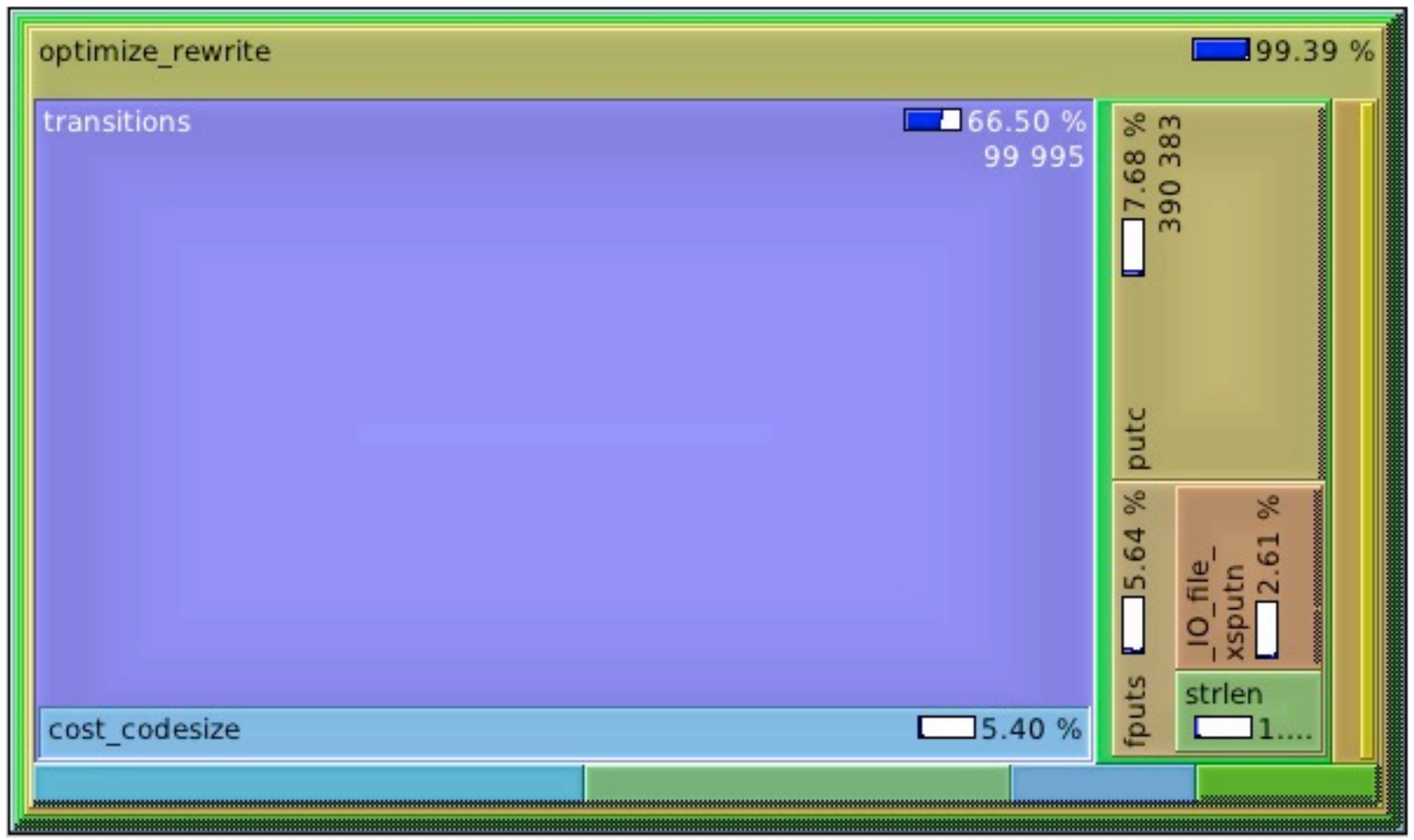
g103

Mallinger Bernhard
Demel Harald
Holzeis Richard
Aigner Erik

Baseline g0



- gcc -O1
- ~ 152.500.000 cycles

Reorder

```
PrimNum s = l->super;
int jcost;
struct cost *c=super_costs+s;
struct waypoint *wi=&(trans[c->state_in]);
struct waypoint *wo=&(inst[c->state_out]);
if (wo->cost == INF_COST)
  continue;
struct waypoint *wi=&(trans[c->state_in]);
jcost = wo->cost + ss_cost(s);
if (jcost <= wi->cost) {
```

- ~ 150.300.000 cycles
- -1,44%

Removed Pointer

```
static int ss_cost(int prim)
static int cost_codesize(int prim)
{
  return priminfos[prim].length;
}

typedef int Costfunc(int);
Costfunc *ss_cost =  /* cost function for optimize_bb */
cost_codesize;
```

- ~ 146.900.000 cycles
- -2,26%

Inline transitions()

```
transitions(inst[ninsts],trans[ninsts]);
transitions(inst[ninsts],trans[ninsts]);
{
   int k;
   struct super_state *l;
   struct waypoint *inst_transitions = inst[ninsts];
   struct waypoint *trans_transitions = trans[ninsts];
   .
   .
   .
}
```

- ~ 143.900.000 cycles
- -2,04%

ss_cost() macro

```
static int ss_cost(int prim)
{
  return priminfos[prim].length;
}
#define ss_cost( prim ) priminfos[prim].length
```

- ~ 135.500.000 cycles
- -5,83%

Defer Init

```
PrimNum s = l->super;
int jcost;
struct cost *c=super_costs+s;
struct waypoint *wo=&(inst_transitions[c->state_out]);
if (wo->cost == INF_COST)
    continue;
struct waypoint *wi=&(trans_transitions[c->state_in]);
jcost = wo->cost + ss_cost(s);
int jcost = wo->cost + ss_cost(s);
if (jcost <= wi->cost) {
    wi->cost = jcost;
```

- ~ 135.200.000 cycles
- -0,22%

Double Size Hashmap

```
#define HASH_SIZE 256
#define HASH_SIZE 512
```

- ~ 131.800.000 cycles
- -2,51%

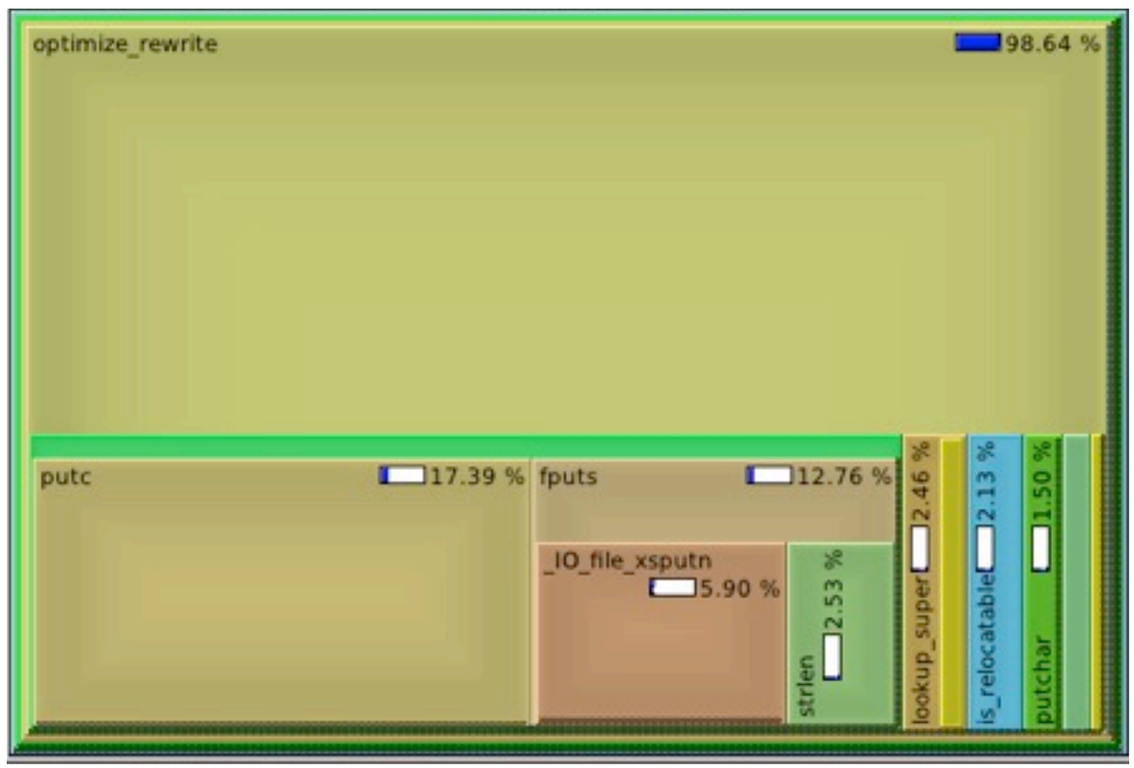
optimize_rewrite cache

caching calculations with hashmap in optimize_rewrite()

- ~ 79.100.000 cycles
- -39,98%

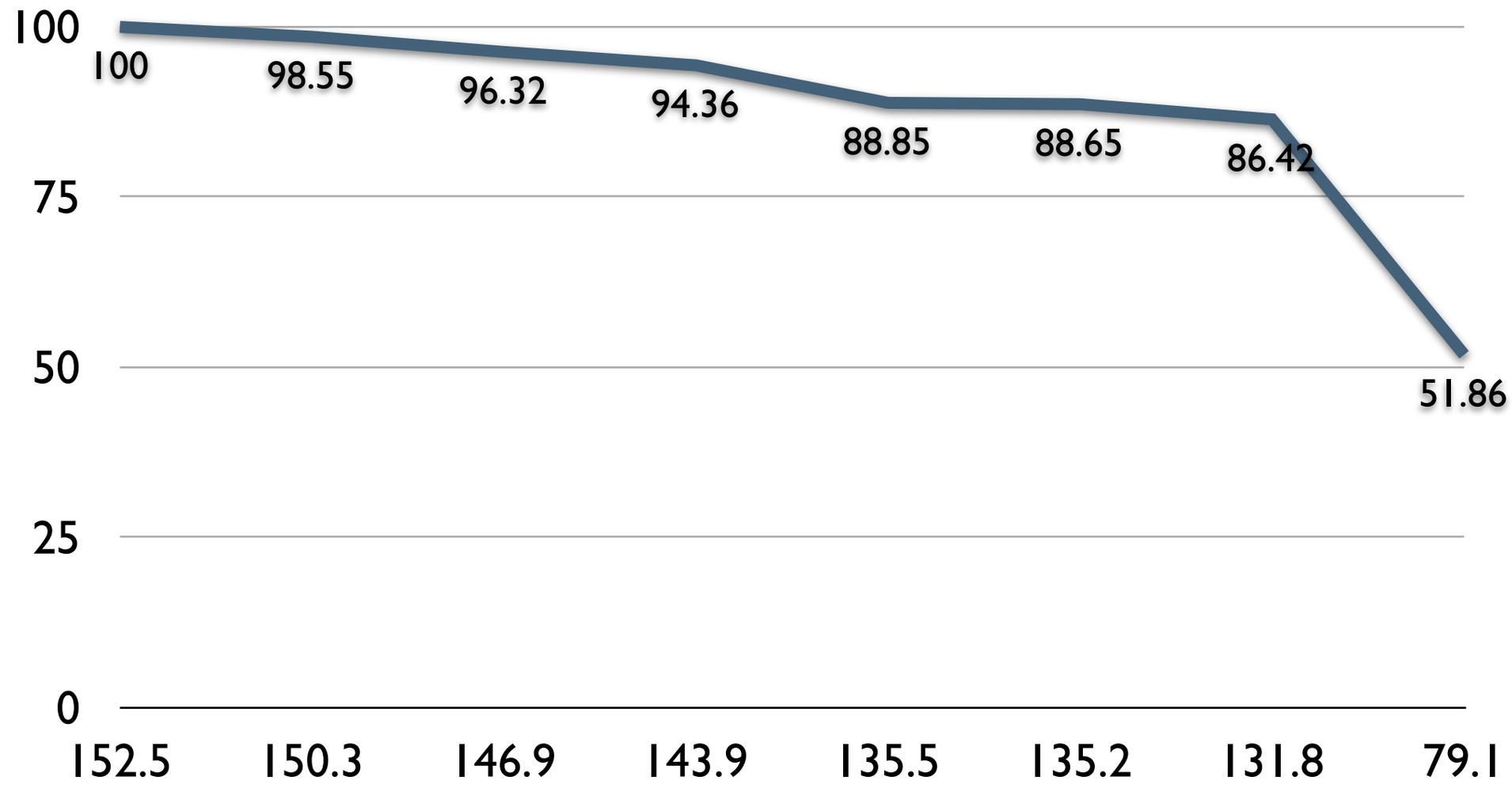
Profile

Baseline



Optimized

Chart


100
75
50
25
0
100
98.55
96.32
94.36
88.85
88.65
86.42
51.86
152.5
150.3
146.9
143.9
135.5
135.2
131.8
79.1

-O3

- ~ 74.000.000
- -6,44%