):2–26, 2005.

[2] Tomás Brázdil, Vojtech Forejt, Jan Krcal, Jan Kretínský, and Antonín Kucera. Continuous-Time Stochastic Games with Time-Bounded Reachability. In *FSTTCS*, volume 4 of *LIPIcs*, pages 61–72, 2009.

[3] Peter Buchholz, Ernst Moritz Hahn, Holger Hermanns, and Lijun Zhang. Model Checking Algorithms for CTMDPs. In *CAV*, 2011.

[4] Ernst Moritz Hahn, Holger Hermanns, Björn Wachter, and Lijun Zhang. INFAMY: An Infinite-State Markov Model Checker. In *CAV*, pages 641–647, 2009.

Workst.	States		100h (tab)	100h (dep)	500h (tab)	500h (dep)
4	808	Time	0s	0s	0s	0s
		Prob.	0.0921146	0.0921155	0.0921146	0.0921155
8	2770	Time	0s	0s	0s	0s
		Prob.	0.1772726	0.1772735	0.1772726	0.1772735
16	10130	Time	0s	0s	0s	1s
		Prob.	0.3243474	0.3243483	0.3243474	0.3243483
64	151058	Time	0s	4s	1s	19s
		Prob.	0.7324401	0.7324406	0.7324401	0.7324406

Workst.	States		1000h (tab)	1000h (dep)	5000h (tab)	5000h (dep)
4	808	Time	0s	0s	0s	1s
		Prob.	0.0921146	0.0921155	0.0921146	0.0921155
8	2770	Time	0s	0s	0s	2s
		Prob.	0.1772726	0.1772735	0.1772726	0.1772735
16	10130	Time	0s	2s	1s	8s
		Prob.	0.3243474	0.3243483	0.3243474	0.3243483
64	151058	Time	2s	37s	6s	3m 21s
		Prob.	0.7324401	0.7324406	0.7324401	0.7324406

Table 5: Statistics for fault tolerant workstation analysis from the analysis of property P_1. We give both number for the time-abstract algorithm (tab) as well as for the time-dependent algorithm (dep)

Workst.	States		
1	Time Prob.	110	0s 0.0000213
2	Time Prob.	274	0s 0.0000376
4	Time Prob.	818	0s 0.0000770
8	Time Prob.	2770	0s 0.0001635
16	Time Prob.	10130	0s 0.0003483
32	Time Prob.	38674	2s 0.0007050
64	Time Prob.	151058	11s 0.0012808
128	Time Prob.	597100	0s 0.0000000

Table 6: Statistics for fault tolerant workstation analysis from the analysis of property P_2

[5] Boudewijn R. Haverkort, Holger Hermanns, and Joost-Pieter Katoen. On the Use of Model Checking Techniques for Dependability Evaluation. In *SRDS*, pages 228–237, 2000.

[6] Andrew Hinton, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM: A Tool for Automatic Verification of Probabilistic Systems. In *TACAS*, volume 3920 of *Lecture Notes in Computer Science*, pages 441–444. Springer, 2006.

[7] Joost-Pieter Katoen, Ivan S. Zapreev, Ernst Moritz Hahn, Holger Hermanns, and David N. Jansen. The Ins and Outs of the Probabilistic Model Checker MRMC. In *QEST*, pages 167–176, 2009.

A Appendix

In the tables below we give the results of our comparison of *MRMC* 1.2, 1.3, 1.4, 1.5 and its predecessor *ETMCC*. The specialised algorithm [1] for maximal reachability probabilities over time-abstract, history-dependent schedulers was used with precision of 1E-6. As we can see, there is a huge performance improvement when changing from *ETMCC* to *MRMC*, and another significant improvement when changing from the previous to the latest version of *MRMC*. Notice that in certain previous publications result tables were shifted by one line, which explains the large differences to the table here.

Workst.	States		100h (tab)	100h (dep)	500h (tab)	500h (dep)
4	808	Time	0s	0s	0s	0s
		Prob.	0.0000770	0.0000770	0.0000770	0.0000770
8	2770	Time	0s	0s	0s	0s
		Prob.	0.0001635	0.0001635	0.0001635	0.0001635
16	10130	Time	0s	1s	0s	1s
		Prob.	0.0003483	0.0003483	0.0003483	0.0003483
64	151058	Time	12s	16s	13s	30s
		Prob.	0.0018185	0.0018185	0.0018187	0.0018187

Workst.	States		1000h (tab)	1000h (dep)	5000h (tab)	5000h (dep)
4	808	Time	0s	0s	0s	1s
		Prob.	0.0000770	0.0000770	0.0000770	0.0000770
8	2770	Time	0s	0s	0s	2s
		Prob.	0.0001635	0.0001635	0.0001635	0.0001635
16	10130	Time	0s	2s	1s	9s
		Prob.	0.0003483	0.0003483	0.0003526	0.0003526
64	151058	Time	13s	49s	11s	3m 23s
		Prob.	0.0018460	0.0018460	1.0000000	1.0000000

Table 7: Statistics for fault tolerant workstation analysis from the analysis of property P_3. We give both number for the time-abstract algorithm (tab) as well as for the time-dependent algorithm (dep)

Workst.	States		100h	500h	1000h	5000h
1	110	Prob.	0.0008828	0.0044933	0.0089880	0.0442228
		ETMCC	0s	0s	0s	0s
		1.2	0s	0s	0s	0s
		1.3	0s	0s	0s	0s
		1.4	0s	0s	0s	0s
		1.5	0s	0s	0s	0s
2	274	Prob.	0.0009394	0.0048464	0.0097087	0.0477613
		ETMCC	0s	0s	0s	0s
		1.2	0s	0s	0s	0s
		1.3	0s	0s	0s	0s
		1.4	0s	0s	0s	0s
		1.5	0s	0s	0s	0s
4	818	Prob.	0.0018491	0.0095426	0.0190762	0.0921146
		ETMCC	0s	0s	0s	0s
		1.2	0s	0s	0s	0s
		1.3	0s	0s	0s	0s
		1.4	0s	0s	0s	0s
		1.5	0s	0s	0s	0s
8	2770	Prob.	0.0037199	0.0191655	0.0381363	0.1772726
		ETMCC	0s	0s	0s	1s
		1.2	0s	0s	0s	1s
		1.3	0s	0s	0s	1s
		1.4	0s	0s	0s	0s
		1.5	0s	0s	0s	0s
16	10130	Prob.	0.0074551	0.0381323	0.0751490	0.3243474
		ETMCC	0s	0s	1s	2s
		1.2	0s	0s	1s	3s
		1.3	0s	0s	0s	2s
		1.4	0s	0s	0s	1s
		1.5	0s	0s	0s	0s
32	38674	Prob.	0.0143335	0.0720260	0.1394164	0.5291840
		ETMCC	1s	2s	3s	14s
		1.2	0s	1s	2s	9s
		1.3	0s	1s	2s	8s
		1.4	0s	0s	0s	2s
		1.5	0s	0s	0s	1s

Workst.	States		100h	500h	1000h	5000h
64	151058	Prob.	0.0251809	0.1228233	0.2312399	0.7324401
		ETMCC	3s	9s	14s	58s
		1.2	1s	5s	9s	40s
		1.3	1s	4s	7s	33s
		1.4	0s	1s	2s	8s
		1.5	0s	1s	1s	4s
128	597010	Prob.	0.0391070	0.1837937	0.3344111	0.8698468
		ETMCC	42s	1m 7s	1m 40s	5m 29s
		1.2	6s	21s	41s	3m 5s
		1.3	5s	18s	33s	2m 38s
		1.4	2s	6s	11s	48s
		1.5	1s	4s	7s	31s

Workst.	States		10000h	30000h	50000h
1	110	Prob.	0.0865099	0.2377584	0.3639644
		ETMCC	0s	0s	1s
		1.2	0s	0s	0s
		1.3	0s	0s	0s
		1.4	0s	0s	0s
		1.5	0s	0s	0s
2	274	Prob.	0.0932774	0.2546009	0.3872219
		ETMCC	0s	1s	1s
		1.2	0s	0s	1s
		1.3	0s	0s	1s
		1.4	0s	0s	0s
		1.5	0s	0s	0s
4	818	Prob.	0.1758129	0.4402362	0.6198248
		ETMCC	1s	2s	3s
		1.2	0s	1s	2s
		1.3	0s	1s	2s
		1.4	0s	0s	1s
		1.5	0s	0s	1s
8	2770	Prob.	0.3232411	0.6901539	0.8581406
		ETMCC	2s	6s	9s
		1.2	1s	4s	7s
		1.3	1s	4s	6s
		1.4	0s	1s	2s
		1.5	0s	1s	2s

Workst.	States		10000h	30000h	50000h
16	10130	Prob.	0.5436606	0.9050388	0.9802392
		ETMCC	5s	14s	23s
		1.2	5s	15s	24s
		1.3	4s	12s	21s
		1.4	1s	4s	6s
		1.5	1s	3s	4s
32	38674	Prob.	0.7784745	0.9891429	0.9994679
		ETMCC	26s	1m 16s	2m 7s
		1.2	18s	54s	1m 27s
		1.3	17s	48s	1m 18s
		1.4	4s	14s	23s
		1.5	3s	7s	12s
64	151058	Prob.	0.9284749	0.9996347	0.9999981
		ETMCC	1m 55s	5m 29s	9m 11s
		1.2	1m 20s	3m 53s	6m30s
		1.3	1m 5s	3m 4s	5m 10s
		1.4	16s	45s	1m 16s
		1.5	8s	25s	40s
128	597010	Prob.	0.9830755	0.9999952	1.0000000
		ETMCC	10m 18s	30m 10s	58m 19s
		1.2	6m 2s	18m 26s	30m 6s
		1.3	5m 8s	15m 28s	24m 15s
		1.4	1m 34s	4m 33s	7m 27s
		1.5	57s	2m 42s	4m 30s