# Post Link Outlining Pass

基于 LLVM BOLT 构建的Post Link Outlining Pass，旨在通过将重复的指令序列提取为函数来减小二进制代码体积。

代码仓库：https://github.com/ddsfda99/llvm-project

## 实测优化效果

### 1. basicmath

使用bolt优化得到basicmath_large.plos

```
cd ~/Lab5-test/basicmath
llvm-bolt basicmath_large -o basicmath_large.plos \
  --enable-outliner \
  --eliminate-unreachable \
  --simplify-conditional-tail-calls \
  --peepholes=all \
  --icf=all \
  --use-old-text \
  --align-text=4 \
  --align-functions=4
```

不加PGO效果

```
root@iZt4ndit8e8tr3irczjsrjZ:~# size -A ~/Lab5-test/basicmath/basicmath_large.plos | grep text
.bolt.org.text               0      2112
.text                    15868      2112
.text.injected             212     17980
```

### 2. blowfish

使用bolt优化得到bf.plos

```
cd ~/Lab5-test/blowfish
llvm-bolt bf -o bf.plos \
  --enable-outliner \
```

```
  --eliminate-unreachable \
  --simplify-conditional-tail-calls \
  --peepholes=all \
  --icf=all \
  --use-old-text \
  --align-text=4 \
  --align-functions=4
```

不加PGO效果

```
root@iZt4ndit8e8tr3irczjsrjZ:~# size -A ~/Lab5-test/blowfish/bf | grep text
.text                  3832     2240
root@iZt4ndit8e8tr3irczjsrjZ:~# size -A ~/Lab5-test/blowfish/bf.plos | grep text
.bolt.org.text            0      2240
.text                  3724      2240
.text.injected          104      5964
```

## 3. lame

使用bolt优化得到lame.plos

```
cd ~/Lab5-test/lame
llvm-bolt lame -o lame.plos \
  --enable-outliner \
  --eliminate-unreachable \
  --simplify-conditional-tail-calls \
  --peepholes=all \
  --icf=all \
  --use-old-text \
  --align-text=4 \
  --align-functions=4
```

不加PGO效果

```
root@iZt4ndit8e8tr3irczjsrjZ:~# size -A ~/Lab5-test/lame/lame | grep text
.text                 87100    12288
root@iZt4ndit8e8tr3irczjsrjZ:~# size -A ~/Lab5-test/lame/lame.plos | grep text
.bolt.org.text            0     12288
.text                 79800     12288
.text.injected         4292     92088
```

### 4. typeset/lout

使用bolt优化得到lout.plos

```
cd ~/Lab5-test/typeset/lout-3.24
llvm-bolt lout -o ../lout.plos \
  --enable-outliner \
  --eliminate-unreachable \
  --simplify-conditional-tail-calls \
  --peepholes=all \
  --icf=all \
  --use-old-text \
  --align-text=4 \
  --align-functions=4
```

不加PGO效果

```
root@iZt4ndit8e8tr3irczjsrjZ:~# size -A ~/Lab5-test/typeset/lout-3.24/lout | grep text
.text                 399640    25024
root@iZt4ndit8e8tr3irczjsrjZ:~# size -A ~/Lab5-test/typeset/lout.plos | grep text
.bolt.org.text             0     25024
.text                 348152     25024
.text.injected         20156    373176
```

## 使用方法

### 基本用法

```
llvm-bolt <input_binary> -o <output_binary> --enable-outliner
```

### Profile 指导用法

```
# Generate instrumented binary
llvm-bolt binary -o binary.instrumented --instrument

# Run with representative workload
./binary.instrumented <workload>
```

```
# Optimize using collected profile data
llvm-bolt binary -o binary.outlined \
  --data=/tmp/prof.fdata \
  --enable-outliner
```

## 命令行选项

| Option | Description | Default |
| --- | --- | --- |
| --enable-outliner | 启用post link outlining pass | false |
| --outliner-cold-only | 只outline冷代码（有profile生效） | true |
| --outliner-cold-threshold | 冷基本块的执行次数阈值 | 0 |
| --outliner-max-iterations | 最大outline迭代次数 (用于 nested outlining) | 8 |

## 相关源码文件

| Path | Description |
| --- | --- |
| `bolt/include/bolt/Passes/Outliner.h` | Outliner Pass 类定义和接口 |
| `bolt/lib/Passes/Outliner.cpp` | Outliner Pass 实现 |
| `bolt/lib/Passes/CMakeLists.txt` | 构建配置，包含 AArch64 头文件依赖 |
| `bolt/lib/Rewrite/BinaryPassManager.cpp` | 注册 `--enable-outliner` 选项及 Pass |