Chapter 1

Experiments and Results

All the experiments were executed on a machine with Intel core i7 processor, with 3.40 GHz CPU, Ubuntu 14.04 and 16 GB RAM. In all the experiments we have taken execution time of programs as our performance metric.

In our first set of experiments, we have tested the performance of Superblock scheduling, done using Edge profile and Path profile. We have also tested the effect of function inlining on superblock scheduling. We have written different LLVM passes for superblock formation using edge profile and path profile. After superblock formation we have optimized our code using LLVM O2 optimization. We have tested our results against the run time of programs optimized by LLVM O2 optimization.

In our second set of experiments, we have tested our framework for input-space clustering of programs. We found out hot functions of programs, and then produced path profile vectors (PPV) of different calls of those functions. We then performed clustering on these PPVs and made copies of the functions, as many as the number of clusters we get of their PPVs. We then perform superblock scheduling according to the representative path profile vectors of functions. After this we use LLVM O2 optimization to optimize the code.

In all the experimental results, we have reported execution time of programs in seconds. Every reading is mean of five executions and with every reading we have

also reported standard deviation of all five executions below them inside brackets. We have used *time* command to get the runtime of programs, and we have reported the user time.

1.1 Benchmarks

We have used SPEC CINT2000 [**?**] suite's nine programs to evaluate the performance of our experiments :

- 164.gzip
- 175.vpr
- 181.mcf
- 186.crafty
- 197.parser
- 253.perlbmk
- 255.vortex
- 256.bzip2
- 300.twolf

Gzip and bzip2 are data compression programs, vpr is a routing program, mcf gives a solution of single-depot vehicle scheduling in public transportation companies. Crafty is a chess playing program, parser is a syntactic parser of English, perlbmk is a cut-down version of a popular scripting language Perl v5.005. Vortex is a single-user object oriented database transaction benchmark, twolf is used in production of microchips for creating the lithography artwork. The remaining programs of SPEC CINT2000 could not be used because of some compilation issues with clang-LLVM on our system.

1.2 Superblock Scheduling

We profiled the programs on different inputs (test, train and ref inputs), created superblocks and then ran the program on all of those inputs. We have also shown the results of superblock formation with inlining. In Table 5.1 to 5.18 we have compared execution time of programs compiled through different ways:

- Normal : Compiling them directly with gcc.
- O2 : Compiling them with O2 optimization.
- Edge : First superblock formation is done using Edge Profile on profiling input, and then compiled with gcc along with O2 optimization.
- Path : First superblock formation is done using Path Profile on profiling input, and then compiled with gcc along with O2 optimization.

An entry in Edge and Path column, gives the run time of program, profiled on the input given in the cell along with Edge and Path. Profiling and execution are done on various inputs executed through their standard commandlines, which are shown in tables as :

- I1 : all test inputs
- I2 : all train inputs
- I3 : all ref inputs

In these results we see that, in most cases, we get slightly better results with path profiling as compared to edge profiling, and function inlining also improves the run time of programs. All the results are also available on :
http://www.cse.iitk.ac.in/users/rishdev/docs/results.pdf

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.059	0.037	0.036	0.034	0.036	0.037	0.035	0.036
	(0.001)	(0.001)	(0.001)	(0.003)	(0.002)	(0.002)	(0.002)	(0.003)
I2	4.984	3.365	3.321	3.324	3.329	3.343	3.310	3.312
	(0.007)	(0.015)	(0.010)	(0.021)	(0.014)	(0.009)	(0.011)	(0.016)
I3	38.936	27.257	27.181	27.099	27.089	27.157	27.124	27.041
	(0.293)	(0.123)	(0.252)	(0.028)	(0.084)	(0.036)	(0.054)	(0.081)

Table 1.1: Results of superblock scheduling without inlining on mcf

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.059	0.037	0.034	0.034	0.035	0.035	0.034	0.036
	(0.001)	(0.001)	(0.001)	(0.001)	(0.002)	(0.002)	(0.003)	(0.002)
I2	4.984	3.365	3.107	3.096	3.109	3.187	3.143	3.201
	(0.007)	(0.015)	(0.011)	(0.014)	(0.012)	(0.011)	(0.016)	(0.019)
I3	38.936	27.257	26.981	26.976	26.967	26.912	26.899	26.875
	(0.293)	(0.123)	(0.097)	(0.045)	(0.061)	(0.045)	(0.053)	(0.058)

Table 1.2: Results of superblock scheduling with inlining on mcf

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.170 (0.003)	0.115 (0.004)	0.113 (0.002)	0.114 (0.003)	0.111 (0.002)	0.108 (0.003)	0.109 (0.004)	0.112 (0.006)
I2	3.795 (0.026)	2.523 (0.044)	2.469 (0.023)	2.481 (0.032)	2.493 (0.010)	2.473 (0.007)	2.479 (0.011)	2.496 (0.021)
I3	33.283 (0.589)	22.970 (0.177)	22.936 (0.179)	22.739 (0.052)	22.636 (0.041)	22.419 (0.021)	22.688 (0.022)	22.517 (0.063)

Table 1.3: Results of superblock scheduling without inlining on vpr

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.170 (0.003)	0.115 (0.004)	0.105 (0.003)	0.106 (0.003)	0.106 (0.002)	0.105 (0.003)	0.106 (0.002)	0.105 (0.003)
I2	3.795 (0.026)	2.523 (0.044)	2.418 (0.008)	2.392 (0.008)	2.395 (0.009)	2.372 (0.016)	2.371 (0.022)	2.401 (0.022)
I3	33.283 (0.589)	22.970 (0.177)	22.307 (0.131)	21.910 (0.181)	21.889 (0.046)	21.915 (0.062)	21.748 (0.113)	21.824 (0.104)

Table 1.4: Results of superblock scheduling with inlining on vpr

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.677	0.444	0.447	0.450	0.448	0.448	0.454	0.457
	(0.005)	(0.004)	(0.002)	(0.002)	(0.003)	(0.003)	(0.001)	(0.003)
I2	4.089	2.625	2.691	2.694	2.673	2.701	2.725	2.747
	(0.005)	(0.009)	(0.002)	(0.005)	(0.002)	(0.007)	(0.002)	(0.005)
I3	28.358	18.698	19.298	19.382	19.316	19.240	19.534	19.526
	(0.032)	(0.010)	(0.024)	(0.025)	(0.026)	(0.013)	(0.020)	(0.021)

Table 1.5: Results of superblock scheduling without inlining on crafty

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.677	0.444	0.437	0.433	0.432	0.444	0.447	0.450
	(0.005)	(0.004)	(0.001)	(0.003)	(0.002)	(0.004)	(0.003)	(0.003)
I2	4.089	2.625	2.578	2.561	2.584	2.637	2.654	2.672
	(0.005)	(0.009)	(0.002)	(0.007)	(0.008)	(0.008)	(0.009)	(0.011)
I3	28.358	18.698	18.461	18.320	18.357	18.756	19.062	19.056
	(0.032)	(0.010)	(0.020)	(0.021)	(0.028)	(0.026)	(0.025)	(0.027)

Table 1.6: Results of superblock scheduling with inlining on crafty

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.954	0.486	0.480	0.488	0.480	0.479	0.492	0.481
	(0.005)	(0.006)	(0.004)	(0.003)	(0.006)	(0.005)	(0.007)	(0.003)
I2	2.521	1.505	1.505	1.499	1.495	1.492	1.509	1.497
	(0.008)	(0.006)	(0.007)	(0.005)	(0.001)	(0.011)	(0.007)	(0.005)
I3	102.010	63.357	63.697	63.969	63.421	62.998	63.625	63.202
	(0.121)	(0.063)	(0.138)	(0.155)	(0.068)	(0.102)	(0.084)	(0.083)

Table 1.7: Results of superblock scheduling without inlining on parser

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.954	0.486	0.463	0.465	0.462	0.470	0.469	0.469
	(0.005)	(0.006)	(0.008)	(0.005)	(0.005)	(0.004)	(0.008)	(0.005)
I2	2.521	1.505	1.406	1.393	1.398	1.431	1.423	1.430
	(0.008)	(0.006)	(0.003)	(0.004)	(0.004)	(0.006)	(0.007)	(0.009)
I3	102.010	63.357	59.763	59.155	59.500	60.399	60.423	60.296
	(0.121)	(0.063)	(0.080)	(0.031)	(0.042)	(0.023)	(0.033)	(0.041)

Table 1.8: Results of superblock scheduling with inlining on parser

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.071	0.052	0.050	0.052	0.053	0.049	0.051	0.054
	(0.006)	(0.002)	(0.008)	(0.005)	(0.004)	(0.007)	(0.004)	(0.003)
I2	3.472	2.049	2.027	2.020	2.027	2.005	1.981	2.019
	(0.005)	(0.015)	(0.014)	(0.014)	(0.037)	(0.016)	(0.019)	(0.020)
I3	62.637	32.760	33.841	33.251	33.225	32.237	34.462	33.481
	(0.046)	(0.061)	(0.047)	(0.146)	(0.095)	(0.098)	(0.035)	(0.052)

Table 1.9: Results of superblock scheduling without inlining on perlbmk

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.071	0.052	0.047	0.053	0.054	0.048	0.055	0.052
	(0.006)	(0.002)	(0.003)	(0.004)	(0.005)	(0.006)	(0.006)	(0.005)
I2	3.472	2.049	2.024	2.012	2.031	1.991	1.978	1.998
	(0.005)	(0.015)	(0.015)	(0.016)	(0.011)	(0.015)	(0.024)	(0.021)
I3	62.637	32.760	33.578	33.512	33.531	32.134	34.198	33.022
	(0.046)	(0.061)	(0.033)	(0.061)	(0.058)	(0.027)	(0.041)	(0.038)

Table 1.10: Results of superblock scheduling with inlining on perlbmk

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	2.513	0.992	0.954	0.956	0.957	0.952	0.955	0.954
	(0.008)	(0.007)	(0.005)	(0.006)	(0.005)	(0.005)	(0.003)	(0.004)
I2	3.429	1.344	1.314	1.308	1.318	1.266	1.258	1.275
	(0.012)	(0.013)	(0.008)	(0.006)	(0.011)	(0.012)	(0.011)	(0.014)
I3	91.468	48.554	46.425	46.431	46.412	47.135	47.181	47.096
	(0.467)	(0.123)	(0.108)	(0.156)	(0.104)	(0.125)	(0.211)	(0.195)

Table 1.11: Results of superblock scheduling without inlining on bzip2

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	2.513	0.992	0.948	0.950	0.949	0.946	0.951	0.948
	(0.008)	(0.007)	(0.004)	(0.006)	(0.003)	(0.005)	(0.004)	(0.005)
I2	3.429	1.344	1.294	1.286	1.298	1.252	1.248	1.254
	(0.012)	(0.013)	(0.007)	(0.005)	(0.009)	(0.010)	(0.015)	(0.009)
I3	91.468	48.554	46.215	46.283	46.182	46.739	46.927	46.683
	(0.467)	(0.123)	(0.118)	(0.158)	(0.181)	(0.185)	(0.256)	(0.212)

Table 1.12: Results of superblock scheduling with inlining on bzip2

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	1.167 (0.004)	0.685 (0.003)	0.687 (0.009)	0.688 (0.003)	0.688 (0.006)	0.688 (0.006)	0.681 (0.002)	0.690 (0.004)
I2	2.064 (0.016)	1.180 (0.006)	1.189 (0.007)	1.197 (0.007)	1.200 (0.005)	1.216 (0.003)	1.207 (0.007)	1.218 (0.004)
I3	49.973 (0.128)	28.492 (0.048)	28.502 (0.032)	28.744 (0.057)	28.614 (0.093)	29.753 (0.098)	29.717 (0.084)	29.681 (0.050)

Table 1.13: Results of superblock scheduling without inlining on vortex

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	1.167 (0.004)	0.685 (0.003)	0.663 (0.005)	0.661 (0.003)	0.665 (0.006)	0.663 (0.005)	0.664 (0.005)	0.669 (0.004)
I2	2.064 (0.016)	1.180 (0.006)	1.168 (0.004)	1.139 (0.007)	1.155 (0.006)	1.163 (0.006)	1.161 (0.007)	1.168 (0.009)
I3	49.973 (0.128)	28.492 (0.048)	28.181 (0.044)	27.752 (0.140)	27.796 (0.108)	27.989 (0.083)	27.828 (0.037)	27.912 (0.045)

Table 1.14: Results of superblock scheduling with inlining on vortex

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.079 (0.005)	0.041 (0.003)	0.041 (0.006)	0.045 (0.005)	0.044 (0.003)	0.048 (0.004)	0.049 (0.003)	0.051 (0.003)
I2	4.616 (0.006)	2.354 (0.010)	2.512 (0.012)	2.467 (0.014)	2.581 (0.016)	2.817 (0.015)	2.798 (0.014)	2.815 (0.018)
I3	119.285 (0.109)	62.065 (0.227)	66.245 (0.128)	66.237 (0.131)	66.206 (0.202)	76.792 (0.156)	76.801 (0.168)	76.775 (0.176)

Table 1.15: Results of superblock scheduling without inlining on twolf

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	0.079 (0.005)	0.041 (0.003)	0.040 (0.003)	0.042 (0.004)	0.045 (0.004)	0.049 (0.005)	0.049 (0.004)	0.051 (0.004)
I2	4.616 (0.006)	2.354 (0.010)	2.179 (0.012)	2.167 (0.011)	2.197 (0.021)	2.297 (0.011)	2.268 (0.018)	2.317 (0.012)
I3	119.285 (0.109)	62.065 (0.227)	65.897 (0.214)	65.837 (0.151)	65.765 (0.234)	76.211 (0.284)	76.201 (0.182)	76.192 (0.258)

Table 1.16: Results of superblock scheduling with inlining on twolf

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	6.212	5.888	5.718	5.807	5.742	5.698	5.772	5.792
	(0.254)	(0.156)	(0.121)	(0.136)	(0.098)	(0.124)	(0.131)	(0.142)
I2	6.374	6.012	6.010	5.972	6.016	6.003	5.931	6.042
	(0.312)	(0.278)	(0.218)	(0.168)	(0.308)	(0.128)	(0.245)	(0.264)
I3	531.617	504.122	503.438	509.124	502.687	501.149	507.318	499.622
	(3.213)	(4.154)	(4.074)	(5.067)	(4.876)	(3.712)	(4.897)	(4.012)

Table 1.17: Results of superblock scheduling without inlining on gzip

Input	Normal	O2	Edge I1	Edge I2	Edge I3	Path I1	Path I2	Path I3
I1	6.212	5.888	5.512	5.649	5.618	5.496	5.587	5.612
	(0.254)	(0.156)	(0.181)	(0.167)	(0.171)	(0.116)	(0.128)	(0.086)
I2	6.314	6.012	5.854	5.782	5.908	5.815	5.762	5.812
	(0.312)	(0.278)	(0.221)	(0.308)	(0.231)	(0.189)	(0.298)	(0.312)
I3	531.617	504.122	498.913	500.142	496.522	496.429	499.722	494.567
	(3.213)	(4.154)	(4.159)	(4.091)	(2.072)	(4.232)	(4.912)	(3.765)

Table 1.18: Results of superblock scheduling with inlining on gzip

1.3 Input-Space Clustering

In this experiment we have compared the run time of programs compiled using input-space clustering against compiling techniques shown in previous section. Table 5.19 shows the comparison of runtime of programs compiled with different techniques:

- Rank : Clustering of Path Profile Vectors is done using rankcluster and perfect predictor is used to predict the functions (clusters) at run time.
- Random :Clustering of Path Profile Vectors is done using rankcluster and at run time functions (clusters) are selected randomly.

Inputs (commandlines) used for input-space clustering of different programs are as follows :

- mcf : inp.in > inp.out 2> inp.err (ref input)
- perlbmk :-I./lib splitmail.pl 850 5 19 18 1500 > 850.5.19.18.1500.out 2> 850.5.19.18.1500.err
- parser : 2.1.dict -batch < ref.in > ref.out 2> ref.err
- bzip2 : input.graphic 58 > input.graphic.out 2> input.graphic.err
- vortex : lendian3.raw > vortex3.out 2> vortex3.err
- vpr : net.in arch.in place.in route.out -nodisp -route_only -route_chan_width 15 -pres_fac_mult 2 -acc_fac 1 -first_iter_pres_fac 4 -initial_pres_fac 8 ¿ route_log.out 2> route_log.err
- twolf : ref > ref.stdout 2> ref.err
- crafty : < crafty_ref.in > crafty.out
- gzip : input.source 60 > input.source.out

Program	Normal	O2	Edge	Path	Rank	Random
mcf	38.936	27.257	27.089	27.041	26.852	26.801
	(0.293)	(0.123)	(0.084)	(0.081)	(0.072)	(0.078)
perlbmk	18.248	9.673	9.612	9.558	9.224	10.820
	(0.031)	(0.028)	(0.012)	(0.024)	(0.031)	(0.143)
parser	102.010	63.357	63.421	63.202	62.227	69.346
	(0.121)	(0.063)	(0.068)	(0.083)	(0.104)	(0.185)
bzip2	34.376	17.626	17.475	17.446	17.328	17.456
	(0.043)	(0.081)	(0.043)	(0.036)	(0.052)	(0.072)
vortex	17.701	10.023	9.967	10.010	9.859	9.914
	(0.032)	(0.041)	(0.021)	(0.032)	(0.012)	(0.081)
vpr	33.696	23.192	22.904	22.473	22.393	27.522
	(0.071)	(0.052)	(0.048)	(0.063)	(0.054)	(0.088)
twolf	119.285	62.065	66.206	76.775	75.702	83.502
	(0.109)	(0.227)	(0.202)	(0.176)	(0.110)	(0.185)
crafty	28.358	18.698	19.316	19.526	19.151	40.236
	(0.032)	(0.010)	(0.026)	(0.021)	(0.028)	(0.710)
gzip	70.611	61.350	60.123	59.103	57.996	68.230
	(0.128)	(1.256)	(0.203)	(0.589)	(0.055)	(0.289)

Table 1.19: Results of clustering by rankcluster

On comparing the results of Rank (rankcluster) by Path (superblock scheduling on path profile), we get 0.7% improvement on mcf, 3.49% improvement on perlbmk, 1.54% improvement on parser, 0.68% improvement on bzip2, 1.51% improvement on vortex, 0.36% improvement on vpr, 1.42% improvement on twolf, 1.92% improvement on crafty, 1.87% improvement on gzip. We have been able to produce some improvements, but could not produce very significant results.