LLVM Essentials-Packt 2016（读书笔记）：TableGen讲解并不透彻，另外我还想知道后端优化步算法到底怎么编写？

原创 2016年12月18日 15:35:20 标签：llvm / 编译器 / SSA / TableGen / Lowering 1373

LLVM Essentials

目录 [隐藏]

Playing with LLVM[编辑]

1. 寄存器变量（%var）、栈变量（alloca，%1 ...）、
2. .c-->.bc：$ clang -emit-llvm -c main.c
3. .bc-->.s：$ llc output.bc –o output.s
4. .ll-->.bc：$ llvm-as add.ll –o add.bc

志_祥

原创 407 粉丝 101 喜欢 44 评论 98

等级：博客 6 访问量：45万+

积分：8594 排名：2812

他的最新文章 更多文章

std::vector的reserve、resize与堆内存破坏

Building LLVM IR[编辑]

```
static LLVMContext &Context = getGlobalContext();
static Module *ModuleOb = new Module("my compiler", Context);
```

```
FunctionType *funcType = llvm::FunctionType::get(Builder.getInt32Ty(), false); //注意这里type被简写为Ty了
Function *foo = llvm::Function::Create(funcType, llvm::Function::ExternalLinkage, Name, ModuleOb);
```

这里的'外部链接'实际上是指导出符号；

```
BasicBlock* BB = BasicBlock::Create(Context, Name, fooFunc);
```

全局变量：

```
ModuleOb->getOrInsertGlobal(Name, Builder.getInt32Ty());
GlobalVariable *gVar = ModuleOb->getNamedGlobal(Name); ...
//得到：
@x = common global i32, align 4
```

插入返回值语句：

```
Builder.SetInsertPoint(entry); //注意，SetInsertPoint API显然是有状态的；
Builder.CreateRet(Builder.getInt32(0));
```

设置函数参数：略

分支语句：需要phi merge节点

```
PHINode *Phi = Builder.CreatePHI(Type::getInt32Ty(getGlobalContext()), PhiBBSize, "iftmp");
Phi->addIncoming(ThenVal, ThenBB);
```

CSS布局里图文混排的缺陷（语义描述不完备）

Why Isomorphic Go（同态Go应用，用Go语言编写前端和后端）

一些记录

文章分类

WebKit 42篇
编译器技术 32篇
程序员心得体会 105篇
系统架构 49篇
读书笔记 283篇
Chromium 9篇
展开

文章存档

2018年3月 2篇
2018年2月 1篇
2018年1月 2篇
2017年11月 1篇
2017年10月 7篇
2017年9月 10篇

循环：略

```
...
Builder.CreateCondBr(EndCond, LoopBB, AfterBB);
...
```

高级IR[编辑]

1. getelementptr：offset支持负值吗？
2. lo
3. st
4. in ement（其实不就是给数组元素赋值吗？）
5. extractelement

   %0 = extractelement <4 x i32> %a, i32 0 //注意这里数组类型的写法，类型写在变量的前面

基本IR变换[编辑]

1. runOn{Passtype}: Module、Function、BasicBlock、Loop
2. getAnalysisUsage：指定pass之间的依赖关系
    1. AU.addRequired<AliasAnalysis>(); //注意这里使用了成员函数模板
    2. addRequiredTransitive
    3. addPreserved
3. 指令简化
    1. if (match(Op0, m_Not(m_Specific(Op1))) || match(Op1, m_Not(m_Specific(Op0)))) //注意这里的匹配模板写法
    2. instcombine：化简成等价且更少的指令

高级IR块变换[编辑]

他的热门文章

AdBlock广告拦截插件的实现原理
9936

腾讯的垃圾电话面试，补充更新
8992

2017.10.14晚，用迅雷下载大部分BT资源出现失败，tracker服务器被封了？FK
7789

深入理解OpenCV：实用计算机视觉项目解析 笔记
6074

Designing Data-Intensive Applications（设计数据密集应用）- O'Reilly 2017 读...
3980

关于“微信订阅号/制作相册”
3872

Mastering Bitcoin（掌握比特币）读书笔记
3864

算法设计手册（第2版）读书笔记, Springer - The Algorithm Design Manual, 2ed Ste...
3746

基于DevTools协议+Chromium headless的客户端爬虫框架
3230

计算机算法设计与分析（第4版） 王晓东 著 2012.2 笔记（这本书还不错，偏实 ...
3170

2. 循环规范化：增加preheader、exit block，只允许一个backedge等等
3. LoopPass基类、LPPassManager（llvm的类方法命名总是喜欢突然来个缩写，fuck）
4. LICM（循环不变式外提）
5. 更多的循环优化：lib/Transforms/Scalar

2. Scalar evolution（更高级的“抽象解释”？）
    1. $ opt -analyze -scalar-evolution scalevl.ll
3. LLvm intrinsics（编译器内置函数）
    1. call void @llvm.memset.p0i8.i64(i8* %a2, i8 0, i64 20, i32 16, i1 false) //这让人感觉所谓的LLVM编译器其实只是解释器？（runtime函数）
    2. %1 = getelementptr inbounds [5 x i32], [5 x i32]* %a, i64 0, i64 0
4. Vectorization（不是特别的清楚，“Loop-Aware SLP in GCC”by Ira Rosen, etc？）
    1. 2种类型：SLP、Loop vectorization
    2. SIMD
    3. $ opt -S -basicaa -slp-vectorizer -mtriple=aarch64-unknown-linuxgnu -mcpu=cortex-a57 addsub.ll –debug

IR到Selection DAG阶段[编辑]

1. SelectionDAGBuilder：以%add = add nsw i32 %a, %b为例
    1. SelectionDAGBuilder::visit
    2. visitAdd

        visitBinary **SDValue**?
2. Legalizing SelectionDAG（合法化，目标平台适配）
    1. 例：X86上sdiv扩展到sdivrem
3. Optimizing SelectionDAG
    1. DAGCombiner

1. X86DAGToDAGISel::SelectCode() TableGen自动生成（llvm很难理解的地方就是TableGen的语法）

5. Scheduling and emitting machine instructions
    1. InstrEmitter::EmitMachineNode：SDNode ==> MachineInstr（MachineBasicBlock）
    2. MachineInstrBuilder
        1. CreateVirtualRegisters（这里还是‘虚拟寄存器’？）
        2. virtual AdjustInstrPostInstrSelection

6. Register allocation

    1. spilling

    2. SSA form deconstruction（phi到reg copy）
    3. 映射虚拟寄存器到物理寄存器：2种方法

        1. 直接映射：TargetRegisterInfo/MachineOperand（程序员自己实现？）
        2. 间接：VirtRegMap::assignVirt2Phys（llvm内置的？）
    4. llvm 4种分配技术：
        1. Basic
        2. Fast
        3. PBQP
        4. Greedy

7. Code Emission：LLVM JIT和MC（生成obj格式的文件）
    1. AsmPrinter：使用平台特定的MCInstLowering接口如X86MCInstLower
    2. MCInst指令传递给MCStreamer对象
    3. 注意，the MC Layer is one of the big difference between LLVM and GCC.（GCC生成汇编格式的代码，依赖于平台外部汇编？）

8. $ llc test.ll -show-mc-encoding -o -

见鬼，我还是没有明白SDAG的作用（LLVM IR里不是有循环吗？为什么SDAG就变成DAG了呢？）

1. 没有tablegen，llvm本身只具有学术意义，有了tablegen，llvm才变成了可工业使用的牛逼库
2. pipeline：SelectionDAG --> MachineDAG --> MachineInstr --> MCInst
3. 定义一个玩具后端：r0-3, sp, pc, cpsr（pc？）

- Defining registers and register sets

  每个寄存器都有一个唯一编号，这要求平台指令中的寄存器位表示是一致的（当然，有些是隐含的比如push/pop）

- Defining calling convention（ABI）

  def CC_TOY : CallingConv<[

  CCIfType<[i8, i16], CCPromoteToType<i32>>, //8位、16位的提升到32位

  CCIfType<[i32], CCAssignToReg<[R0, R1]>>,

  CCIfType<[i32], CCAssignToStack<4, 4>> //开始2个参数R0，R1寄存器传递，剩余的通过栈传递

  C_Save : CalleeSavedRegs<(add R2, R3)>;

- Defining the instruction set

  def ADDrr : InstTOY<(outs GRRegs:$dst), (ins GRRegs:$src1, GRRegs:$src2), "add $dst, $src1,z$src2", [(set i32:$dst, (add i32:$src1, i32:$src2))]>;

- Implementing frame lowering
  - Frame lowering involves emitting function prologue and epilogue.（llvm ir是直接定义函数的，包括ret指令）
  - void TOYFrameLowering::emitPrologue(MachineFunction &MF) const {

    const TargetInstrInfo &TII = *MF.getSubtarget().getInstrInfo();

    MachineBasicBlock &MBB = MF.front();

    MachineBasicBlock::iterator MBBI = MBB.begin();

    uint64_t StackSize = computeStackSize(MF);

    unsigned StackReg = TOY::SP;

    unsigned OffsetReg = materializeOffset(MF, MBB, MBBI, (unsigned)StackSize);

    ... //略

- Lowering instructions

  代码略

目前您尚未登录，请 登录 或 注册 后进行评论

0

LLVM.Essentials.1785280

2016年02月22日 1.42MB 下载

PDF

LLVM Essentials

2016年09月02日 20:43 4.16MB 下载

PDF

如何洞悉PHP的精要？

多年技术专家应用剖析。

LLVM图书合集

2016年02月18日 11:22 15.45MB 下载

ZIP

LLVM（二）：后端结构

windiwen 2014年02月20日 16:12 896

LLVM后端的功能就是处理LLVM IR（中间表达式）并生成目标机器指令。 要实现一个LLVM编译器的后端需要以下步骤： 1. 描述目标机器的特性 2. 描述目标机器的寄存器 3. ...

用llvm简单遍历一个bc文件

u010260808 2016年09月21日 16:25 647

我是个编程菜鸟，学了很久编程，都不知道通过main函数参数传递的强大，之前想写个llvm小程序去遍历bc文件里的指令，通过文件流的方式去读写，最终错漏百出。 今天带来一个简单的入门程序，...

一秒创造无[illegible]算的价值

你或许应该[illegible]的LLVM

khlljm 2016年07月04日 18:57 5677

原文链接 作为iOS或者Mac开发者，你也许非常眼熟LLVM这个字眼，但也许没有太去在意它。在很长的一段时间内，我就是处于这个状态，不知道它背后是在干嘛。随着苹果新语言swift的发布，我看...

LLVM全时优化

dashuniuniu 2015年12月23日 11:14 2238

引子由于实验室项目原因，从本科开始接触Clang和LLVM开始到现在已经有2年时间了，期间都是针对Clang做一些边边角角的工作，没有潜下心来好好研读一下LLVM的paper。最近闲下来读了Chris...

LLVM中TableGen工具的使用

u010902721 2014年10月18日 20:42 2471

在描述处理器结构时，需要

LLVM代码生成器进一步深入，第一部分

wuhui_gdnt 2014年11月12日 15:59 2681

原作：http://[illegible]

LLVM（三）：Tablegen简介


windiwen 2014年03月04日 10:03 1304

上一篇介绍实现llvm后端需要做的一些工作，有很大一部分工作是描述目标
体系结构的特征，包括指令集，寄存器等信息。Tablegen就是用于记录这些信息的描述性语言。目标体系结构目录下的*.td文件都是用...

LLVM Table 介绍


dreammeard 2014年02月19日 17:04 1947

1.TableGen间介 TableGen是用来帮助程序员开发和维护某一领域想逛的信息记录。它定义了一套描述这些信息记录的数据结构和语法规范。编写代码时，如果使用TableGen描述信息记录，我们可...

LLVM Essentials-Packt 2016（读书笔记）：TableGen讲解并不透彻，另外我还想知道...

LLVM Essentials 目录 [隐藏] 1 Playing with LLVM2 Building LLVM IR3 高级IR4 基本IR...


cteng 2016年02月18日 15:35 1373

在llvm的clang中添加新的后端和Intrinsic function

本文记录一下如何在llvm的clang源码中添加一种新的后端（暂时命名为cpu0），并在其中添加Intrinsic function。涉及到的文件列

u010902721 2015年05月14日 21:31 2709

编写LLVM的后端(一)

michael_kang 2010年12月26日 11:29 3295

关于作者：SkyEye项目的维护者，长期从事系统软件的开发和培训工作。 本文档是指如何编写一个可以把LLVM的中间表示转换成一个特定的机器或者其他语言的后端。对于一个特定机器的代码或者是汇编语言或者是...

.NET 开源开发项目【翻译】

wangkaiming123456 2016年11月24日 08:55 819

原文地址 本文列出 .NET 开源开发项目（open source developer projects）。意在包括对开发过程的所有方面有所帮组的项目。对于消费项目（consumer proj...

[it-ebooks]书列表v0.1.1

mapoor 2015年02月06日 00:17 4838

感谢it-ebooks团队 #### it-ebooks电子书质量很好，但搜索功能不太完善 #### 格式说明： [年份] 书名 || 副标题 || 页码 || 链接 #...

LLVM学习笔记（9）

wuhui_gdnt 2017年04月20日 11:39 488

3. TableGen生成的代码 3.1. 概述 在编译LLVM时，首先会调用TableGen解析TD文件，产生C++源代码，然后这些C++源代码与LLVM的其他源代...

LLVM学习笔记（16）

wuhui_gdnt 2017年08月18日 12:05 220

LLVM TableGen

qq_23484921 2018年03月18日 23:31 2

不是很了解，看起来是一种td是一个文件描述类的文件，然后通过工具可以生成什么东西。...

让我们致力于一个LLVM超级优化器

wuhui_gdnt 2016年10月28日 11:38 957

作者：JohnRege ofessor of Computer Science, University of Utah, USA 原文地址：http://blog.regehr.org/a...

0

有眼睛的扫 器人，看得见才能不漏扫

首发预定，减20

LLVM后端开发

jinweifu 2017年01月06日 08:20 1886

LLVM后端 介绍这个文档描述了编写编译器后端的技术，将llvm IR转化为定制的机器代码或者其他语言。意图生成的特定机器码可以是汇编形式或者二进制形式（能够被JIT编译器使用）。LLVM的后端有一个...

【转】为LLVM移植一个新的后端所需的几个基本步骤

csdidi 2009年12月06日 22:35 1650

To write a compiler backend for LLVM that converts the LLVM IR tocode for a specified target (machin...

致新手 android中ScrollView嵌套listview

liuwei6551234 2016年10月15日 15:47 151

先看效果 画质有点垃圾 格式工厂转换的 布局文件

java到底如何才能学的透彻？

javaniuniu 2016年10月01日 14:53 495

java到底如何才能学的透彻？

我到底是选择前端还是后端

wjy632975989 2011年03月23日 13:22 508

过几天新团队要开始做网站了，学长让我选择一条路走。我不确定自己
适合哪方面，前端设计的话，我现在至少看得明白CSS和少部分的JS 知道JQ。然后后端方面 PHP看了一点，最近在准备ACM的市赛，主要是C...

PCA的数学[illegible](讲解很透彻)

aiaiai010101 2017年05月25日 20:18 299

PCA（Principal Component Analysis）是一种常用的数据分析方法。PCA通过线性变换将原始数据变换为一组各维度线性无关的表示，可用于提取数据的主要特征分量，常用于高维数据的降...

K-均值和K-SVD算法—最简单最透彻的讲解,不要看其他资料了

一。K-均值算法：（每个信号由一个Ck表达） 输入参数： 目标函数： ek为自然向量，除第k个分量为1，其余为0 输出：（1）求坐标矩阵X（本质上是稀疏编码，每一列非零元只有一...

qq1028850792 2013年10月21日 14:51 38530

LLVM Pass 初探

xuzhezhaozhao 2014年09月12日 22:46 2241

1. 首先要配置环境

LLVM每日谈之四 Pass初探

snsn1984 2012年11月08日 09:29 10588

作者：snsn1984 LLVM 的Pass框架是LLVM系统的一个很重要的部分。每个Pass都是做优化或者转变的工作，LLVM的优化和转换

llvm學習（三）————如何編譯自己的第一個Pass

一句話編譯自己的Pass： test$ `llvm-config --bindir`/clang -shared -fPIC `llvm-config --cxxflags` `llvm-con...

ningxialieri 2014年01月14日 21:21 2202

人民日报读 记-2015年1月份 高端制造

ffm83 2015年02月01日 19:38 14298

2015年1月份已经 将人民日报上的能看懂的内容进行整理和简单的分类；在我的屌丝梦中，有一个媒体，它所报道出来的内容和信息，应该能 表这个国家的发展的声音；如果我需要了解这些内容，那么人民日报可...

深入理解JVM读书笔记四：（早期）编译器优化

xunzaosiyecao 2016年10月19日 21:28 1527

10.1概述Java 语言的 “编译期” 其实是一段 “不确定” 的操作过程，因为它可能是指一个前端编译器（其实叫 “编译器的前端” 更准确一些）把 .java 文件转变成 .class 文件的过程；...

前瞻-全时优化和LLVM-1（转）

aust_syj 2012年04月18日 22:12 151

前瞻-全时优化和LLVM-1（转） 1，写在前面的话 全时优化(LifeLong Optimization)对于每个编译爱好者来说，太有魅力了。我在起初也是被这个题目所吸引打算一探究竟。...

编译器架构的王者LLVM——（12）使用JIT引擎

sun_xiaofan 2016年01月08日 20:17 4194

LLVM从设计之初就考虑了解释执行的功能,这非常其作为一款跨平台的中间字节码来使用,可以方便地跨平台运行。又具有编译型语言的优势，非常的方便。 我们使用的LLVM3.6版，移除了原版JIT，改换成了新...

漫谈机器学习经典算法—理解EM算法

 lanbing510 2015年11月15日 15:46 5478

公式显示有问题，可移步http://lanbing510.info/2015/11/12/Master-EM-Algorithm.html写在前面EM（Expectation Maximization...

llvm各种格式文件转换图

 u012033027 2014年11月14日 23:14 1715

llvm主要具有以下 格式的文件。


0

”标准答案没出来，我怎么知道我想表达什么？……“这样的话，根本就不仅仅是笑话，而...

转帖：http://www.jianshu.com/p/f6342f581f47 每年高考都有这样的事情，再美的灵感也抵不过“套路！！套路！！套路！！”看完这篇，被大学上过的，现在有孩...

 coolhe21cn 2017年06月09日 16:07 467

LLVM学习笔记（10）

 wuhui_gdnt 2017年04月28日 11:37 254

3.3. 寄存器的后端描述 选项“-gen-register-info”会促使Tablegen根据文件TargetRegisterInfo.td中的寄存器定义生成源文件Target...

使用LLVM分析函数CFG

 wuhui_gdnt 2017年08月11日 11:35 387

作者：Eli Bendersky http://eli.thegreenplace.net/2013/09/16/analyzing-function-cfgs-with-llvm 在Stack ...