

Soar Data Collection

Jonathan Voigt
31st Soar Workshop
June 2011



SOARTECH

Modeling human reasoning.
Enhancing human performance.

Motivation

- Stats accumulates values but not history
- Stats history useful for agent evaluation
 - What is the worst-case reactivity?
- Optimizing agents for long-running robot experiments

```
Soar 9.3.1 on coffee.aa.soartech.com at Tue Jun 14 10:07:10 2011

96 productions (72 default, 12 user, 12 chunks)
+ 0 justifications
```

Phases:	Input	Propose	Decide	Apply	Output	Computed Totals
Kernel:	0.000	0.000	0.003	2.909	0.000	2.912
Input fn:	0.000					0.000
Outpt fn:					0.000	0.000
Callbcks:	0.000	0.000	0.000	0.000	0.000	0.000
Computed-----						-----+
Totals:	0.000	0.000	0.003	2.909	0.000	2.912

```

Values from single timers:
Kernel CPU Time:      2.916 sec.
Total CPU Time:      2.920 sec.

3060 decisions (0.953 msec/decision)
9239 elaboration cycles (3.019 ec's per dc, 0.316 msec/ec)
9240 inner elaboration cycles
3058 p-elaboration cycles (0.999 pe's per dc, 0.953 msec/pe)
54070 production firings (5.852 pf's per ec, 0.054 msec/pf)
104816 wme changes (52512 additions, 52304 removals)
WM size: 208 current, 211.522 mean, 367 maximum

```

Approach

- Kernel changes to stats command: “max-cycle” stats collection
 - Time, production firings, working memory size
 - EpMem/SMem time each cycle
 - Efficient history storage
- External tool: Implemented in Java in SML
 - Idea: Flexible for easy integration with existing environments
- Focus on performance
 - Minimize expensive API calls, “uncertainty principle”
- Collect data rows in periods of decision cycles or elapsed time
- Output to spreadsheet-compatible format for easy inspection
 - CSV (comma separated values)

```
Single decision cycle maximums:
Stat      Value      Cycle
-----
Time (sec) 0.000000    0
EpMem Time (sec) 0.000000    0
SMem Time (sec) 0.000000    0
WM changes      0          0
Firing count    0          0
test:raw> |
```

Data

- Agent name and settings (e.g. what learning systems are on)
- Wall clock, Kernel times, CPU times
- Production firing counts and firing time
- Working memory element counts: max, mean, min, additions, removals
- Episodic and semantic memory: time, retrievals, queries, stores etc.

- Many statistics record maximums and means that reset on each call to collect
 - “How long did the slowest decision cycle take since I last checked?”

Example Data

agent	wall clock	dc num	kernel sec	avg msec/dc	cpu sec	pf total	average mse	wm current	wm mean	wm max
seek	10.009	23891	0.84496	0.035367	1.032217	24082	0.035087	454	391.517149	455

wm addition	wm removal	max dc time	max dc time	Episodic w/ft	max dc chan	max dc chan	max dc chan	max dc pf	cy	max dc pf	va
83833	83379	1	3155	3.155	1	1	272	1	22		

agent	wall clock	dc num	kernel sec	avg msec/dc	cpu sec	pf total	average mse	wm current	wm mean	wm max	wm addition	wm removal	max dc time	max dc time	Episodic w/ft	max dc chan	max dc chan	max dc chan	max dc pf	cy	max dc pf	va													
agent	10.009	23891	0.84496	0.035367	1.032217	24082	0.035087	454	391.517149	455	83833	83379	1	3155	3.155	1	1	272	1	22															
seek	20.008	47964	1.669812	0.034814	2.083903	48075	0.034026	914	429.238687	915	174563	174049	47272	955	0.955	42721	282	42722	21	2.04819	42779	0.000751	888456	38	0.000005	0.455079	42772	0.000739	888456	0	2	20	0.000001		
seek	30.009	71915	2.423637	0.031701	3.190607	73168	0.033124	1376	676.373548	1378	287454	288879	84144	1726	1.176	82104	326	84144	41	0.380902	84139	0.001366	888456	41	0.000005	0.819623	84144	0.001361	888456	1	11	48	0.000008		
seek	40.008	94711	3.171563	0.033668	4.452051	97136	0.032623	1916	913.678184	1918	429889	434146	86306	2356	1.356	83164	355	85997	43	0.487597	86006	0.0021299	888456	73	0.000005	0.792489	83919	0.002078	888456	2	17	75	0.000007		
seek	50.008	119716	3.947566	0.033976	5.486331	122597	0.0322	2645	1193.513109	2647	645 53153309	673	517033	516358	110359	641	0.641	110359	106	0.6181001	110359	0.002007	888456	81	0.000005	0.951361	1103160	0.004007	888456	2	17	76	0.000006		
seek	60.008	144960	4.800265	0.033123	6.521519	148838	0.032274	3405	1645.5173412	3407	645 55173412	673	622278	611633	137388	650	0.65	137388	89	0.758822	138959	0.0031884	888456	125	0.000006	1.138029	129697	0.003079	888456	2	17	76	0.000007		
seek	70.008	173482	5.651235	0.032623	7.979122	177549	0.031828	4297	2157.85188464	4299	597 56248664	722	717577	716980	158980	3016	3.016	157873	161	0.940807	159585	0.002954	888456	139	0.000005	1.322281	152783	0.003156	888456	4	26	115	0.000007		
seek	80.008	201276	6.529566	0.032441	8.709532	206880	0.031561	5176	2676.473375	5178	695 576473375	722	829761	829056	181462	740	0.74	181462	180	1.184855	37	0.168007	181462	0.002	888456	167	0.000005	1.50286	181463	0.002524	888456	4	26	116	0.000007
seek	90.008	229657	7.370476	0.032212	9.936224	235524	0.031294	6182	3365.8191087	6184	592 58131087	722	966324	964732	228963	2267	2.267	2292457	346	208462	58	1.205484	228963	0.002196	888456	178	0.000005	1.681733	228962	0.003062	888456	5	33	146	0.000006
seek	100.009	258601	8.180739	0.032015	11.08576	252313	0.031371	7299	4089.5839967	7301	669 58930967	722	107074	1069705	237599	723	0.723	237598	126	2.137523	237598	0.002215	888456	196	0.000005	1.932221	237599	0.002578	888456	5	33	147	0.000007		
seek	110.009	287550	9.969061	0.032541	12.977162	296119	0.03161	8599	4997.350508	8601	669 59735058	722	1277375	1276706	288692	470	0.47	278646	86	281490	6	1.512952	281489	0.000164	888456	232	0.000005	2.193423	286838	0.000688	888456	5	33	147	0.000009
seek	120.009	316607	10.91317	0.032562	15.572273	326005	0.031633	10197	5907.900271	10200	541 597900271	722	1848434	1848293	300139	1969	1.969	294296	307	380128	52	1.666449	300139	0.0191	919096	262	0.000005	2.407176	300128	0.000996	919096	6	39	173	0.000007
seek	130.009	345602	11.579601	0.032393	16.756033	352595	0.031486	12144	7095.983903	12146	516 5964903	722	1489360	1489247	373265	1765	1.765	339225	298	344385	44	1.80839	34492	0.001096	952856	281	0.000005	2.608812	339030	0.000929	952856	7	44	196	0.000007
seek	140.008	374351	12.164135	0.032446	18.187177	384145	0.031604	14488	8389.070027	14490	598 58870027	722	1493860	1493407	427365	630	0.63	373265	110	373272	74	1.958038	373272	0.000996	952856	285	0.000005	2.788374	373266	0.000447	952856	7	45	198	0.000007
seek	150.009	401515	13.092873	0.032609	17.160779	412613	0.031732	16982	9502.282579	16984	619 592282579	722	1616238	1615540	397415	801	0.801	381164	299	388644	56	2.120071	397428	0.000583	952856	359	0.000006	2.997533	388650	0.000444	952856	17	52	199	0.000008
seek	160.009	430730	14.037344	0.032591	18.318825	442557	0.031672	19999	10995.548664	20001	699 595548664	722	1722187	1722088	468085	417	0.417	468207	84	468133	64	2.281289	468005	0.000188	952856	372	0.000006	3.210863	469298	0.000078	952856	17	52	199	0.000007
seek	170.008	459558	14.958286	0.032561	19.442015	472058	0.031687	23997	12695.72982	24000	697 60572982	722	1824258	1823551	454444	443	0.443	450957	84	439015	6	2.444483	450488	0.000068	952856	403	0.000006	3.414573	449575	0.000047	952856	17	52	199	0.000007
seek	180.009	490001	15.958855	0.032459	20.764156	501414	0.031655	28233	15156.038279	28235	623 612038279	720	1960512	1960289	490001	2924	2.924	498985	312	489986	78	2.624667	490001	0.00285	950506	444	0.000006	3.641931	489986	0.001316	950506	18	60	233	0.000007
seek	190.009	524653	17.160755	0.032095	22.395642	548624	0.031828	33888	18348.664626	33890	688 6118664626	700	2118442	2112954	533917	782	0.782	533914	139	533919	32	2.817931	533914	0.000215	950506	446	0.000004	3.880328	533917	0.000085	950506	18	60	234	0.000006
seek	200.009	572529	18.800819	0.03231	23.818087	596908	0.031443	40916	21919.248785	40918	696 617297885	700	2256218	2254542	564422	1284	1.284	564267	326	564268	316	3.040389	564275	0.000684	950506	478	0.000005	4.203474	564272	0.001044	950506	18	63	245	0.000008
seek	210.008	615562	19.914012	0.032351	25.647579	631400	0.031539	49141	25324.20426	49143	641 61920426	700	2422563	2421922	576967	2643	2.643	576979	330	576880	62	3.285627	576967	0.002174	950506	480	0.000006	4.532041	576880	0.001504	950506	19	69	272	0.000008
seek	220.009	657445	21.956003	0.032484	27.86571	675602	0.031607	58262	30226.849955	58264	732 62649955	700	2576787	2573715	615588	840	0.84	615587	155	615590	40	3.501017	641588	0.000222	950506	519	0.000006	4.849507	615588	0.000053	950506	19	69	273	0.000008
seek	230.009	700511	22.818862	0.032375	29.111935	727189	0.031614	68146	36342.656208	68148	636 632656208	702	2720488	2719852	699496	2184	2.184	699186	349	699485	49	3.742314	699496	0.002025	952676	624	0.000006	5.1762	699181	0.00104	1062576	20	74	295	0.000008
seek	240.009	741105	24.547066	0.032383	30.712638	782512	0.031666	79339	43016.6888	79341	639 43016688	702	2861282	2860643	724158	797	0.797	724158	156	716935	8	3.942903	724128	0.000246	1062576	643	0.000005	5.484991	721217	0.000079	1062576	20	74	295	0.000007
seek	250.008	78073	25.956627	0.032696	32.480816	801734	0.031813	91268	52828.620477	91270	628 62820477	702	3003139	3003171	758223	607	0.607	758222	87	758227	13	4.139596	758222	0.002002	1062576	658	0.000005	5.814524	758223	0.00043	1062576	20	75	297	0.000009
seek	260.008	82453	26.881607	0.032004	34.252974	847486	0.031719	105124	60302.329294	105126	618 630239294	702	3201124	3200506	821384	3643	3.643	821387	343	821378	44	4.385141	821384	0.001575	1062576	699	0.000005	6.179872	821385	0.001183	1062576	22	83	322	0.000008
seek	270.008	870541	28.1793	0.03213	35.746154	897799	0.031528	124660	69294.88467	124662	660 62988467	702	3373714	3373683	862015	795	0.795	870516	90	862020	13	4.585441	862015	0.000366	1062576	707	0.000004	6.530388	862016	0.000511	1062576	22	84	324	0.000008
seek	280.009	916079	29.542505	0.032429	37.373587	940648	0.031495	145048	79337.181207	145050	740 633181207	702	3515632	3514892	877358	1007	1.007	889493	305	889498	25	4.810238	889693	0.000414	1062576	742	0.000005	6.850951	877358	0.000708	1062576	24	87	326	0.000007
seek	290.008	965646	30.885162	0.03215	39.216477	987133	0.031203	17408	87820.32053	17410	740 87820533	702	3697207	3696667	951014	504	0.504	923729	96	959939	6	5.030485	954765	0.000239	1062576	768	0.000005	7.152962	948999	0.000111	1062576	24	87	326	0.000007
seek	300.009	1001190	32.215284	0.032117	40.897787	1028863	0.031312	21399	96442.401417	21400	736 96442401417	702	3866522	3866053	967510	494	0.494	979227	88	967859	12														

Usage Details

- Single class Java library
- Instantiate once for all agents
- Call methods during three events:
 - Soar system start: onStart
 - Update (after all agents pass output): onUpdateEvent, collect
 - Soar system stop: onStop, collect
- Utility methods to help timing collection
 - Every n decision cycles: setPeriodCycles
 - Every n milliseconds: setPeriodMillis
- Flushing to file is expensive, doesn't happen while agents are running
 - Should explicitly call every so often in case of catastrophic failures

Nuggets/Coal

- Lots of work to
 - Support different output formats: sqlite, speedy
- Tricky to use because of the performance goal
- Successfully used to collect data during long runs
- Useful to optimize agents

Download

- Currently on github
 - <https://github.com/voigtjr/soar-datacollector>
- Will be moved in to SoarSuite