**ShengBTE声子计算的性能分析与GPU加速**

摘 要

ShengBTE是计算玻耳兹曼声子输运方程常用的软件之一。虽然该程序可以计算声子散射率的收敛集，并利用它们来获得声子散射率κℓ和许多相关量，但是它的计算过程漫长。我们需要对代码进行GPU移植来实现加速效果。本文深入分析了性能瓶颈产生的原因，提出了相应的代码优化方案。实验结果表明，该方法在不降低精度的前提下，实现了单温度测试下最高4.28倍的加速。

关键词：玻耳兹曼声子输运方程，ShengBTE, 性能分析优化

**Phonon computation performance analysis and**

**GPU acceleration in ShengBTE**

**Abstract**

ShengBTE is a software package for computing the Boltzmann transport equation. Though the program can compute converged sets of phonon scattering rates and use them to obtain κℓ and many related quantities, it takes a long time to compute. We need to port the code on GPU to achieve acceleration effect. In this paper, we thoroughly analyze the cause of the performance bottleneck and propose corresponding optimization for performance speedup. The experiment results show that our approach achieves a maximum speedup of 4.28x without degrading the precision.

**Key words:** Boltzmann transport equation, ShengBTE, Performance Analysis and Optimization

# Introduction

对于热电[1]、热管理[2]和基于相变材料[3]的非易失性存储器的开发等需要具有特定导热系数材料的重要技术来讲，晶格热导率κℓ是一个重要的参数。晶格热导率是声子对总热容的贡献，而研究固体声子输运的一个重要方法是玻耳兹曼输运方程[5]。

ShengBTE是Wu等人提出的一个新的声子玻耳兹曼输运方程的求解软件包。ShengBTE对于寻找具有目标导热性能的新型材料，以及深入理解固体中的热传导实验测量具有重要的价值[1]。该程序可以计算声子散射率的收敛集，并利用它们来获得声子散射率κℓ和许多相关的量。此外，它还实现了作者中的一些人开发的一种近似，以有效和准确地预测纳米线的导热系数[1]。ShengBTE完全由FORTRAN语言构成，是一个仅能运行在CPU上的程序。通过全面的性能分析，我们发现了几个性能瓶颈，并且提出了一部分代码优化与GPU加速方法，显著提高了ShengBTE计算速度。

具体而言，本文的贡献如下：

-分析了ShengBTE的性能，确定了Ind\_\*函数调用的Vp\_\*函数是计算瓶颈。

-提出了几种提高算法性能的优化策略，包括计算冗余优化和GPU加速

-在应用该算法后，我们对ShengBTE的性能进行了评估，程序在不降低迭代次数与计算精度的前提下，实现了最高4.28x的加速。

本文的其余部分组织如下。在第2节中，我们提供了ShengBTE的瓶颈分析。第3节给出了热点函数的优化策略。第4节阐述了实验环境，并对实验结果进行了分析。第5节介绍了相关工作，第6节给出了本文的结论。

# Bottleneck Analysis

## 2.1 Execution Model Analysis

待写。

2.2 Bottleneck Identification

我们使用Vtune分析了计算温度为300条件下的算例热点函数。通过观察发现七个算例的热点产生了两种不同的状态：一种与Sn2Si-F类似，热点非常突出，有六个算例如此；另一种是penta-graphene，算例热点远不如其余六个算例突出。

&插入图 Sn2Bi-F 热点分析比例&

&插入图 penta-graphene 热点分析比例&

# Optimization Strategies

## 3.1 Calculation Redundancy Optimization

## 3.2 GPU Acceleration

|  |  |
| --- | --- |
| **算法 A1** 优化的方向选择算法 | |
| **Input:** *rot\_angles*, *sigma\_rot*, *sigma\_tilt*, *prior\_rot*, *prior\_tilt*, *tilt\_angles* | |
| **Output:** *pointer\_dir\_nonzeroprior*, *directions\_prior*, *best\_ang, best\_idir* | |
| **1:**  **2:**  **3:**  **4:**  **5:**  **6:**  **7:**  **8:**  **9:**  **10:**  **11:**  **12:**  **13:**  **14:**  **15:**  **16:**  **17:**  **18:**  **19:**  **20:**  **21:**  **22:** | *do j=1,Nbands*  *do ii=1,nptk*  *...*  *do k=1,Nbands*  *...*  *if ((omegap.ne.0).and.(omegadp.ne.0)) then*  *...*  *if(abs(omega+omegap-omegadp).le.(2.d0\*sigma)) then*  *N\_plus\_count=N\_plus\_count+1*  *Indof2ndPhonon\_plus(N\_plus\_count)=(ii-1)\*Nbands+j*  *Indof3rdPhonon\_plus(N\_plus\_count)=(ss-1)\*Nbands+k*  *fBEdprime=1.d0/(exp(hbar\*omegadp/Kb/T)-1.D0)*  *...*  *Gamma\_plus(N\_plus\_count)=hbarp\*pi/4.d0\*WP3\*abs(Vp)\*\*2*  *Gamma\_plus(N\_plus\_count)=Gamma\_plus(N\_plus\_count)\*5.60626442\*1.d8/nptk*  *end if*  *end if*  *end do*  *end do*  *end do* |

# Evaluation

## 4.1 Experimental Setup

所有的实验都在本地服务器上进行，使用28进程执行，机器型号为[？浪潮M4]，软硬件配置如下：

|  |  |  |
| --- | --- | --- |
| 操作系统 | 硬件 | 软件 |
| CentOS7.6  (内核3.10.0 x86 64) | CPU 2xIntel E5-2680v4  GPU 2xNvidia P100  Mem 24x16G | GCC v4.8.5  ICC v2018.5.274  CUDA v10.0  IFORT v2018.5.274  Intel VTune v2018.4.0.573462  Intel MPI v2018.4.274 |

## 4.2 单温计算

首先，我们测试了所有优化后ShengBTE运行性能的提升。图【？时间，加速比】展示了我们使用不同的七个算例在温度为300条件下计算得到的时间与加速比，最高得到了4.28x的加速比，但是一些算例表现仅有1.42x。这些算例部分来自ASC19，部分来自almaBTE的算例，所有算例已经上传到【github？】。对于加速比不高的算例，分析源代码与Vtune收集的数据发现penta-graphene不是一个计算密集的算例，在Ind\_\*函数里调用Vp\_\*热点函数的次数远远不如其他算例。总体来看，平均加速比已经达到了3.x。

## 4.3 连温计算

同时，ShengBTE支持连续温度下的计算，我们使用Sn2Bi-F算例进行了以300K开始的步长为100K的多次运行测试。

## 4.4 kernel性能改善

待写。

# Related Work

almaBTE是ShengBTE的继承者。almaBTE编译产生了一组可执行文件，允许用户灵活使用。同时almaBTE保留了一个ShengBTE的模拟器，可以处理与ShengBTE相同的输入。almaBTE产生的模拟器由FORTRAN与C混合生成，而ShengBTE完全是FORTRAN生成。我们在实验中运行单温度测试时发现alamBTE并没有比ShengBTE有运行时间上的缩短。

# Conclusion and Future Work

随着材料技术发展，新的特定性能材料的寻求对现有软件计算性能的优化起到了重要的推动作用。本文对ShengBTE进行了综合性能分析，找出了性能优化的瓶颈。此外，我们还提出了一些提高ShengBTE性能的优化策略，包括计算冗余优化和GPU加速。实验结果表明，在不降低精度的前提下，我们的优化方法实现了单温度测试下最高4.28倍的加速。在未来的工作中，我们希望进行knl重核CPU的移植以及次要热点函数的GPU移植。

# Acknowledgments

本课题由北京航空航天大学中德所资助。通讯作者杨海龙。