

Fotran->LLVM IR->C

Fortran のプログラム (なすばら) を LLVM の中間表現にして ,
そこから C 言語に変換して , MIPS のクロスコンパイラでバイナリ生成という
長い道のりでプログラムを実行してみた . (IS は , C->LLVM IR->C)
だいたい , のような感じでコンパイル .
BT , LU , SP は各ソースコードからまず OBJ 作ってリンク

```
# compile
OBJ=`echo $1 | sed 's/%.f$/./g'`
CBE=`echo $1 | sed 's/%.f$/./cbe.c/g'`
LL=`echo $1 | sed 's/%.f$/./ll/g'`
BC=`echo $1 | sed 's/%.f$/./bc/g'`
/usr/local/llvm-gcc-4.2-2.6-x86_64-linux/bin/llvm-gfortran ¥
-S -emit-llvm -l../lib/MPI -O2 -Wall $1 -o $LL
/usr/local/llvm-2.6/bin/llvm-as -o - $LL | /usr/local/llvm-2.6/bin/opt -std-compile-opts -o $BC
/usr/local/llvm-2.6/bin/llc -march=c $BC
/usr/local/mipsel-gcc4.2/usr/bin/mipsel-linux-gcc -O2 -Wall -l../lib/MPI -O2 -c $CBE -o $OBJ
# link
/usr/local/mipsel-gcc4.2/usr/bin/mipsel-linux-gcc ¥
-O2 -Wall -static ¥
-o ../bin/cg.S.4 ¥
../common/print_results.o ../common/timers.o ../common/randdp.o ¥
../lib/MClib.o ../lib/MPI/mpi.o ../lib/MPI/mpif.o ¥
$OBJ ¥
-lgfortran -lgfortranbegin -lm
```

まず , うまくいくというのがすごいなあ , と .
で , 実行時間の比較をしてみると , 仕方ないのだけど , 割と遅くなってしまっている .
llvm-gfortran のところで最適化しない方がいいとか , ちょこちょこ試してみたけど ,
実行時間におけるたいした違いは確認できなかった .
ここで , BT , LU , SP はいくつかの Fortran ソースから構成されているベンチマーク ...
うーん , コアのソースの部分だけ LLVM 介すとかやってみるかなあ .
ちなみにコード数
... コード数っていう , ざっくりな割に , なんとなくの相関がみられるのが面白い

日々メモ

- VMWareFusion 上の debian とのファイル共有が面倒になったので , すっぱりと home を /mnt/hgfs/miyo に変更 . 楽になった . (Mon Jan 25 17:51:18 2010 +0900)
- Parsec の blackscholes を C->LLVM->C でコンパイルしたバイナリの実行時間は 2% 増 . これは嬉しい . (Mon Jan 25 17:40:59 2010 +0900)
- やっぱり LLVM で Fortran->C 変換したプログラムは断然遅い . コア部分だけ変換するようにはしないとだめだな ... (Mon Jan 25 14:38:01 2010 +0900)