



NKUF4

编译系统设计赛

队员：徐文斌 许友锐 栗心武 聂志强

指导教师：王刚 李忠伟

南开大学



目录 CONTENTS

01

整体架构

02

中间代码优化

03

目标代码优化

04

效果与总结

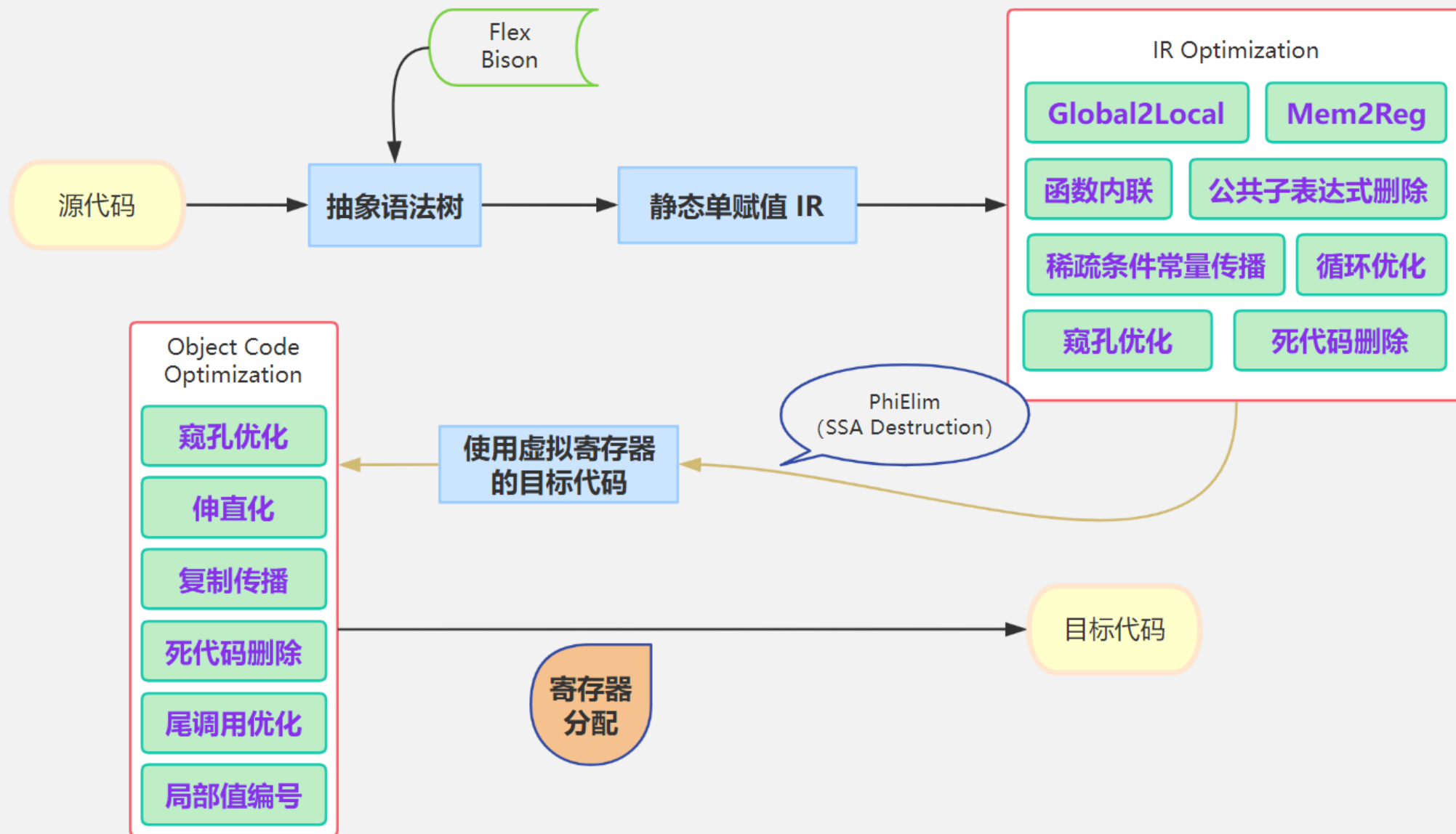
整体架构





设计架构

DESIGN ARCHITECTURE



中间代码优化



- 全局变量局部化
- Mem2Reg
- 函数内联
- 稀疏条件常量传播
- 公共子表达式删除
- 控制流简化
- 死代码删除
- 全局值编号
- 循环不变代码外提
- 循环展开
- 循环强度削弱
- 窥孔优化

■ 全局变量局部化

- ✓ 对于全局int/float类型变量，转换为函数内的局部变量。
- ✓ 在函数内发生函数调用前，store 被调函数需要 load 的全局变量，调用后 load 被调函数 store 的全局变量
- ✓ 返回前 store 所有发生修改的全局变量
- ✓ 对于全局从未发生 store 的全局变量，将其视为常数处理。

■ memory to register

- ✓ 将 load/store 形式的 SSA 转换成 phi 形式的 SSA
- ✓ 减少多余 load/store，并便于后续优化

■ 稀疏条件常数传播

- ✓ 利用流图的边和SSA边来传递信息实现程序的符号执行
- ✓ 更精确地传播常数及移除无用代码

■ 公共子表达式删除

- ✓ 基于可用表达式分析进行
- ✓ 结合纯函数分析删除重复函数调用
- ✓ 在 store 指令后添加 load 指令，从而删除更多 load 指令

■ 函数内联

- ✓ 对于非递归的函数，如果函数参数过多或函数代码较短，则选择将函数内联。
- ✓ 减少由于函数调用带来的额外开销，为后续优化创造条件。

■ 死代码删除

- ✓ 标识所有定义必要值的指令，包括部分函数调用及返回语句。
- ✓ 迭代标识所有对定义必要值有贡献的指令 (`markBasic` & `markStore`) 。
- ✓ 到达不动点时，未被标记的指令均为死代码，从而进行删除。

■ 全局值编号

- ✓ 按逆拓扑序遍历的全局值编号，将所有计算尽可能前提。
- ✓ 把phi节点中的值伴随前提，将冗余表达式删除。
- ✓ 同时完成了循环代码外提和phi节点化简。

循环不变代码外提

减少循环中重复代码的计算。

循环展开

若循环次数确定，则全部展开；
否则以4为单位展开，减少跳转语句，并为冗余删除创造条件。

循环优化

循环强度削弱

计算归纳变量和循环不变操作数信息，接下来将循环体中与部分归纳变量相关的计算转化为加法指令。

目标代码优化



- 死代码删除
- 控制流优化
- 窥孔优化
- 局部值编号
- 复制传播
- 强度削弱

■ 死代码删除

- ✓ 基于活跃变量分析完成死代码删除

■ 局部值编号

- ✓ 基本块内值编号，主要删除数组相关的冗余

■ 控制流优化

- ✓ 单前驱与单后继块的合并
- ✓ 只有一句无条件跳转的基本块的删除
- ✓ 删除顺序连接基本块间的无条件跳转
- ✓ 如果条件跳转目标块内指令较少，将该块并入当前块

目标代码优化

OBJECT CODE OPTIMIZATION

■ 强度削弱

- ✓ 优化目标代码中的乘除模指令

■ 复制传播

- ✓ 清除由其他优化引入的mov指令
- ✓ 清除ir代码至asm代码翻译中引入的mov指令

■ 窥孔优化

- ✓ 使用更快速的指令替换特定指令序列

强度削弱

Origin Calculation	Optimized Calculation
$y = x / 8$	$y = x \gg 3 \ (x > 0)$
$y = x * 64$	$y = x \ll 6$
$y = x \% 4$	$y = x \& 3$

窥孔优化

Origin Calculation	Optimized Calculation
<code>add v11, fp, #-12</code> <code>ldr v277, [v11]</code>	<code>ldr v227, [fp, #-12]</code>
<code>mul v0, v1, v2</code> <code>add v3, v4, v0</code>	<code>mla v3, v1, v2, v4</code>

图着色寄存器分配

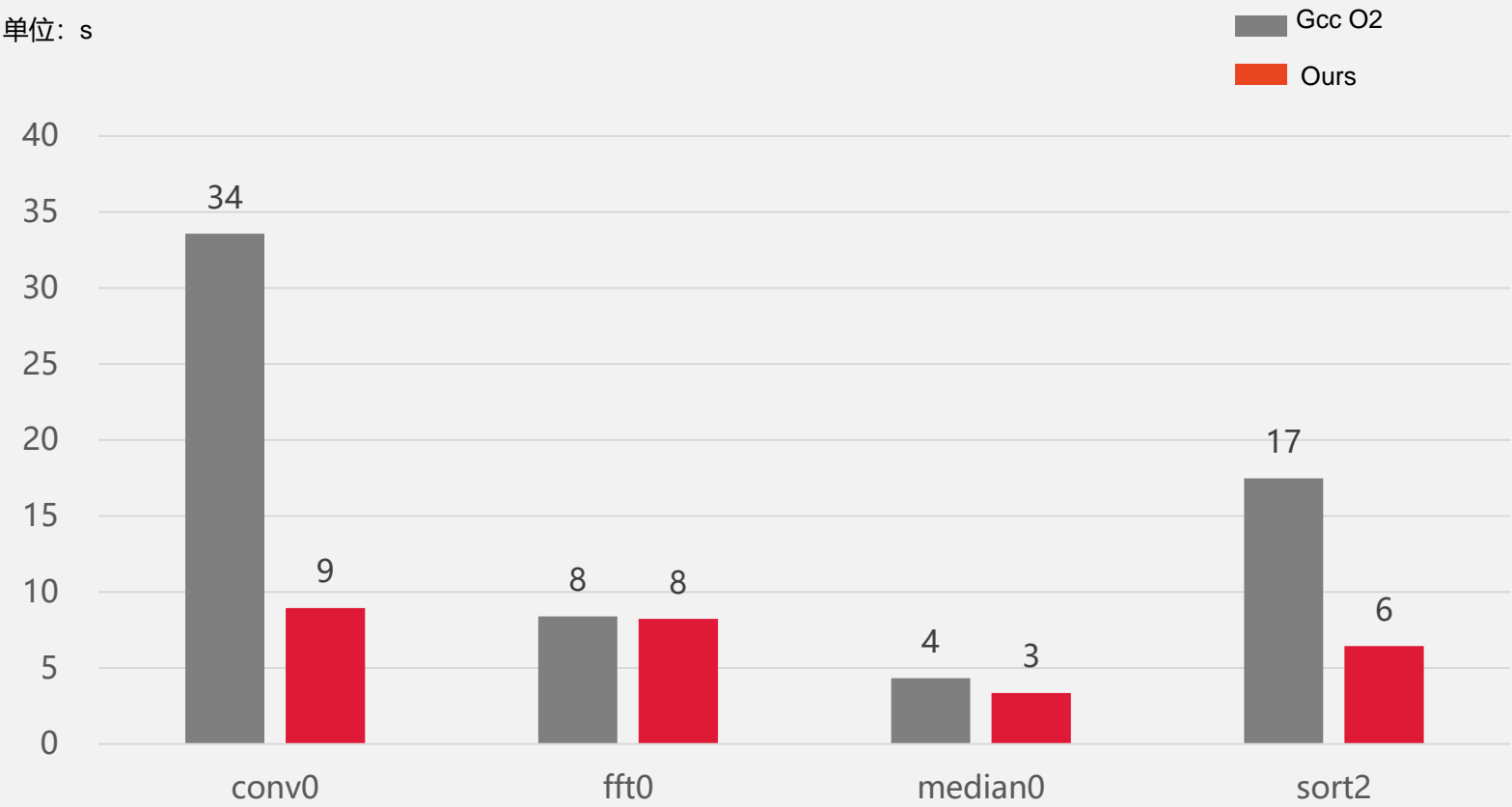
- 基于图着色算法，将虚拟寄存器映射为arm通用寄存器
- 使用到达定值和活跃变量分析构建干涉图
- 使用george算法进行干涉图聚合
- 根据循环嵌套层数以及节点的度数估算溢出代价
- 尽可能多地使用寄存器，使用ip寄存器和lr寄存器，共有13个寄存器可供分配，减少寄存器溢出

效果与总结



优化效果

优化前后运行时间对比



决赛成绩

比赛提交到排行榜更新有20秒左右的延迟

详情

#	用户名	队伍	提交次数(ASC)	最后提交时间(ASC)	正确分	性能分	总分
1	202310006201934	那一年喵喵变成了光/ 北京航空航天大学	16	2023-08-21 16:08:17	100	83.7092	87.3258
2	202310006201725	ATRI/ 北京航空航天大学	18	2023-08-21 17:15:19	100	61.0970	69.7335
3	202310055201422	NKUF4/ 南开大学	16	2023-08-21 15:52:09	100	51.5978	62.3431
4	202310055201751	北关大学第83号代表队/ 南开大学	68	2023-08-21 17:40:29	100	51.1093	61.9631
5	202310614201437	ARM32栈错误/ 电子科技大学	38	2023-08-21 18:20:21	100	51.0590	61.9239
6	202310246201860	去偷毕昇杯/ 复旦大学	11	2023-08-21 13:01:46	100	50.8414	61.7546
7	202310532201184	卷窝鸣人/ 湖南大学	8	2023-08-21 12:01:08	100	50.2735	61.3128
8	202310006201717	快码加编队/ 北京航空航天大学	9	2023-08-21 14:26:21	100	49.6768	60.8486
9	202310055201427	没有op就不配拿奖吗/ 南开大学	2	2023-08-21 17:59:53	99	45.9520	57.7286
10	202310006201896	编译三缺一/ 北京航空航天大学	7	2023-08-21 17:05:28	100	44.9699	57.1866

謝謝
觀看