

Tablebases bremsen die Suche aus, das gilt als gesichertes Wissen. Stimmt bloß nicht! Nicht mehr, wie CSS-Tester Gerhard Sonnabend herausfand. Er speicherte die Fünfsteiner auf USB-Sticks, und der Unterschied zu Tablebases auf Festplatte ist derselbe wie zwischen Porsche und Polo!

Tablebases liefern perfekte Bewertungen, aber zu welchem Preis! Die Knotenzahlen der Programme brechen drastisch ein durch den Zugriff auf die riesigen Datenbanken, und die Engines können nicht so tief in den Suchbaum vordringen. Das liegt hauptsächlich an den verwendeten Datenträgern: zwar haben Festplatten in den letzten Jahren viel Geschwindigkeit zugelegt, sind aber im Vergleich beispielsweise zum Hauptspeicher dennoch schneckenlangsam, weil sie mechanisch arbeiten, also bewegte Teile enthalten. Eine Untersuchung hat ergeben, dass vor allem die Zugriffszeit das Killerkriterium bei Tablebase-Zugriffen ist, die Zeit also, die ein Laufwerk braucht,



Einer der verwendeten USB-Sticks

um eine bestimmte Information auf dem Datenträger zu suchen. Bei Festplatten hängt die Zugriffszeit ausschließlich von der Drehzahl ab und von der Geschwindigkeit, mit der der Schreib-/Lesekopf bewegt wird. Die ist bei verschiedenen Platten durchaus unterschiedlich, und es zeigte sich, dass Server-Platten spürbar schnelleren TB-Zugriff ermöglichen, was auch auf die Lösezeiten durchschlägt. Typische Zugriffszeiten von Festplatten bewegen sich zwischen 20 Millisekunden bei Notebook-Platten, 10 ms bei Desktop-Laufwerken und 4 ms bei Server-Boliden. Schneller geht es wegen der bewegten Teile kaum.

USB-Sticks enthalten keine bewegten Teile, weil sie aus Flash-Speicher bestehen. Der wird adressiert wie der Hauptspeicher auch und liefert seine Ergebnisse innerhalb einer einzigen Millisekunde zurück. Mittlerweile gibt es USB-Sticks mit vier GByte Kapazität im Handel; zwei davon reichen, um die kompletten Fünfsteiner zu lagern. Statt USB-Sticks kann man ebensogut CF-Karten verwenden, entweder in einem per USB angeschlossenen Kartenleser oder in einem PCMCIA-Adapter. Dass der TB-Zugriff damit schneller gehen müsste als bei Festplatten, war die Vermutung, die CSS-Tester Gerhard Sonnabend bewog, es einfach mal auszuprobieren. Wieviel schneller, sollte ein Experiment ermitteln. Von den Messergebnissen war nicht nur Gerhard sehr überrascht, sondern auch ein gewisser Datenträger-Experte aus der CSS-Redaktion.

Festplatte vs. Flash-Speicher

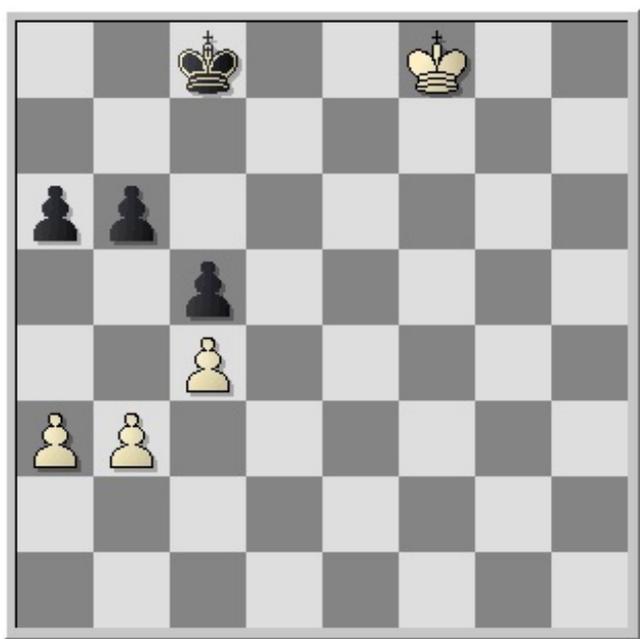
Referenz war eine Fujitsu-Notebookplatte MHU2100AT. Das Laufwerk schafft eine mittlere Dauertransferrate von 24 MByte pro Sekunde und eine mittlere Zugriffszeit von 20,35 Millisekunden. Alternativ wurden die Fünfsteiner auf zwei Vier-GByte-USB-Sticks gelagert, die mit einer Transferrate von 18,4 MByte pro Sekunde und einer mittlere Zugriffszeit von nur einer einzigen Millisekunde aufwarten. Desweiteren gab es Vergleichsmessungen mit einem älteren 2-GByte-USB-Stick, der ein Subset der Fünfsteiner enthielt, um herauszufinden, ob es signifikante Unterschiede zwischen supermodernen und älteren USB-Sticks gibt. Die Transferrate spielt bei Tablebase-Zugriffen absolut gar keine Rolle, weil das Laufwerk sie nur erreichen kann, wenn die zu lesenden Daten hübsch hintereinander auf der Platte liegen und ziemlich groß sind. Bei Tablebase-Zugriffen dagegen handelt es sich um klitzekleine Häppchen von nur vier oder acht KByte, die zudem von verschiedenen Stellen der Platte gelesen werden. Unter diesen Bedingungen läge die Dauertransferrate ohnehin nicht über fünf MByte pro Sekunde, eine Geschwindigkeit, die vor zehn Jahren schon als langsam galt.

Datenträger	mittlere Dauertransferrate [kByte/s]	mittlere Zugriffszeit [ms]
Festplatte MHU2100AT	24479	20,35
4-GByte-Sticks	18820	1,00
2-GByte-Stick	11764	2,43

Einzig die Zugriffszeit zählt bei Tablebase-Zugriffen, wie frühere Experimente gezeigt haben. Zu erwarten wäre also eine angesichts des großen Unterschiedes ziemlich deutliche Steigerung. Ebenso stellte sich heraus, dass der Tablebase-Cache völlig bedeutungslos ist, weil der Windows-Cache mindestens ebenso effektiv ist. Ohnehin spielt der Cache eine umso kleinere Rolle, je schneller das Medium die Daten liefert. Auch die USB-Schnittstelle ist locker schnell genug, wie CSS Online schon früher gezeigt hat.

Für die Experimente verwendete Gerhard die Engine Anaconda 2.0.1. Es zählt ja die Reproduzierbarkeit der Ergebnisse, nicht die reine Spielstärke, und bei Anaconda stimmten Hauptvariante, Knotenzahl sowie die Anzahl der TB-Zugriffe bei jeder Wiederholung überein, egal wo die Tablebases lagen, ob auf Platte oder USB-Stick – das Programm durchforstete also jedesmal exakt denselbe Suchbaum, nur eben mit unterschiedlicher Geschwindigkeit. Gerhard Sonnabend wiederholte jede Stellungsanalyse drei Mal und startete den Rechner zwischen zwei Versuchen jeweils neu, um absolut identische Bedingungen zu gewährleisten.

Die erste Teststellung erfordert unheimlich viele Tablebase-Zugriffe und ist überdies sogar recht hübsch:

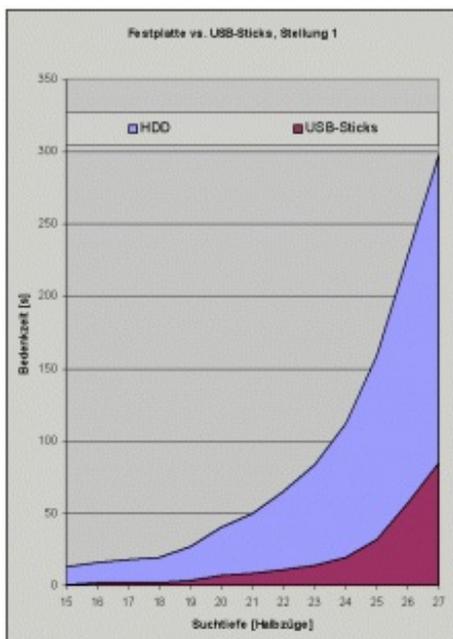


Weiß gewinnt ausschließlich mit 1.a4, während das von einigen Programmen ebenfalls für interessant gehaltene 1.Ke8 nur remisiert. Der schachliche Teil interessiert diesmal aber nicht, sondern nur der Vergleich zwischen Festplatte und USB-Stick.

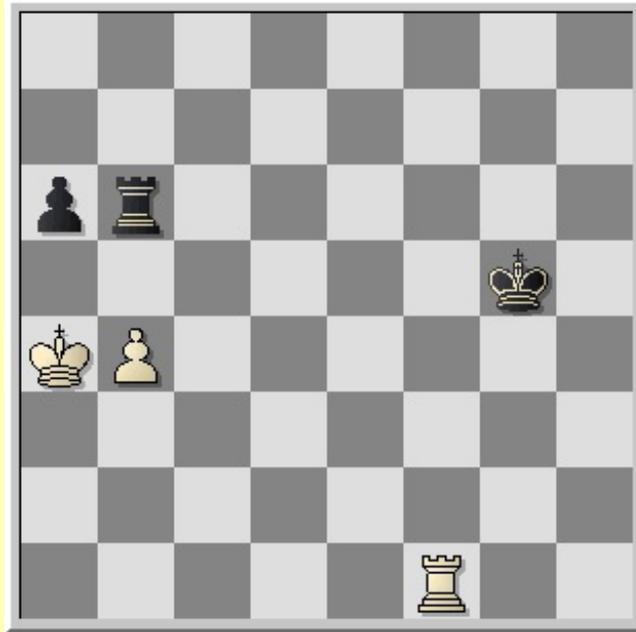
2k2K2/8/pp6/2p5/2P5/PP6/8/8 w - - 0 1						
Suchtiefe	Zeit [s]			TB-Hits	TB-Hits pro s Bedenkzeit	
	HDD	Sticks	Steigerung	Anzahl	HDD	Sticks
15/29	8	1	8,0	1.898	237	1898

15/41	13	1	13,0	3.772	290	3772
16/41	15	2	7,5	4.581	305	2291
16/41(2)	16	2	8,0	5.076	317	2538
17/41	18	2	9,0	6.104	339	3052
18/41	20	2	10,0	6.961	348	3481
19/41	27	4	6,8	12.161	450	3040
20/51	38	6	6,3	18.676	491	3113
20/51(2)	40	7	5,7	19.504	488	2786
21/57	50	9	5,6	28.796	576	3200
22/57	65	11	5,9	36.489	561	3317
23/57	84	14	6,0	50.050	596	3575
24/57	112	20	5,6	76.698	685	3835
25/57	159	32	5,0	138.584	872	4331
26/57	229	57	4,0	269.669	1178	4731
27/57	299	85	3,5	412.862	1381	4857
28/57	903	539	1,7	2.137.912	2368	3966

Die Tabelle zeigt einen dramatischen Geschwindigkeitsgewinn, besonders am Anfang. Für eine Suchtiefe von 18 Halbzügen rattert die Festplatte 20 Sekunden lang, während die USB-Sticks das Resultat bereits nach zwei Sekunden liefern! Nach etwas mehr als einer Minute Rechenzeit, genau: 65 Sekunden, hat Anaconda mit Tablebases auf Platte 22 Halbzüge tief gerechnet. Genau derselbe Spiel-baum wird in nur 11 Sekunden durchsucht, wenn die Tablebases auf den Sticks liegen! Oder, in andere Zahlen übersetzt: Anaconda berechnet fast sechsmal so viele Stellungen pro Sekunde mit den USB-Sticks bei Messpunkt 22 Ply; sogar Faktor 10 ergibt sich bei Messpunkt 18 Ply! Bei steigender Bedenkzeit und großer Suchtiefe fällt die Steigerung moderater aus, trotzdem schaffen die USB-Sticks eine Analyse in 85 Sekunden, für die die Festplatte drei Minuten braucht.



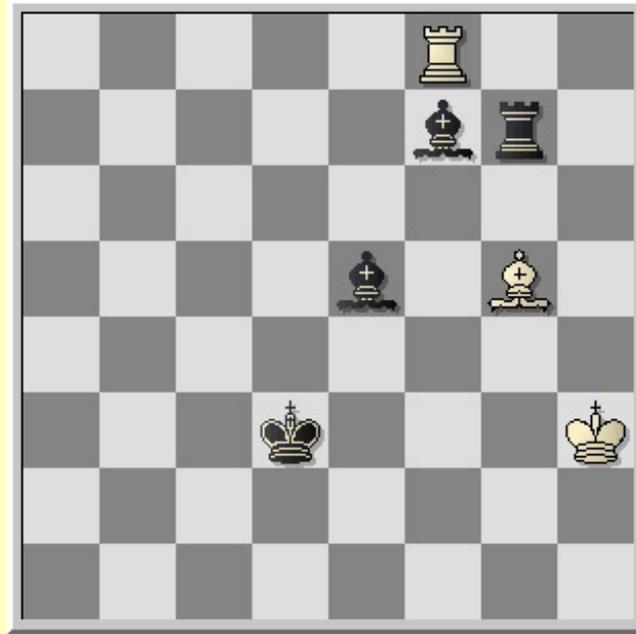
Auch in anderen Stellungen zeigt sich ein ähnliches Bild, hier die Resultate der weiteren Teststellungen:



8/8/pr6/6k1/KP6/8/8/5R2 w - - 0 1

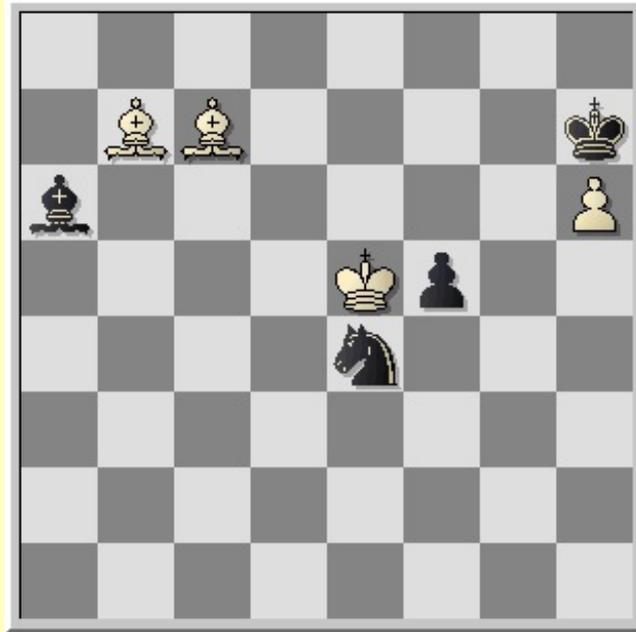
Suchtiefe	Zeit [s]			TB-Hits	TB-Hits pro s Bedenkzeit	
	HDD	Sticks	Steigerung	Anzahl	HDD	Sticks
8/16	5	1	5,0	2.799	560	2.799
8/19	6	1	6,0	3.555	593	3.555
8/19 (2)	6	1	6,0	3.760	627	3.760
9/19	8	1	8,0	5.416	677	5.416
10/21	14	2	7,0	11.257	804	5.629
11/21	18	3	6,0	15.690	872	5.230
11/23	23	4	5,8	22.868	994	5.717
12/23	24	4	6,0	25.798	1.075	6.450
12/23(2)	25	4	6,3	26.535	1.061	6.634
12/23(3)	26	4	6,5	29.638	1.140	7.410
13/23	30	5	6,0	35.869	1.196	7.174
13/23(2)	31	6	5,2	37.540	1.211	6.257
13/23(3)	32	6	5,3	41.248	1.289	6.875
14/23	33	7	4,7	46.863	1.420	6.695
15/25	37	8	4,6	54.561	1.475	6.820
15/26	42	9	4,7	61.978	1.476	6.886
16/32	62	14	4,4	102.680	1.656	7.334
16/32(2)	68	16	4,3	114.736	1.687	7.171
16/34	80	19	4,2	145.850	1.823	7.676
17/37	108	29	3,7	265.770	2.461	9.164
17/37(2)	110	30	3,7	273.291	2.484	9.110
18/37	123	34	3,6	329.471	2.679	9.690
18/37(2)	127	36	3,5	349.535	2.752	9.709
19/37	136	39	3,5	392.724	2.888	10.070
19/56	202	71	2,8	896.334	4.437	12.624
20/56	227	87	2,6	1.788.374	7.878	20.556
21/57	315	144	2,2	2.940.498	9.335	20.420

22/57	336	158	2,1	3.416.255	10.167	21.622
23/57	778	449	1,7	9.174.867	11.793	20.434



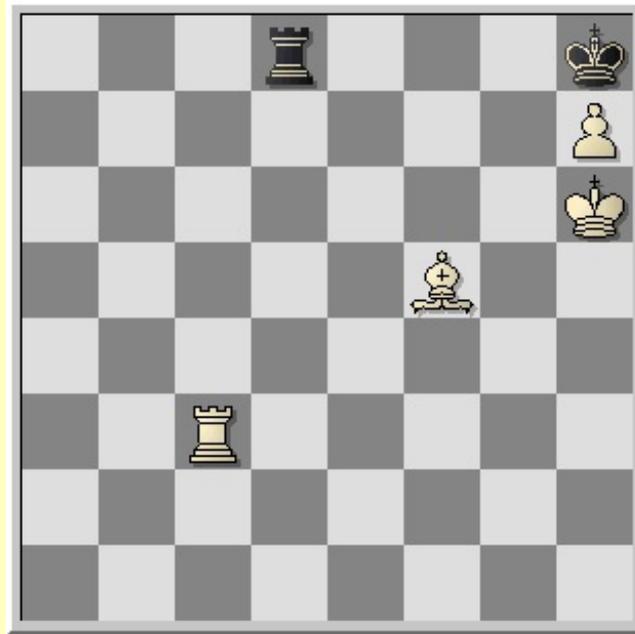
5R2/5br1/8/4b1B1/8/3k3K/8/8 b - - 0 1

Suchtiefe	Zeit [s]			TB-Hits	TB-Hits pro s Bedenzeit	
	HDD	Sticks	Steigerung	Anzahl	HDD	Sticks
8/22	8	1	8,0	4.759	595	4759
9/25	12	2	6,0	12.368	1031	6184
9/25(2)	15	3	5,0	18.508	1234	6169
9/26	16	4	4,0	22.060	1379	5515
10/27	17	4	4,3	25.048	1473	6262
10/29	18	5	3,6	27.449	1525	5490
10/29(2)	20	6	3,3	31.380	1569	5230
11/31	30	9	3,3	52.732	1758	5859
12/32	43	13	3,3	81.820	1903	6294
13/35	60	18	3,3	126.889	2115	7049
13/37	67	22	3,0	161.105	2405	7323
14/37	81	28	2,9	207.677	2564	7417
14/41	104	41	2,5	338.102	3251	8246
14/41(2)	151	74	2,0	677.128	4484	9150
14/41(3)	205	124	1,7	1.158.839	5653	9345
15/41	224	135	1,7	1.292.805	5771	9576
16/41	283	181	1,6	1.798.918	6357	9939
17/41	397	284	1,4	3.086.889	7776	10869



8/1BB4k/b6P/4Kp2/4n3/8/8/8 w - - 0 1

Suchtiefe	Zeit [s]			TB-Hits	TB-Hits pro s Bedenkzeit	
	HDD	Sticks	Steigerung	Anzahl	HDD	Sticks
10/20	7	1	7,0	3.981	569	3981
10/21	9	1	9,0	5.536	615	5536
11/22	12	2	6,0	8.122	677	4061
11/22(2)	15	2	7,5	9.608	641	4804
12/25	17	3	5,7	12.040	708	4013
12/45	18	3	6,0	13.082	727	4361
12/45(2)	27	5	5,4	21.349	791	4270
13/49	34	6	5,7	29.239	860	4873
13/50	38	7	5,4	34.031	896	4862
13/51	46	10	4,6	45.233	983	4523
13/57	80	24	3,3	148.211	1853	6175
14/57	81	25	3,2	149.195	1842	5968
15/57	82	25	3,3	151.506	1848	6060
16/57	84	25	3,4	155.087	1846	6203
17/57	100	29	3,4	181.942	1819	6274
18/57	151	41	3,7	303.215	2008	7395
19/57	250	67	3,7	521.238	2085	7780
20/57	400	118	3,4	1.132.646	2832	9599
21/57	554	175	3,2	1.936.196	3495	11064

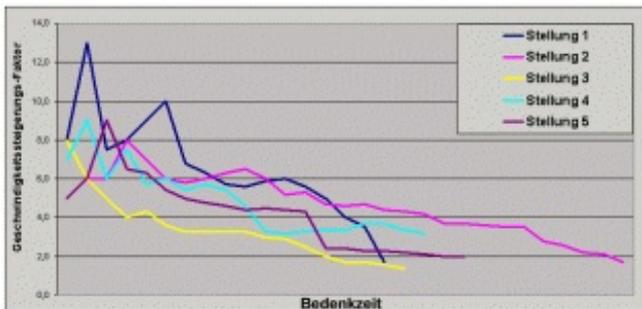


3r3k/7P/7K/5B2/8/2R5/8/8 w - - 0 1

Suchtiefe	Zeit [s]			TB-Hits	TB-Hits pro s Bedenzeit	
	HDD	Sticks	Steigerung	Anzahl	HDD	Sticks
11/18	5	1	5,0	5.096	1019	5096
12/18	6	1	6,0	7.445	1241	7445
13/22	9	1	9,0	11.093	1233	11093
14/26	13	2	6,5	19.741	1519	9871
15/26	19	3	6,3	31.271	1646	10424
16/30	27	5	5,4	51.082	1892	10216
17/35	35	7	5,0	75.192	2148	10742
18/36	48	10	4,8	120.529	2511	12053
19/38	64	14	4,6	178.674	2792	12762
19/38(2)	70	16	4,4	235.020	3357	14689
19/38(3)	72	16	4,5	239.184	3322	14949
19/38(4)	74	17	4,4	242.453	3276	14262
19/45	85	20	4,3	281.837	3316	14092
20/55	205	86	2,4	1.405.907	6858	16348
21/55	233	98	2,4	1.650.129	7082	16838
22/55	266	114	2,3	1.927.153	7245	16905
23/57	273	119	2,3	2.035.073	7454	17101
23/57(2)	421	191	2,2	3.960.010	9406	20733
23/57(3)	493	231	2,1	5.114.741	10375	22142
24/57	545	268	2,0	5.839.128	10714	21788
25/57	580	294	2,0	6.609.199	11395	22480

Leicht und schnell

Der Geschwindigkeitsgewinn wird kleiner, je länger die Engine rechnet. Das ist auch logisch, denn jedes Schachprogramm greift nur bis zu einer bestimmten Suchtiefe auf die Tablebases zu. Je tiefer die Engine in den Suchbaum vorstößt, desto seltener werden Tablebase-Zugriffe, und desto weniger kann sich der Geschwindigkeitsvorteil der Sticks auswirken. Auch die verschiedenen Zwischenspeicher spielen eine Rolle. Einmal der Hash-Speicher: je länger die Engine rechnet, desto besser gefüllt sind die Hashtables, die ohnehin so programmiert wurden, dass sie wichtige Stellungen, die viel Rechenzeit gekostet haben, länger bewahren. Auch die Datenträger-Caches tragen ihr Scherflein bei: die gelesenen Tablebase-Dateien landen zuerst im Windows- und dann im Tablebase-Cache. In beiden Fällen benötigt die Engine nur mehr einen Hauptspeicherzugriff, um an die begehrten Daten zu gelangen, und Hauptspeicher ist noch schneller als Flash-Speicher. Könnte man die Tablebases auf einer 8-GB-Byte-RAM-Disk lagern, ginge es noch schneller. Auch die Shredderbases, die ebenfalls komplett im RAM liegen, liefern noch schneller Resultate. Bei unendlich langer Analyse wäre also gar kein Geschwindigkeitsvorteil mehr zu erwarten – aber nur wenige Leute haben ja unendlich lange Zeit zum Analysieren. Alle anderen sparen Zeit, wenn sie die Tablebases auf USB-Sticks auslagern!



Am stärksten wirkt sich die Beschleunigung durch die USB-Sticks gegenüber einer Festplatte am Anfang der Rechnerei aus, wenn die verschiedenen Zwischenspeicher erst noch gefüllt werden müssen. Alle Daten müssen ja erst einmal in den Cache gelangen, was passiert, wenn sie einmal vom Medium gelesen wurden. Und genau da schlägt der Geschwindigkeitsvorteil am meisten auf die Rechenleistung durch.

Um den Cache-Einfluss zu untersuchen, hat Gerhard Sonnabend auch ein Kontrollexperiment veranstaltet. Dafür hat er den TB-Cache auf das Minimum von einem MByte eingestellt und den Windows-Cache fast völlig eliminiert, indem er den nach dem Start des Schachprogramms noch Speicher komplett belegt hat:

2k2K2/8/pp6/2p5/2P5/PP6/8/8 w - - 0 1					
Suchtiefe	Zeit [s]				TB-Hits
	HDD	HDD (kleiner Cache)	Sticks	Sticks (kleiner Cache)	Anzahl
15/29	8	7	1	1	1.898
15/41	13	13	1	1	3.772
16/41	15	14	2	2	4.581
16/41(2)	16	16	2	2	5.076
17/41	18	18	2	2	6.104
18/41	20	20	2	3	6.961
19/41	27	28	4	5	12.161
20/51	38	40	6	7	18.676
20/51(2)	40	42	7	8	19.504
21/57	50	54	9	10	28.796
22/57	65	73	11	13	36.489
23/57	84	105	14	17	50.050
24/57	112	160	20	25	76.698
25/57	159	235	32	40	138.584
26/57	229	328	57	70	269.669
27/57	299	433	85	106	412.862

Ganz ausschalten kann man den Cache-Einfluss natürlich nicht – einmal ist der Tablebase-Cache immer da, und Windows behält auch immer ein paar Speicherhäppchen für sich übrig. Dennoch, es ist einerseits zu sehen, dass der Cache für die langsamere Festplatte etwas wichtiger ist als für die Sticks, andererseits aber, dass auch die Sticks noch vom Cache profitieren. Also, liebe Kinder, immer schön Platz für den Windows-Filecache lassen, selbst dann, wenn die Tablebases auf USB-Sticks liegen!

Schnelle und langsame Sticks

Die im Test verwendeten Sticks gehören zu den schnellsten, die man momentan für Geld kaufen kann. Und sie sind nicht ganz billig. Für weniger Geld bekommt man kleinere und langsamere USB-Sticks. Um den Unterschied festzustellen, hat Gerhard Sonnabend auch noch einen alten 2-GByte-Stick mit den modernen Renn-Sticks verglichen, der eine mittlere Zugriffszeit von 2,4 Millisekunden hat, gegenüber einer Millisekunde bei den 4-GByte-Sticks. Natürlich passten nicht die kompletten Fünfsteiner auf den 2-GByte-Stick, sondern nur ein Subset. Dasselbe Subset hat Gerhard auch auf einen der schnellen Sticks und auf seine Festplatte kopiert und dann eine Stellung rechnen lassen.

8/5Q2/1K3P2/8/1k6/2n5/1p6/8 b - - 0 1					
Suchtiefe	Zeit HDD [s]	Zeit 2-Gbyte-Stick [s]	Zeit 4-Gbyte-Stick [s]	Steigerung (Sticks)	TB-Hits
6/19	7	1	1	1,00	2.249
7/23	13	3	2	1,50	6.427
8/27	17	5	4	1,25	16.533
9/31	28	13	11	1,18	60.588
10/31	37	17	16	1,06	90.396
11/31	49	22	20	1,10	115.056
12/40	84	42	38	1,11	243.891
13/40	130	67	61	1,10	415.721
14/40	224	128	118	1,08	868.899
15/40	363	231	217	1,06	1.685.850
16/42	817	627	607	1,03	4.925.328

Es zeigt sich, dass der Unterschied zwischen den „schnellen“ und den „langsamen“ USB-Sticks vernachlässigbar klein ist, zumindest im Vergleich zur Festplatte. Wer also keine so großen Sticks besitzt, kann wenigstens ein paar Tablebase-Dateien auf einen älteren Stick kopieren und trotzdem vom Geschwindigkeits-Vorteil profitieren, denn die Geschwindigkeitsangaben der Sticks beziehen sich nur auf die für Computerschächer ganz unwichtige Dauertransfertrate.

Fazit

Für jemanden, der öfter mal mit seinem Laptop Endspiele analysiert, lohnt es sich auf jeden Fall, die Tablebases auf Flash-Speicher auszulagern, denn der Geschwindigkeitsgewinn ist immens. Es müssen auch keine USB-Sticks sein, ebensogut eignen sich beispielsweise CF-Karten, die über einen Adapter am USB- oder PCMCIA-Anschluss hängen. Die Hauptsache ist, dass die Dinger mit Flash-Speicher arbeiten; CF-Karten gibt es auch mit Mini-Festplatten, und die sind natürlich nicht geeignet. In tablebasenahen Stellungen wird das Laptop bei Kurzanalysen unter einer Minute pro Stellung drei- bis zehnmal so schnell durch den Suchbaum rauschen, und auch, wenn man das Schachprogramm mal von der Leine und länger analysieren lässt, geht es meist noch doppelt so schnell.



Beide Sticks, voll mit all den Fünfsteinern, am Test-Notebook

Das gilt für den Vergleich von Flash-Speicher mit einer *aktuellen* Notebook-Festplatte – wer ein älteres Notebook mit einer noch langsameren Platte besitzt, wird noch mehr profitieren! Aber auch Analytiker, die einen Desktop-PC benutzen, haben etwas davon. Eine moderne IDE-Platte, egal ob Serial ATA oder Parallel ATA, hat eine mittlere Zugriffszeit um 10 Millisekunden. Halb so viel wie die 20 Millisekunden des getesteten Notebook-Laufwerks, aber doch noch sehr viel mehr als die eine einzige Millisekunde, mit welcher die USB-Sticks protzen. Auch hier wird der TB-Zugriff mindestens doppelt so schnell erledigt werden, bei sehr kurzer Bedenkzeit noch schneller. Und warum sollte man eine Minute auf eine Engine-Ausgabe warten, wenn man dasselbe Ergebnis in 20 Sekunden oder noch schneller haben kann?

Engine-Fans sollten ebenfalls mal nachdenken, ob sich die Sache für sie lohnen könnte, besonders wenn sie eine Engine auf irgendeinem Server blitzen lassen. Denn gerade in den ersten Sekunden des Denkprozesses einer Engine zeigt sich der größte Effekt. Die Tablebase-Bremse in

Engine-Matches war einmal!

Kurz gesagt: Der Zugriff geht unter allen Umständen schneller, wenn die Tablebases statt auf Festplatte auf USB-Sticks lagern. Wenn der Stick nicht groß genug ist, hilft es auch schon, nur einige der vielen Tablebase-Dateien darauf zu kopieren. Noch schneller als Flash wäre nur der Hauptspeicher, also RAM-Disk oder Shredderbases. (*Lars Bremer*)
