

# 计算机体系结构论文复现

20305055 熊明

## 概述

我复现的是一篇关于体系结构与深度学习的文章，该文章旨在探索基于机器学习的计算机体系结构仿真方法的可能性。本文提出了一种全新的架构SimNet，该指令是一种新颖的以指令为中心的模拟框架，它将程序模拟分解为单个指令延迟，并使用定制的ML技术进行指令延迟预测。

## 背景

周期精确的离散事件模拟器(DES)广泛应用于计算机体系结构研究和工程中，可以实现新的建筑思想，以及设计空间的探索。DES由不同的模块组成，这些模块模拟不同硬件组件的行为。在某些事件中(例如，推进一个周期)，这些单独的组件及其相互作用被模拟以模仿处理器的行为。不幸的是，DES对计算的要求非常高，这大大降低了它在整个系统和应用程序规模上的实用性和适用性。

通过软件工程优化、多核/多核仿真并行化和统计方法，人们已经做出了许多努力来提高传统的仿真速度。其中，由于gpu等大规模并行加速器的广泛应用，并行化是一个很有前途的方向。然而，这些传统模拟器的一个基本限制是，由于不同组件的异构性、组件之间的频繁交互和不规则行为，它们本质上难以并行化。因此，现有的并行模拟器侧重于粗并行化策略，其中单个内核的模拟是并行化的。这些模拟器的可扩展性有限，不能利用现代并行加速器。

## 研究问题

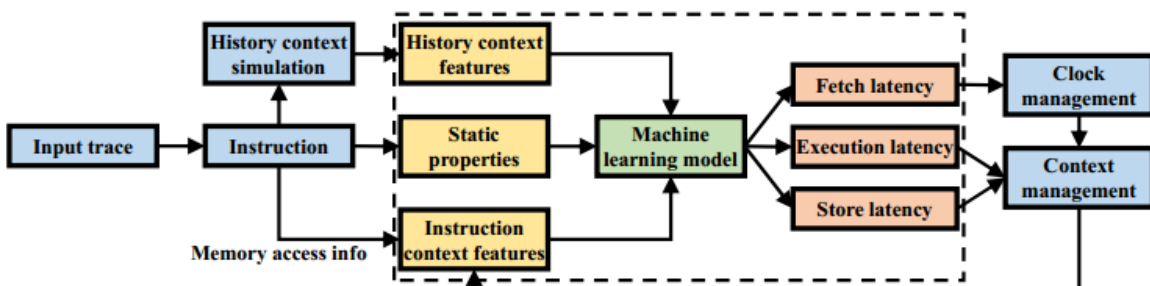
本文旨在探索基于机器学习的计算机体系结构仿真方法的可能性。首先，机器学习模型，特别是深度神经网络，已经被证明是许多领域的优秀函数逼近器，从计算机视觉到科学计算。我们期望它们也可以应用于近似复杂和隐式延迟计算，这对计算机体系结构模拟至关重要。其次，与基于ml的分析建模相比，基于ml的仿真更灵活，因为它不需要对每个程序/输入进行训练。此外，基于ML的模拟器可以带来性能优势，因为ML推理是高度并行的，以及最先进的加速器(例如GPU;TPU)和软件基础设施针对此类任务进行了很好的优化。

SimNet是一种新颖的以指令为中心的模拟框架，它将程序模拟分解为单个指令延迟，并使用定制的ML技术进行指令延迟预测。

程序性能是通过综合所有执行指令的延迟预测结果来获得的。SimNet在保持相同水平的仿真精度的同时实现了明显更高的仿真吞吐量，因为

1. 它将仿真处理器抽象为一个整体，并且消除了对处理器内单个组件进行仿真的需要
2. 它经过很好的优化，可以在gpu上有效地执行。此外，SimNet可以模拟复杂的处理器架构和具有数十亿/数万亿指令的实际应用工作负载。

该模型框架如下：



# 实验结果

本次实验未能完全复现该项技术，因为虚拟机和电脑内存不够，无法完全将gem5安装下来，且编译一次的时间过高（运行了大约两个小时）。接下来汇报本人已经实现的部分：

## 1. 安装gem5

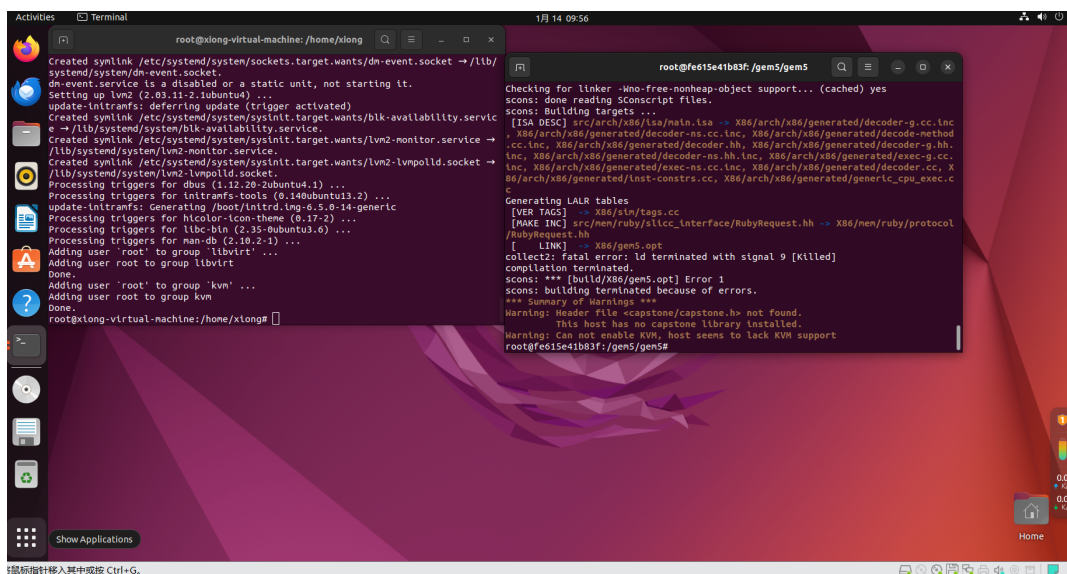
gem5是一款面向计算机体系结构的软件仿真模拟器。在研究计算机体系结构的时候，我们会遇到非常多的概念和方法，动手验证这些方法在处理器中起到的效果是非常困难的一件事情。一般来说，模拟器的模拟层次分为功能级的模拟和时钟级的精确模拟。毫无疑问，设计师希望能得到一个精确的模拟结果，RTL代码就是一种时钟级别的精确模拟，但是因为编写RTL代码是一件费时费力的事情，所以在每次验证时都特地写RTL代码并仿真是不可能的，因此我们需要一个更加便捷且不失强大的模拟工具，而gem5正是这么一个模拟工具。

1. 尝试将gem5在本地编译运行。结果报了一堆错误，始终安装不下来，于是尝试利用docker编译gem5

2. 借鉴下面两篇文章：

- [gem5官方网站](#)
- [2023.11使用wsl2运行gem5 full system ubuntu boot](#)
- [GEM5 full system Parsec tutorial 2024 parsec 全系统教程2024年可运行](#)

最终，还是失败了，可能是因为给虚拟机分配的内存不足导致运行出错：



## 2. 使用SimNet

### 1. 依赖库

Python  
Pytorch  
CUDA (for GPU support)  
Libtorch

### 2. 文件夹作用

- dp:  
包含用于数据预处理和运行模拟器的脚本。
- ml:  
包含用于机器学习训练和测试的脚本。
- sim:  
包含用于生成模拟器的脚本。

data:

链接到数据文件夹。

### 3. 编译运行方法

SimNet 需要来自 gem5 的指令跟踪进行训练和模拟。

修改后的 gem5 可以使用

```
git clone -b simnet https://github.com/lingda-li/gem5.git
```

它在模拟程序执行时生成完整的指令跟踪和存储队列跟踪。

## 训练机器学习模型

第一步是训练一个模型，该模型可以预测给定指令的延迟。

### 准备训练数据

1. Generate a.qq: `./dp/buildQ a.txt a.sq.txt`
2. Deduplicate qq file: `python dp/1_unique.py a.qq`
3. Make npy file: `python dp/2_Q-to-mmap.py --total-rows=<entry num> a.qq`

### 训练模型

按照ml文件夹中的说明训练模型。经过训练的模块将由模拟器使用。

## 编译模拟器

按照sim文件夹中的说明构建模拟器。

## 运行模拟器

为了运行模拟器，我们首先将指令跟踪从 gem5 处理为模拟器格式。按照第 2 节（构建模拟器输入）“dp”文件夹中的说明处理模拟器的输入。

将步骤 2 中所需的所有文件和“模拟器”文件构建放在同一目录中。

注意：使用转换为 libtorch 版本的模型。

```
./simulator_qq <trace> <aux trace> <lat module> <class module> <variances (optional)>
```

或

```
./simulator_qq <trace> <aux trace> <lat module> <variances (optional)>
```

## 问题

在复现该论文时遇到了许多问题：

1. gem5版本问题，导致编译了很多次都会出bug。由于 gem5 开源项目仍然在被持续维护和开发，部分代码结构已经发生变化，有些较早的教程和案例已经不再适用。并且，在 gem5 官网的一些资料和教程，也没有及时更新，导致有些部分内容和代码没有同步。所以本地编译很难实现，浪费了许多时间。后采用docker编译后，虽然大部分编译成功，但还是出现了问题，所以最终实验未能完成。
2. SimNet采用深度学习模型，依赖 gem5，但是由于 gem5 未能成功编译，所以未能将其复现。还需要进一步探索，因为是第一次做论文复现的工作，经验不足，后序会尝试进一步探索该源码的复现工作。

仓库如下：

