

Effiziente Programme

WS 2010/2011

Gruppe 111

Thema

- Sudoku Solver
 - Bill DuPree
 - http://www.techfinesse.com/game/sudoku_solver.php
 - Version: 1.20
 - LOC: ~ 2660

4		2		9		6	7	
1				5	2			9
	3				4		8	
	4	5				2	1	
			5	4				3
	6	8		7	1		4	
8					3			4
	7	3		8	5		9	
5			1				2	8

Sudoku - Algorithmus

- Initialisierung der Zellwerte mit allen möglichen Werten
- Eintragen der gegebenen Zellwerte

- Elimination von unerlaubten Möglichkeiten (Markup-Phase)
- Naked Single
- Naked Subset/Hidden Subset (zB 2-Tupel)
- Chute Reasoning
- Brute Force - Ausprobieren mittels Backtracing

Sudoku - Ausgangslage

- Testaufruf:
 - `papiex -e PAPI_TOT_CYC -e PAPI_TOT_INS -e PAPI_BR_MSP -e PAPI_FP OPS ./sudoku_solver -f input.txt -o output.txt -G`

- Ergebnis:

Cycles	3,31373e+08
Instructions Completed	3,61562e+08
Mispredicted Branches	8,64681e+06

Sudoku - Optimierung

- Schritt 01 (rev 2):
 - Compiler Optimierung umgestellt von -O zu -O3

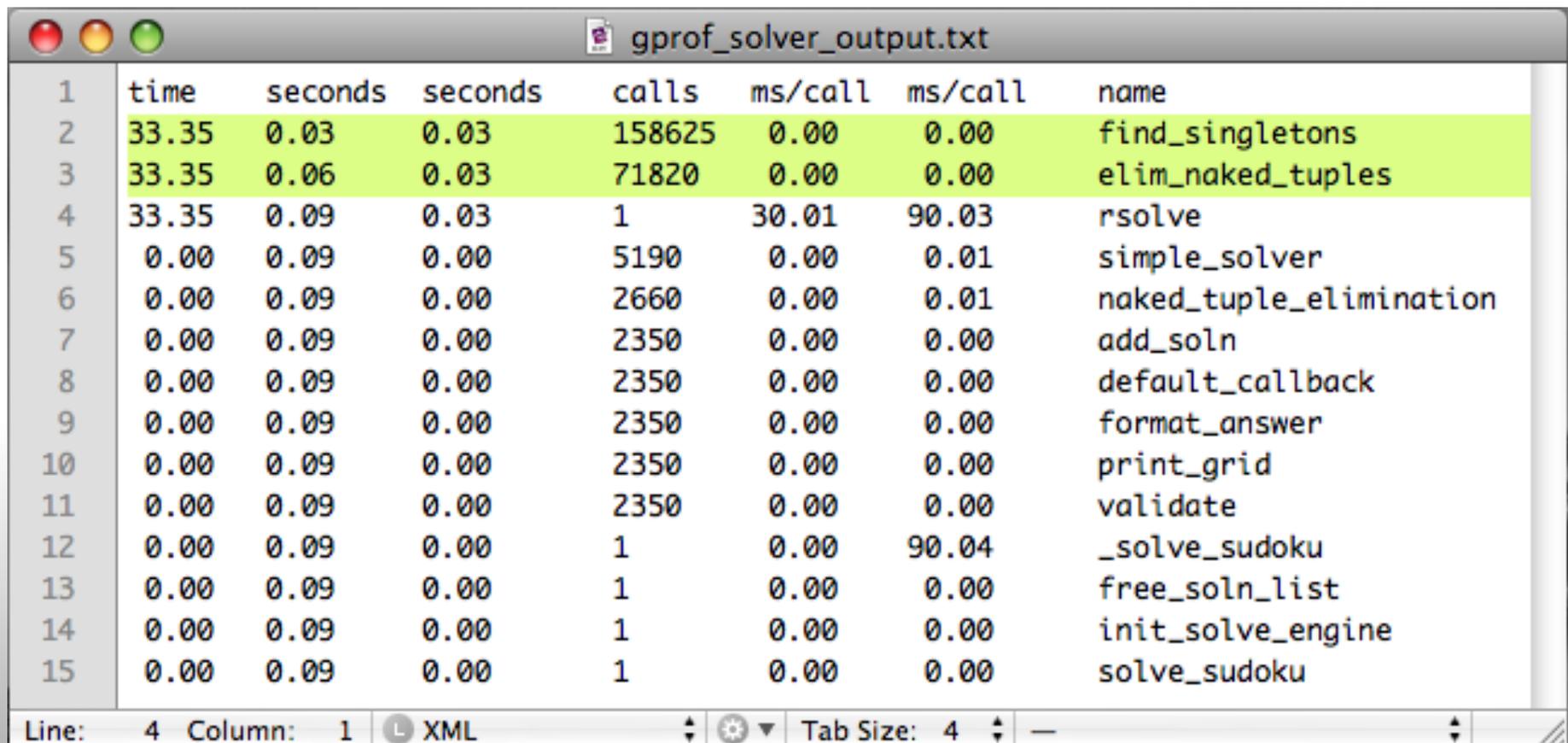
- Ergebnis:

Cycles	2.42975e+08 (-64.82%)
Instructions Completed	2.72182e+08 (-63.48%)
Mispredicted Branches	4.05635e+06 (-55.74%)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 02:
 - lohnende Methoden mit *gprof* und *gcov* finden



The screenshot shows a window titled "gprof_solver_output.txt" displaying the output of a gprof profiler. The output is a table with 8 columns: line number, time, seconds, seconds, calls, ms/call, ms/call, and name. The first two rows are highlighted in yellow.

1	time	seconds	seconds	calls	ms/call	ms/call	name
2	33.35	0.03	0.03	158625	0.00	0.00	find_singletons
3	33.35	0.06	0.03	71820	0.00	0.00	elim_naked_tuples
4	33.35	0.09	0.03	1	30.01	90.03	rsolve
5	0.00	0.09	0.00	5190	0.00	0.01	simple_solver
6	0.00	0.09	0.00	2660	0.00	0.01	naked_tuple_elimination
7	0.00	0.09	0.00	2350	0.00	0.00	add_soln
8	0.00	0.09	0.00	2350	0.00	0.00	default_callback
9	0.00	0.09	0.00	2350	0.00	0.00	format_answer
10	0.00	0.09	0.00	2350	0.00	0.00	print_grid
11	0.00	0.09	0.00	2350	0.00	0.00	validate
12	0.00	0.09	0.00	1	0.00	90.04	_solve_sudoku
13	0.00	0.09	0.00	1	0.00	0.00	free_soln_list
14	0.00	0.09	0.00	1	0.00	0.00	init_solve_engine
15	0.00	0.09	0.00	1	0.00	0.00	solve_sudoku

Line: 4 Column: 1 XML Tab Size: 4

Sudoku - Optimierung

- Schritt 03 (rev 3):
 - Entfernung unnötiger Variablen
 - Variablen, welche nur 1x zugewiesen und 1x verwendet wurden
 - Zuweisungscode direkt bei der Verwendung einfügen

- Ergebnis:

Cycles	2,24714e+08 (+0.02 %)
Instructions Completed	2,72760e+08 (+0.00 %)
Mispredicted Branches	4,89508e+06 (+0.03 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 04 (rev 4):
 - Initialisierung verbessern
 - Anfangs auf allen Feldern alle möglichen Zahlen (1-9) erlauben
 - Anschließend die gegebenen Felder eintragen

- Ergebnis:

Cycles	2,24579e+08 (-0.06 %)
Instructions Completed	2,72763e+08 (0.00 %)
Mispredicted Branches	4,89219e+08 (-0.06 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 05 (rev 5):
 - Loop Fusion
 - 3 Schleifen

```
/* Check rows */
for (i = 0; i < PUZZLE_DIM; i++) {
    for (rowmask = j = 0; j < PUZZLE_DIM; j++) {
        [...]
    }
}

/* Check columns */
for (i = 0; i < PUZZLE_DIM; i++) {
    for (colmask = j = 0; j < PUZZLE_DIM; j++) {
        [...]
    }
}
```

- Ergebnis:

Cycles	2,2477e+08 (+0.09 %)
Instructions Completed	2,7257e+08 (-0.07 %)
Mispredicted Branches	4,9083e+06 (+0.33 %)

- Entscheidung: **nicht übernehmen**

Sudoku - Optimierung

- Schritt 06 (rev 6):
 - "!=" -Vergleiche durch binäres XOR ersetzen

- Ergebnis:

Cycles	2,2510e+08 (+0.15 %)
Instructions Completed	2,7257e+08 (+0.00 %)
Mispredicted Branches	4,9187e+06 (+0.21 %)

- Entscheidung: **nicht übernehmen**

Sudoku - Optimierung

- Schritt 07 (rev 7):
 - Loop Fusion

```
/* Do rows (horizontal chutes) */  
for (i = 0; i < PUZZLE_DIM; i++) {  
    found |= find_singletons(g, row[i], "row");  
}  
  
/* Do columns (vertical chutes) */  
for (i = 0; i < PUZZLE_DIM; i++) {  
    found |= find_singletons(g, col[i], "column");  
}  
  
/* Do boxes */  
for (i = 0; i < PUZZLE_DIM; i++) {  
    found |= find_singletons(g, box[i], "box");  
}
```

- Ergebnis:

Cycles	2,2652e+08 (+0.63 %)
Instructions Completed	2,7427e+08 (+0.63 %)
Mispredicted Branches	4,8677e+06 (-1.04%)

- Entscheidung: **nicht übernehmen**

Sudoku - Optimierung

- Schritt 08 (rev 8):
 - Loop Unrolling

```
for (i = 0; i < PUZZLE_DIM; i++) {  
    found |= find_singletons(g, row[i], "row");  
}
```

```
found |= find_singletons(g, row[0], "row");  
found |= find_singletons(g, row[1], "row");  
found |= find_singletons(g, row[2], "row");  
found |= find_singletons(g, row[3], "row");  
found |= find_singletons(g, row[4], "row");
```

- Ergebnis:

Cycles	2,4209e+08 (+6,87 %)
Instructions Completed	2,7195e+08 (-0,84 %)
Mispredicted Branches	4,1428e+06 (-14,89 %)

- Entscheidung: **nicht übernehmen**

Sudoku - Optimierung

- Schritt 09 (rev 9):
 - Vergleich auf 0 (`x == 0`) durch `!x` ersetzen

- Ergebnis:

Cycles	2,42476e+08 (+0.16 %)
Instructions Completed	2,71957e+08 (+0.00 %)
Mispredicted Branches	4,15385e+06 (+0.27 %)

- Entscheidung: **nicht übernehmen**

Sudoku - Optimierung

- Schritt 10 (rev 10):
 - Loop Unrolling

```
for (i = 0; i < PUZZLE_DIM; i++) {  
    rc |= elim_naked_tuples(g, row[i], "row", i);  
}
```

```
rc |= elim_naked_tuples(g, row[0], "row", 0);  
rc |= elim_naked_tuples(g, row[1], "row", 1);  
rc |= elim_naked_tuples(g, row[2], "row", 2);  
rc |= elim_naked_tuples(g, row[3], "row", 3);
```

...

- Ergebnis:

Cycles	2,51479e+08 (+3,71 %)
Instructions Completed	2,72072e+08 (+0,04 %)
Mispredicted Branches	4,14016e+06 (-0,33 %)

- Entscheidung: **nicht übernehmen**

Sudoku - Optimierung

- Schritt 11 (rev 11):
 - Hilfsarray, welches Elemente einer Schleifeniteration mitspeicherte, entfernt

- Ergebnis:

Cycles	2,10375e+08 (-16,34 %)
Instructions Completed	2,70463e+08 (-0,59 %)
Mispredicted Branches	4,15139e+06 (+0.27 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 12 (rev 12):
 - Naked-Single suchte immer über alle Zahlen (1-9)
 - Speicherung aller bereits gefundenen Zahlen pro Reihe/Spalte/Box
 - Somit müssen nur noch die noch nicht gefundenen Zahlen durchsucht werden
- Ergebnis:

Cycles	1,61792e+08 (-23,09 %)
Instructions Completed	2,25948e+08 (-16,46 %)
Mispredicted Branches	2,77091e+06 (-33,25 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 13 (rev 15):
 - Verbesserung von Schritt 12 auch hier anwendbar

```
/* Compute bitmap of all possible candidates for this list of unsolved
cells. */
for (j = totalmask = 0; j < PUZZLE_DIM; j++) {
    if (g->cellflags[cell_list[j]] == UNSOLVED)
        totalmask |= g->cell[cell_list[j]];
}
```

- Ergebnis:

Cycles	1,5733e+08 (-2,76 %)
Instructions Completed	2,2330e+08 (-1,17 %)
Mispredicted Branches	2,6684e+06 (-3,70 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 14 (rev 16):
 - Variable i entfernt in For-Loop, aufgrund vorheriger Optimierungen.

- Ergebnis:

Cycles	1,48661e+08	(-5,51 %)
Instructions Completed	2,04247e+08	(-8,54 %)
Mispredicted Branches	2,66896e+06	(+0,02 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 15 (rev 21):
 - Downcount für Schleifen und verschachtelte Schleifen in `box_col_chute_elim` und `box_row_chute_elim`

```
for(i = 0; i < PUZZLE_DIM; i++)  
for(i = PUZZLE_DIM; i >= 0; i--)
```

- Ergebnis:

Cycles	1,46506e+08 (-1,45 %)
Instructions Completed	2,04234e+08 (-0,01 %)
Mispredicted Branches	2,53061e+06 (-5,18 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 16 (rev 23):
 - Ein weiterer Downcount einer For-Schleife

- Ergebnis:

Cycles	1,4457e+08 (-1,32 %)
Instructions Completed	2,0145e+08 (-1,36 %)
Mispredicted Branches	2,5141e+06 (-0,65 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 17 (rev 24):
 - Downcount in rsolv Funktion.

- Ergebnis:

Cycles	1,41559e+08 (-2,09 %)
Instructions Completed	1,99888e+08 (-0,78 %)
Mispredicted Branches	2,38336e+06 (-0,52 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 18 (rev 27):
 - Berechnung von Masken nicht in 3 separaten For-Schleifen, sondern in einer einzigen.

- Ergebnis:

Cycles	1,41331e+08 (-0,16 %)
Instructions Completed	1,98072e+08 (-0,91 %)
Mispredicted Branches	2,41515e+06 (+1,33 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 19 (rev 28):
 - Lineare Suche nach Cell mit wenigsten möglichen Kandidaten durch Downcount verbessert.

- Ergebnis:

Cycles	1,1738e+08 (-16,94%)
Instructions Completed	1,6634e+08 (-16,02 %)
Mispredicted Branches	1,9763e+06 (-18,17 %)

- Entscheidung: **übernehmen**

Sudoku - Optimierung

- Schritt 20 (rev 31):
 - Strategie entfernen führt zu starker Verbesserung der Cycles.
 - Unsere Testfälle wurden weiterhin erfolgreich gelöst.
 - Für alle möglichen Probleminstanzen gültig?

- Ergebnis:

Cycles	0,93312e+08 (-21,51 %)
Instructions Completed	1,33724e+08 (-19,61 %)
Mispredicted Branches	1,56315e+06 (-20,91 %)

- Entscheidung: **übernehmen?**

Sudoku - Optimierung

- Conclusio:

- Vergleich mit Ausgangsversion (gcc -O)

Cycles	-64,58 %
Instructions Completed	-53,99 %
Mispredicted Branches	-77,14 %

- Vergleich mit Ausgangsversion (gcc -O3)

Cycles	-47,75 %
Instructions Completed	-39,02 %
Mispredicted Branches	-59,61 %