运行指南

环境需求

需要Ubuntu系统并安装docker

（下面使用机器为Ubuntu20.04系统，16GB内存，docker版本20.10.21）

docker容器内布局

```
.
|-- bench
|   |-- *.bc           # Bitcode we will analyse.
|   |-- bench.sh       # Benchmarking scripts -- run analyses
|   |-- table.awk      #   and produce tables.
|-- qbe                # Example code we will produce analysable bitcode from.
`-- svf                # SVF source and build tree.
    |-- knl-build      # Build for Knights Landing microarchitecture.
        |-- bin/wpa    # SVF binary (similarly for the following two builds).
    |-- haswell-build  # Build for Haswell microarchitecture.
    |-- base-build     # More portable AMD64/x86-64 build.
    `-- * (remainder)  # SVF source.
```

如何运行

1.到原文地址下载文件 compacting-points-to-sets.tar.gz

2.使用docker运行，在文件目录下打开终端，先后运行如下指令

(1) `docker load < compacting-points-to-sets.tar.gz`

(2) `docker run -it compacting-points-to-sets bash`

如图：

```
ah@Nitro:~$ docker load < compacting-points-to-sets.tar.gz
Loaded image: compacting-points-to-sets:latest
```

```
ah@Nitro:~$ docker run -it compacting-points-to-sets bash
root@62b3b6aa8da7:~#
```

尝试如下命令可以运行一个简单的例子：

(1) `cd $HOME/bench`

(2) `./bench.sh $HOME/svf/base-build/bin/wpa 1 1 1 dhcpcd.bc`

如图开始运行

```
root@62b3b6aa8da7:~# cd $HOME/bench
root@62b3b6aa8da7:~/bench# ./bench.sh $HOME/svf/base-build/bin/wpa 1 1 1 dhcpcd.
bc
 ===== Clustering Benchmark =====
  = args =
    # runs        : 1
    svf binary    : /root/svf/base-build/bin/wpa
    time limit    : 1 hours
    memory limit  : 1 gigabytes
    bitcode files : dhcpcd.bc (1)
    raw data dir  : craw_1673887641
  = start =
    = run #1 of 1 =
      = dhcpcd.bc (file #1 of 1) =
        = running: /root/svf/base-build/bin/wpa -fspta  -fs-time-limit=3600  -op
t-svfg=false  -marked-clocks-only  -node-alloc-strat=dense -staged-pt-type=sbv d
hcpcd.bc
```

指令(2)拆解：

`./bench.sh SVF_BIN NUM_RUNS TIME_LIMIT MEM_LIMIT BITCODE...`

SVF_BIN默认 $HOME/svf/base-build/bin/wpa 即可

NUM_RUNS为执行分析的轮次

TIME_LIMIT为运行时间限制，以小时为单位，超出时间会跑出OOT(超时)的结果

MEM_LIMIT为内存限制，以GB为单位，超出时间会跑出OOM(超内存)的结果

BITCODE...为要运行的文件列表，如我要跑dhcpcd.bc和gawk.bc文件，则可以是 `./bench.sh $HOME/svf/base-build/bin/wpa 1 1 1 dhcpcd.bc gawk.bc`

想要跑出论文中类似的结果，我们使用了如下指令

```
./bench.sh $HOME/svf/base-build/bin/wpa 1 2 15 dhcpcd.bc gawk.bc       keepassxc.bc lynx.bc mutt.bc    ruby.bc table.awk time.txt tmp.txt xpdf.
```

因为我们仅使用了15GB内存，所以很多结果OOM了，如下：

TABLE 4: Required words for SBV

Benchmark	Theoretical	Orginal	Single	Complete	Average	Reduction
dhcpcd.bc	3317195	24726024	*4991412*	6635746	6635082	4.95x
gawk.bc	58007460	429843180	*82989102*	132528508	99502900	5.18x
bash.bc	26586881	289532162	*42914256*	42914700	53173774	6.75x
mutt.bc	51298142	490532984	*102662924*	145767658	160026830	4.78x
lynx.bc	133664618	965029716	267599228	319144056	*215831960*	4.47x
xpdf.bc	--	--	--	--	--	--x
ruby.bc	--	--	--	--	--	--x
keepassxc.bc	13770856	315331336	*54312908*	74407698	74381880	5.81x
					Geo. mean	5.27x

TABLE 5: Required words for CBV

Benchmark	Theoretical	Orginal	Single	Complete	Average	Reduction
dhcpcd.bc	3317195	23911464	*4961417*	6605816	5784023	4.82x
gawk.bc	58007460	429739626	*82783110*	140588641	148836214	5.19x
bash.bc	26586881	295168815	*31731607*	36861568	47120912	9.30x
mutt.bc	51298142	548971337	*87213543*	260457927	259746461	6.29x
lynx.bc	133664618	1015676938	*237113529*	289849510	302122259	4.28x
xpdf.bc	--	--	--	--	--	--x
ruby.bc	--	--	--	--	--	--x
keepassxc.bc	13770856	1399785524	*107456545*	134257494	120881288	13.03x
					Geo. mean	6.59x

TABLE 6: Time

Benchmark	Unclustered			Clustered		SBV/SBV	BV/CBV
	SBV [s]	BV [s]	CBV [s]	SBV [s]	CBV [s]		
dhcpcd.bc	54.85	51.03	63.92	46.13	46.74	1.19x	1.09x
gawk.bc	879.08	732.79	731.83	624.72	601.62	1.41x	1.22x
bash.bc	204.30	165.37	166.57	140.13	140.62	1.46x	1.18x
mutt.bc	496.33	438.60	427.53	370.20	359.54	1.34x	1.22x
lynx.bc	OOM	OOM	OOM	1532.27	1560.07	--x	--x
xpdf.bc	OOM	OOM	OOM	OOM	OOM	--x	--x
ruby.bc	OOM	OOM	OOM	OOM	OOM	--x	--x
keepassxc.bc	688.88	OOM	OOM	515.54	511.04	1.34x	--x
					Geo. mean	1.34x	1.18x

TABLE 7: Memory

Benchmark	Unclustered			Clustered		SBV/SBV	BV/CBV
	SBV [GB]	BV [GB]	CBV [GB]	SBV [GB]	CBV [GB]		
dhcpcd.bc	1.20	0.92	0.91	0.74	0.68	1.62x	1.34x
gawk.bc	12.76	8.00	7.79	4.63	3.67	2.75x	2.18x
bash.bc	9.00	4.93	5.06	3.23	2.66	2.79x	1.85x
mutt.bc	14.28	11.67	11.45	5.47	4.56	2.61x	2.56x
lynx.bc	OOM	OOM	OOM	11.52	9.37	>=1.30x	>=1.60x
xpdf.bc	OOM	OOM	OOM	OOM	OOM	--x	--x
ruby.bc	OOM	OOM	OOM	OOM	OOM	--x	--x
keepassxc.bc	12.41	OOM	OOM	6.30	6.21	1.97x	>=2.41x
					Geo. mean	>=2.09x	>=1.94x

原论文使用了100GB的机器，所以能跑出来的结果更多。