Ethan Schreiber and Rosa Vitiello – Profiling Data

OS: Red Hat Enterprise Linux Workstation release 7.7 (Maipo)
Model: Intel(R) Core(TM) i7-6700 CPU @ 3.4GHz
gcc version: 9.1.0

Benchmark	Time	Instructions	Rel to Start	Rel to Prev	Change
Sandmark Adventure Midmark	75.685s 27.07s 3.05s	- - 28.79 billion	1 1 1	- - -	None
Sandmark Adventure Midmark	53.81s 19.42s 2.18s	- - 23.75 billion	0.710 0.717 0.715	0.710 0.717 0.715	Compiled with -O1 and linked against -lcii40-O1
Sandmark Adventure Midmark	53.87s 18.13s 2.05s	- - 23.35 billion	0.712 0.669 0.672	1.001 0.934 0.940	Compiled with -O2 and linked against -lcii40-O2
Sandmark Adventure Midmark	43.88s 15.63s 1.78s	- - 20.42 billion	0.580 0.577 0.584	0.815 0.862 0.868	Remove function calls: static inline all functions with optimization
Sandmark Adventure Midmark	23.68s 7.75s 0.96s	- - 8.1 billion	0.313 0.286 0.315	0.540 0.496 0.539	Ran kcachegrind, found that Bitpack_getu taking the most time. Static inline bitpack function (remove library and use solutions implementation of bitpack)
Sandmark Adventure Midmark	8.09s 2.42s 0.32s	- - 4.2 billion	0.106 0.089 0.105	0.342 0.312 0.333	Ran kcachegrind, found that Seq stuff was taking the most time. Used dynamic C array instead of Sequence for main memory.
Sandmark Adventure Midmark	7.81s 2.37s 0.31s	- - 4.1 billion	0.103 0.088 0.102	0.965 0.979 0.969	Continued pointer chasing improvements by changing pointer register array to a standard global C array of size 8