从格点量子色动力学应用看国产超算环境的基础软件

宫明[1,2]，蒋翔宇[1,2]，陈莹[1,2]，刘朝峰[1,2]
1. 中国科学院高能物理研究所，北京 100049；2. 中国科学院大学，北京 100049

摘要
格点量子色动力学（LQCD）是用数值模拟方法研究基本粒子的重要科学领域，因其巨大的数据量和计算规模而成为国际上超级计算机的主要科研应用之一。随着国产新一代超级计算机的发展，LQCD的计算软件由于其传统编程模型的限制，面临着更新换代的关键节点。从格点量子色动力学的视角出发，分析大规模科学应用软件对底层基础软件的需求特点，面向国产超算平台的发展方向，提出适配于大规模高效异构计算和大数据处理的新编程模型，为国产超算环境的基础软件建议了一个有潜力的发展方向。

关键词
格点量子色动力学；高性能计算；大数据处理；基础软件；编程模型

中图分类号：O4-39，O411.2，O572.24+3 文献标识码：A doi: 10.11959/j.issn.2096-0271.2021047

Software infrastructures for Chinese supercomputers from the perspective of lattice QCD applications

GONG Ming[1,2], JIANG Xiangyu[1,2], CHEN Ying[1,2], LIU Zhaofeng[1,2]
1. Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China
2. University of Chinese Academy of Sciences, Beijing 100049, China

Abstract
Lattice QCD is a frontier scientific field for studying elementary particles by numerical simulation methods, which has become one of the major scientific research applications of supercomputers. With the rapid development of Chinese supercomputers, the LQCD softwares need to be refactored due to the limitation of its traditional programming model. The characteristics of scientific applications on super computers from the perspective of lattice QCD were reviewed. A novel programming model targeted to Chinese super computers was proposed to adapt large-scale scientific applications with big data processing, which is a promising development direction for the basic softwares of Chinese supercomputing ecosystem.

Key words
lattice QCD, HPC, big data processing, basic software, programming model

1 格点量子色动力学及其应用特征

人类所处的物质世界，从微观粒子到整个宇宙，是按怎样的规律运行的？答案的最新版本自20世纪中期以来逐渐成形，目前其被称作“标准模型”。这个标准模型把万物的相互作用归结为4种：用广义相对论描述的引力，其支撑了宇宙学模型；用量子规范场论描述的强相互作用、弱相互作用、电磁相互作用，其支撑了粒子物理标准模型。

为了找到更深刻的答案，对标准模型的检验工作也在持续进行中。目前绝大多数实验结果与理论计算符合得相当好，但也有一些不确定性，这些不寻常的迹象有可能是新物理理论的滥觞之处。比如最近的一个热点——关于缪子的反常磁矩的测量，实验结果与理论计算似有分歧[1]。然而，微妙之处在于，其中的强相互作用贡献的计算是非常困难的，需要采用超级计算机进行大规模数值模拟才有可能理清结果的系统误差和统计误差。这样的计算采用的理论和方法就是格点量子色动力学（lattice quantum chromo dynamics，LQCD）。

一般说来，电磁相互作用和弱相互作用可以采用“微扰”的方法把计算逐级展开逼近，可以达到很高的精度。但这种方法对于强相互作用，尤其在较低的能量标度下，是彻底失效的。因此，基于计算机数值模拟的LQCD几乎成为唯一有效的方法。在人类寻找超出标准模型的新物理理论的征途上，它的重要性不言而喻。这些年来，在国际顶级超级计算机上进行的自然科学计算中，LQCD往往是占用计算资源最多的应用。

LQCD把时空切分成四维的格子，把闵可夫斯基时空上的量子场论问题转化为欧几里得时空上的统计问题。在这里，时间和空间是等同对待的，而且都是分立且有限的，可以用外推的办法把有限的计算推广到无限的自然世界上。把代表夸克的（伪）费米子场定义在时空格点上，把代表胶子的规范场定义在相邻格点之间的连线上。每个格点上的费米子场可以用12个复数表示，每条连线上的规范场可以用3×3的复矩阵表示，这样的模型就可以在计算机上用内存对象来实现了。

LQCD的计算流程是：先产生规范场组态，再基于这些组态计算费米子的传播子，然后用这些传播子构造各种关联函数，最后从这些关联函数中拟合抽取目标物理量。这个过程涉及几十种复杂的算法，计算规模和中间数据非常庞大，耗时常以年计，数据文件规模常以PB计。然而，这些不同算法的热点函数颇为集中，数据虽大但非常规整，这为软件设计和优化提供了重要的有利条件。

LQCD的主要热点函数可分为两类：以Dslash函数为核心的大规模稀疏矩阵求解、以关联函数缩并为代表的大规模张量计算。

稀疏矩阵求解的典型阶数为10^8~10^9数量级，由于物理上的近邻相互作用的特点，矩阵往往是稀疏的（也有不稀疏的情况，可以用稀疏矩阵的分式级数来逼近）。在其他研究领域的一般情况的稀疏矩阵往往是不规则的，需要在运行时动态寻址，优化难度较大[2]。而LQCD的矩阵非常规则，数据的并行切分和矩阵元的构造都很方便直观，有较大的潜在优化空间。因此类似于很多其他差分问题，LQCD的矩阵操作往往写成模板计算（stencil）的形式，只不过是更复杂的四维九点的模板计算。在具体的实现中，往往采用“内部+外晕”的数据模型，并采用异步通信实现计算与数据传输的重叠。矩阵求解算法一般采

用Krylov子空间迭代算法，并配合各种预处理算法，如多重网格算法和域分解算法等。这部分的常用算法非常多样，如LQCD特殊的multi-shift算法、奇偶预处理技术等。

相对来说，张量计算比较清晰简单，一般可以用爱因斯坦求和记号简洁地进行描述，在程序中的实现往往是多重循环配合对称并行的。但考虑到它实际消耗的计算资源的比例，现有的软件大多严重地低估了这类函数，并没有给出通用的接口，针对性的优化也比较少见。

2 格点量子色动力学计算软件的现状

国内外的每个LQCD研究合作组都拥有自己独特的代码积累，其中一些较通用的部分往往以开源的方式与其他合作组共享。美国USQCD合作组在美国SciDac经费的支持下开发的USQCD软件集是一个相当成熟完善的开源范例，形成了一定程度上的事实标准。这个软件集包括4层框架，每层有若干特定功能的软件，它们按照接口协议相互协作，可实现复杂且高效的计算功能。

USQCD软件集的最底层包括负责基本线性代数操作的QLA（QCD linear algebra）、包装多进程和多线程并行的QMP（lattice QCD message passing）和QMT（QCD support for multi-thread）；第2层包括QDP（SciDAC QFT data parallel library）协议的各种实现、包装文件输入输出的QIO（QCD input/output applications programmer interface）和c-lime；第3层包括实现稀疏矩阵求解的各类算法；第4层负责整合功能，并提供人机接口，比如Chroma提供了一个集成框架和XML交互协议，QLua为QDP协议提供了一个Lua语言的包装，CPS（columbia physics system）以C++库包的形式提供了编程接口。

在这个4层框架中，最关键的特点是USQCD独创的QDP协议。QDP为上层提供了一个方便的编程模型：在这里，规范场或费米子场等数据可以被自动拆分，并被安排到不同的计算节点上，所有的操作也都是单程序多数据（SPMD）风格的并行实现，然而这些实现都被包装在一些C++对象里，上层程序不需要考虑任何有关并行的细节。考虑到LQCD数据往往都是四维的场，QDP利用四维时空指标切分数据，并提供了全局的平移（shift）操作。QDP协议有C语言和C++语言的实现，后来又有了采用运行时编译技术的C++实现，即目前最常用的QDP-JIT。

随着GPU在高性能计算中的普遍应用，在NVIDIA公司的大力支持下，QUDA逐渐成熟。QUDA在USQCD框架中的定位是第3层，即它充分利用GPU的性能实现了各种稀疏矩阵的高效求解。然而，为了充分利用GPU硬件的特点，QUDA自身实现了大量底层功能，击穿了下面的两层协议，偏离了USQCD框架关于分层解耦的设计初衷。

另外一个流行软件Grid试图代替QDP重新构建一个框架。Grid的一个设计特点是面向CPU的向量指令，不仅把四维时空的切分放在不同的进程上，也放在向量指令的内部。这个方法可以实现大量程序的自动向量化，而且对于不同字长的向量指令具有良好的可移植性。后来，随着GPU的发展，Grid也提供了对GPU的良好支持，它采用一些宏和预编译指令精巧地实现了跨平台运行。

中国格点合作组（China Lattice QCD Collaboration，CLQCD）在近十几年的科学研究中，积累了很多特定功能的计算代码，但没有一个成体系的软件框架。随着

国产超级计算（以下简称超算）的发展，这个问题在代码研发和移植过程中越来越凸显。

近年来，研究人员基于神威·太湖之光超算从零开始研发了神威格点原型软件（SWLQCD），通过优化流程细节、嵌入汇编、手工向量化等细致的工作实现了较理想的运行效率[3-4]。但由于目前没有国产的软件框架，只能通过制作QSunway接口与QDP协议连接，然后把SWLQCD作为一个模块嵌入USQCD的Chroma软件内，实现与上层的其他算法协作运行。

同时，笔者团队也在“天河三号”原型机上移植了Chroma和Grid，并针对国产处理器的向量指令进行了优化[5]；针对曙光的新超算环境，移植了QUDA，把原本面向CUDA的代码改写成面向HIP的代码[6]。

目前，在国产的3个超算环境上，都实现了LQCD软件的研发或移植。然而，由于软硬件环境与国外区别很大、国产环境的不完善，开源软件的移植效率、自主研发的可持续性等都是研究者面临的严峻挑战。

3 编程模型的演化与国产环境的特点

软件更新换代的动因往往是编程模型的革新，而底层编程模型依赖于硬件的实现。随着计算机硬件的不断发展，对应的编程模型和基础软件应运而生，应用层的软件面临更替或重构的挑战。近年来，国产超算的硬件平台发展速度非常快，至少有3个主要的架构在并行发展，而对应的编程模型和基础软件尚未跟上步伐，这使得应用层的软件无所适从，难以做到高效且可移植。在多种异构环境下，提出新的编程模型，并将其实现为高效且可移植的标准库，这是国产环境的软件生态发展的关键环节。

纵观LQCD软件的发展历程，这个脉络是很清晰的。

20世纪末，随着向量机的退场，Beowulf集群成为大规模高性能计算和大数据处理的主流平台。在这样的集群下，节点之间通过网络互联，无法直接访问远端内存，天然适合多进程和消息传递的并行模型。因此，各合作组研发的LQCD软件都开启了MPI（message passing interface）并行化的变革。随后，片上对称多核处理器的流行推动了共享内存的线程模型，可以很方便地用OpenMP预编译指令指导热点循环块的本地并行化。USQCD的QDP协议就是在这个时候抓住了关键的抽象要素——把对称的数据放在对称的处理器上，设计了一套针对LQCD计算的SPMD抽象层，从而封装了单进程、多进程、多线程的各种底层实现。

Grid的设计大约是在Intel公司提出Xeon Phi系列处理器的时候进行的，此时各种不同的单指令多数据流（SIMD）指令对代码的高效移植已形成了很大的负担。考虑到SIMD指令的向量维度其实也可被看作对称多处理器的一个新维度，因此可以把时空点在这个维度上进行切分。这个思路启发了Grid的设计与实现。

然而，Intel公司并没有继续发展融核技术，目前美国最新的超级计算机都是基于CPU+GPGPU架构的。因此，用CUDA编程模型开发的QUDA逐步走上前台，成为在GPU上计算LQCD的首选软件。

随着C++标准的不断升级，C++能够承载的编程模型更加丰富。ROCm/HIP是在C++语法基础上重建的一套类似CUDA的编程模型和相应的基础库，可以适配AMD公司和曙光公司的GPU。未来C++标准中有可能会加入异构计算的支持，目前比较接近的编程模型是SYCL和Kokkos，它们的特点是把匿名函数作为

核函数下放给加速设备去运行。这两个编程模型下的LQCD原型程序已经实现了初步的版本。

然而，我国LQCD软件的未来发展路径需要走向另一个方向，这是由国产软硬件环境决定的。

美国的几个商业公司分别正在推动CUDA、HIP、SYCL几个编程模型的互相交叉兼容。因为美国最先进的超算都是基于CPU+GPGPU的硬件模型的，所以编程模型也是类似的。又由于对CUDA的路径依赖，HIP需要兼容CUDA，SYCL需要兼容CUDA和HIP。它们的编程模型有一个基本的共同之处：都是基于“主-从”模型的，默认的主控代码运行在CPU端，它可以控制加速设备执行热点函数。

我国的超算有3个并列发展方向，软硬件环境各不相同，编译器对语法标准的支持也很有限，因此没必要盲目跟随SYCL等编程模型。由于没有路径依赖的历史包袱，我国的基础软件研发可以面向更长远的目标，寻找最适合未来国产超算发展方向的新编程模型。

国产申威处理器是片上异构的设计，包含若干核组，每个核组包含一个主核与64个从核。软件环境提供的是Athread编程接口，这个编程模型是面向硬件特性设计的，因此抽象程度不足。虽然Athread中也用主核调度从核，但研究人员在实际的代码研发实践中发现，把大部分逻辑留给从核处理是很好的策略，主核只需要被动地做一些数据传输和输入输出等辅助性的工作。代码设计的视角和重心在从核一侧，这与传统异构编程的思路有所不同。

日本的超级计算机“富岳”采用的处理器也是片上异构的，每核组也是一个辅助核心和一组计算核心阵列的组合。

实际上，即使是在GPU硬件环境下，这种不协调的感觉也已经浮现。由于PCIE（peripheral component interconnect express）的带宽有限，在多GPU节点上，NVIDIA公司采用NVLink或NVSwitch来实现GPU之间的直接通信，这种做法绕过了CPU和主内存。另外，统一内存映射和远程直接数据存取（remote direct memory access，RDMA）也弱化了CPU在数据传输中的重要性。因此，未来的代码会不可避免地向加速设备倾斜，目前流行的以CPU代码为主线的编程模型并非最优之选。

国产的神威、天河、曙光3个系列大规模超算平台的共同点是异构。异构意味着计算单元和存储单元都不是单一、对称的，它们之间应当有丰富的分工协作，未必是单纯的“主-从”模型。笔者期望，下一代编程模型应该更细致地刻画不同的处理器和内存资源，并定义出通用的控制权转移和数据传输的规则和相关接口。

LQCD计算的性能瓶颈经常体现在数据流动上。代码优化的重点经常是节点间的MPI传输的时机、CPU缓存的合理复用、GPU显存的充分安排、申威从核直接内存访问（direct memory access，DMA）的错峰传输等。然而，目前的高性能计算的编程模型很少有面向数据流动和异构内存空间的支持，绝大多数细节需要手动安排。申威和NVIDIA公司都为各自的硬件实现了OpenAcc编程模型，这对于初学者而言入门方便，但也阻挡了优化的可能性，无法支撑优秀的应用。笔者期望，下一代的编程模型应当在数据流动方面做出更合理的抽象，使得这方面的优化可以被自动或半自动地完成。

4 应用软件层与基础软件层的新编程模型

把前述两个对下一代编程模型的期望

进行综合考虑，笔者发现符合要求的编程模型难以用库包或语法设计实现，它很可能更像一个大数据处理引擎，只不过它的粒度比Spark等传统引擎要细得多，要直接深入硬件细节中，才能保证极限的计算效率。为了配合一个引擎风格的编程模型，应用层的程序需要被拆分成很多独立的算子，这些算子根据数据依赖关系构成了有向无环图，被引擎依序分配给不同的资源进行计算，这种做法与传统的大数据处理引擎相同。

由于算子的粒度较小，算子切换的代价需要被极限优化，这在不同硬件环境上可能需要完全不同的优化技术。在国产申威架构下，从核有能力独立处理复杂的任务切换，但多从核互相配合进行异步操作时的灵活性不足。这是因为从核是单线程的，没有天然的并发支持。为此，笔者团队做了一个试验性的代码OSP，它是在程序块粒度上的超轻量级协程库，能够支持多个伪多线程并发，并提供了私有变量支持。虽然笔者团队采用了一些非常特殊的语法技巧，但全部代码完全符合C89标准语法，可移植性非常高。笔者团队利用OSP在申威平台上为从核DMA做了异步包装，初步测试成功。与此同时，笔者团队还设计了一套被称作虚拟机生成器（CVM）的优化框架，采用遗传算法为64个申威从核寻找最优的静态任务调度方案，相比人工方案，优化了13%的效率。这两项工作分别成功验证了动态调度和静态调度的可行性，为今后在各类国产平台上建立轻量级的任务调度引擎奠定了基础。

与粗粒度的大数据处理引擎不同，贴近硬件的细粒度引擎需要细致地考虑各种不同内存之间的数据传输，包括主内存（可能有核心绑定）、多级缓存、GPU的多种不同访问模式的显存、申威从核便笺存储器（LDM）等。目前不同软硬件环境的接口各不相同，没有统一的公共编程模型，这是需要努力创新的地方。

一般来说，通信有两种效果：数据传输和控制流同步。从形式上看，MPI和申威寄存器通信等基于消息传递模型的方式是显式的数据传输、隐式的同步；而线程模型的共享内存方式是显式的同步、隐式的数据传输。目前看来，后者似乎更有希望成为未来高性能计算的主流，MPI-3版本也增加了共享内存的接口。考虑到访问局部性对效率至关重要，完全的统一编址并不符合人们的期待。在目前已有的访问模型中，远程内存访问（remote memory access，RMA）可能是比较适用且高效的。

那么，新的编程模型需要提供对不同内存的统一描述，对远程数据的读写提供统一的接口。异步的远程读写往往会有一些标志用于判断传输完成情况，这就给任务调度引擎提供了一个重要的信息：它可以根据当前数据的完备情况安排算子的运行。

在这样的编程模型下，应用层的软件编写应当分为两部分：一部分是各类算子的代码实现，另一部分是描述性地列出数据依赖关系和传输的拓扑关系。这种编程模型与目前常见的模型迥然不同，它是面向数据的，代码是连接数据的桥梁，是零散的。

这样的编程模型有利于极限编译优化和高效运行，但对于大部分程序员来说并不友好。因此，还需要另外一个面向应用层的编程模型，把底层模型进一步抽象化，让它符合用户的思维方式。

回顾LQCD的计算需求：大部分计算热点函数可以描述为张量表达式。求解大规模稀疏矩阵时，此稀疏矩阵作用在向量上的操作也可以写成张量表达式的形式——在物理理论里就是如此表述的。从计算机程序的角度看，稠密的张量就是多

维数组，稀疏的张量往往是简单的枚举函数，用这两种简单的元件可以构筑出LQCD绝大多数的计算过程。可以进一步猜想，张量运算形式的描述能力可能覆盖了各领域的科学计算中的很大一部分，做好张量计算的封装和优化可能会为科学计算提供强有力的支持。

在LQCD的计算实践中，因处理数据的需求，笔者团队设计并实现了QScheme语言。它在Scheme语言的基础上，增加了xdata数据格式和对应的xfile文件格式。这两种格式本质上就是“带有元数据的多维数组”，这个元数据包括数组的维数、每个维度的名称、每个维度的指标列表。笔者团队为xdata实现了丰富的操作功能，这些操作的基础是xdata四则运算的规则：

- 运算结果的维度集合是各参数的维度集合的并集；
- 运算结果的每个维度的指标集合是含有这个维度的参数的对应指标集合的交集；
- 程序实现保证对各参数维度的对应指标分别进行计算，与它们在内存中的存储顺序无关；
- 普通数值被视作包含0个维度的xdata。

这个规则可以被视为扩展的张量运算规则（与数学规则相比，多了一个“自动丢弃无法匹配的指标”的例外规则），可以把那些平时不被视为张量分量的维度纳入张量处理的框架中，进一步提高了模型的适用性。

笔者团队使用QScheme进行了长时间的实际科研工作，发现xdata的表达能力强、优化方便，非常适合科学计算和数据处理。因此，前述的底层编程模型如果能够进一步被抽象为类似xdata的张量计算模型，会非常适合科研人员使用和进行二次开发。

张量表达式的形式简单、规则明确，可以进行静态分析，进而能够用程序估算出表达式的计算资源需求，包括浮点操作计数、内存占用等预算信息。可以基于项或指标拆分表达式，利用子表达式的资源预算找到最优的拆分方案，然后用元编程技术把子表达式输出为算子的计算程序，把数据拆分情况整理成数据描述信息，最后把这些信息传递给下层的调度引擎。这样，两层计算模型之间就可以用一套静态分析算法和代码生成算法连接起来。

至此，笔者团队期待的高性能大数据处理的基础软件已逐渐成形：它包含上下两层编程模型，下层实现高效跨平台的远程内存访问接口和算子调度引擎；上层是在此基础上构建的扩展张量计算模型，为研究者提供张量表达式形式的代码接口。

5 结束语

本文回顾了LQCD的计算特点和软件沿革，基于国产超算的软硬件环境和发展方向提出了面向LQCD和其他科学计算的上下两层编程模型。笔者团队研发了试验性的软件，用原型测试和实际使用的经验为这个设想中的编程模型提供了可行性的支撑。这个编程模型的实现既是国产环境基础软件的一项挑战，也是繁荣软件生态的一个可能的契机。

参考文献:

[1] ABI B, ALBAHRI T, AL-KILANI S, et al. Measurement of the positive muon anomalous magnetic moment to 0.46 ppm[J]. Physical Review Letters, 2021, 126(14): 141801.

[2] 胡正丁, 薛巍. 面向异构众核超级计算机的

大规模稀疏计算性能优化研究[J]. 大数据, 2020, 6(4): 40-55.
HU Z D, XUE W. Research on performance optimization for large-scale sparse computation over many-core heterogenous supercomputer[J]. Big Data Research, 2020, 6(4): 40-55.

[3] 张淼, 周宇, 陈建海, 等. LQCD Dslash在神威·太湖之光上的研究分析与MPI实现[J]. 计算机科学与探索, 2019, 13(10): 1664-1676.
ZHANG M, ZHOU Y, CHEN J H, et al. Analysis and MPI implementation of LQCD Dslash on Sunway TaihuLight[J]. Journal of Frontiers of Computer Science & Technology, 2019, 13(10): 1664-1676.

[4] ZHANG Z X, LUAN Z Z, XU C Y, et al. Accelerating lattice QCD on sunway many-core processor[C]//Proceedings of 2018 IEEE International Conference on Parallel and Distributed Processing with Applications, Ubiquitous Computing and Communications, Big Data and Cloud Computing, Social Computing and Networking, Sustainable Computing and Communications. Piscataway: IEEE Press, 2018: 605-612.

[5] 毕玉江, 周超, 吴郁飞, 等. 格点量子色动力学Grid数值模拟软件的并行计算特征分析[J]. 计算机系统应用, 2020, 29(7): 199-204.
BI Y J, ZHOU C, WU Y F, et al. Parallel computing feature analysis of Grid numerical simulation software for lattice quantum chromodynamics[J]. Computer Systems & Applications, 2020, 29(7): 199-204.

[6] BI Y J, XIAO Y, GUO W Y, et al. Lattice QCD GPU inverters on ROCm platform[C]//Proceedings of the 24th International Conference on Computing in High Energy and Nuclear Physics. [S.l.:s.n.], 2020: 09008.

作者简介

宫明(1981-), 男, 博士, 中国科学院高能物理研究所副研究员, 主要研究方向为格点量子色动力学。

蒋翔宇(1996-), 男, 中国科学院大学博士生, 主要研究方向为格点量子色动力学。

陈莹(1970-), 男, 博士, 中国科学院高能物理研究所研究员, 主要研究方向为格点量子色动力学。

刘朝峰（1977– ），男，博士，中国科学院高能物理研究所研究员，主要研究方向为格点量子色动力学。

收稿日期： 2021–05–31

通信作者： 宫明，gongming@ihep.ac.cn

基金项目： 国家重点研发计划资助项目（No.2017YFB0203202）；国家自然科学基金资助项目（No.11775229，No.11935017，No.12075253）

Foundation Items: The National Key Research and Development Program of China(No.2017YFB0203202), The National Natural Science Foundation of China(No.11775229, No.11935017, No.12075253)