

工业软件研发中处理超大模型(5)--求解器通用篇

原创 邓子平 [多物理场仿真技术](#)



[工业软件研发中处理超大模型](#)

[工业软件研发中处理超大模型\(2\)](#)

[工业软件研发中处理超大模型\(3\)](#)

[工业软件研发中处理超大模型\(4\)](#)

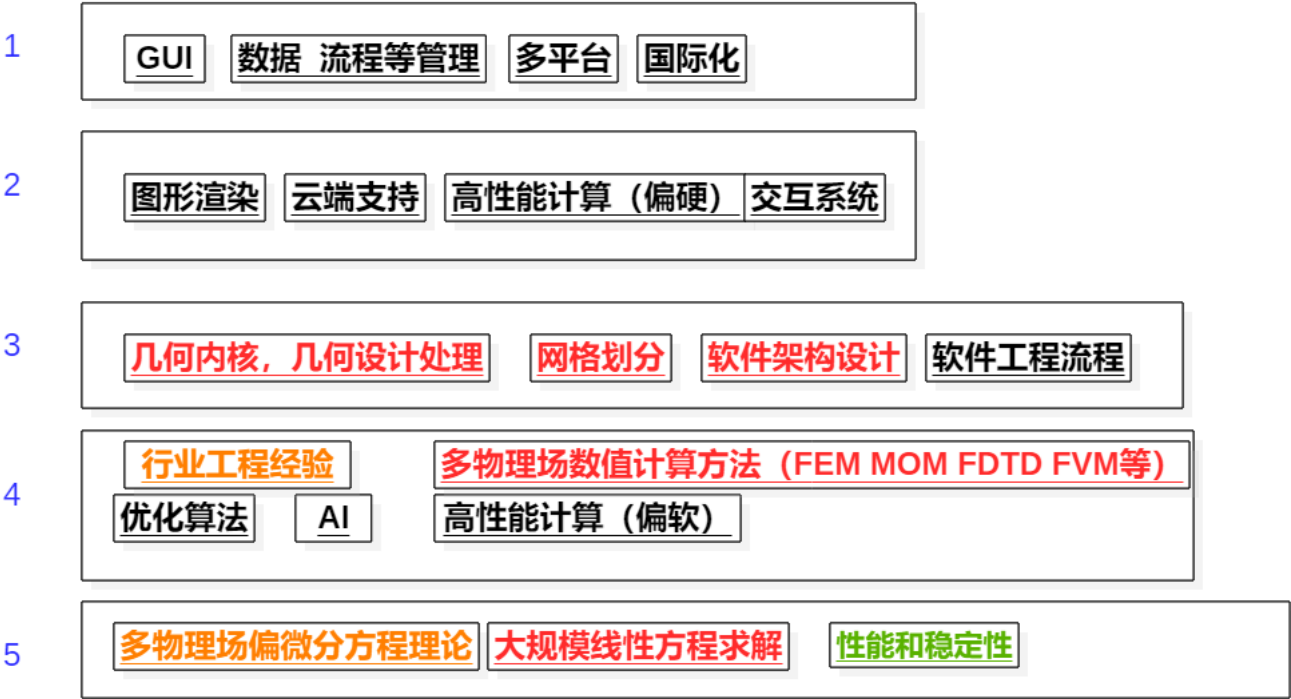
[工业软件研发中处理超大模型\(5\)--求解器通用篇](#)

全文约2500字，仔细阅读约10分钟

刚关注的朋友建议先阅读下文：

[工业软件的行业技术框架](#)

在以上文章中(点击链接查看)介绍了在前后处理中处理超大规模模型，本文以及后续将讨论在[求解器中如何求解超大规模的模型](#)，在硬件和软件介绍中[偏软件部分](#)。超大模型的求解是工业仿真软件核心的核心，在模型处理生命周期中，大部分时间都花在了[求解器运行](#)上。对于工程师来讲，下班点击“开始仿真”，第二天上班或者过个周末再来查看结果是常态。



多物理场仿真技术

图1

对于研发而言，从整体框架看，超大模型求解主要涉及到以下几个方面：

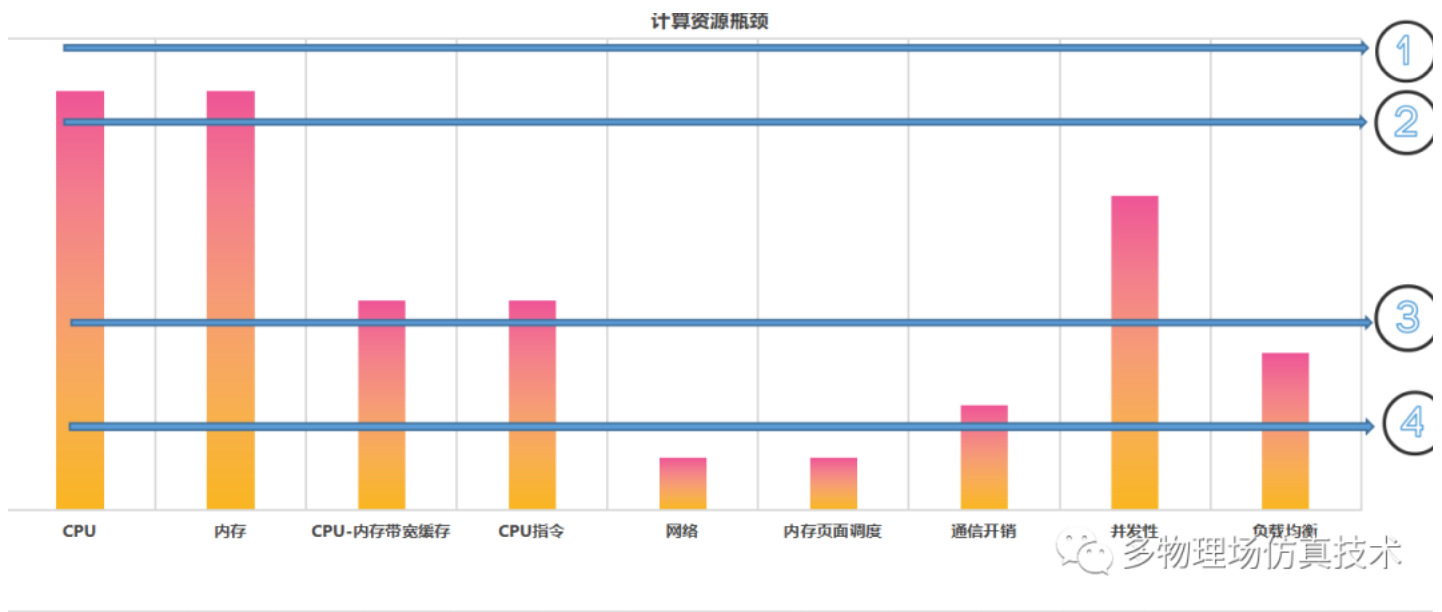
1. 硬件资源

即如何有效的利用（压榨）所有硬件资源，包括：

1.1. 硬件本身资源合理利用，例如内存页面缓存，CPU优化指令

1.2. 硬件和硬件之间瓶颈，比如CPU-内存之间带宽，缓冲瓶颈

1.3. 硬件资源调度，硬件资源负载分配，负载均衡等，把软件的资源调度归到高性能计算硬件资源这块。在图1中，高性能计算（偏硬）。



参考上图，将仿真计算中对硬件依赖简单分成四层：

第一层，对于小模型我们无需考虑硬件的情况，即不需要做硬件方面的设置，普通台式机即能完成仿真。也就是我们通常所说的“仿真还没有受到硬件的限制”。

第二层，当模型规模上来后，求解时间大幅增加，性能瓶颈开始显现，主要表现在对CPU和内存的依赖上，此时可以通过更新CPU，扩大内存容量提升性能。但是主机上CPU和内存容量会有上限，不可能持续增加，性价比也会降低，因此到一定程度要求第三层，需要使用集群分布式计算，同时CPU-内存带宽瓶颈，命中率，内存页面缓存等问题显现，对CPU指令性能也提出要求。

第四层，当使用超大规模集群进行计算时，并行性，负载均衡，网络，通信开销等问题出现，在资源调度方面对软件开发提出了要求。

除了常规的硬件，可以利用性价比高的专用硬件，最常用的GPU，TPG。

2. 框架并行性

通常我们聚焦于大规模线性方程组的求解方法，事实上对超大模型而言，在现有硬件条件下，计算必须使用分布式计算。常见情况是数据无法一次性读入内存。

（win95时期硬件内存一般只有32/64M，在申请内存时有可能出现硬件内存不够而分配失败的情况）。

从硬件角度看，硬件关心的是数据的整个流程的读入，计算，存储和传输效率以及硬件和硬件之间的协调。以矩阵操作为例，框架并不关心大规模矩阵的分解，分块，求逆等算法操作，因为算法底层最后都是简单的矩阵加减乘(乘法也是加法)操作。

以超算为例，单个硬件的数量可能达到百万级，仅硬件的管理就可能造成瓶颈。数据量大后硬件管理是个运筹学问题。

框架的并行性主要考虑资源的分配，资源调度，沟通开销，根据各自硬件的特性，尽可能利用已有硬件资源。

通常的多线程，多进程方法，以及各种

OpenMP，OpenMPI，MPICH，CUDA等工具可以看作是框架并行性的具体体现。

3. 算法并行性

从软件的角度看，算法的并行性就是如何将大的数据尽可能的分块，因为模型之间相关性以及计算顺序问题，这种分块不单是矩阵层面，更是模型数据和仿真业务层面，之前有过简要介绍，参考(点击链接查看)

[一篇文章入门求解器模型降阶方法](#)

后续会结合硬件资源使用，重点介绍各种大规模线性方程组迭代解法。

[Krylov子空间迭代方法：历史](#)

针对整体模型求解，通常不考虑常规(算法复杂度高于 $n*n*n$)的求解方法。

4. 高性能C++编程

常规编程中，可能对一些C++细节要求并不是很高，比如类似字节对齐，除法改乘法，数据遍历查找，谨慎使用智能指针等，有些编译器会帮助优化代码，比如将乘法优化成位操作等。在大模型处理时，这些细节改进可以带来明显的性能提升。关于高性能C++编程，网上多有描述，后续有空给出详细总结。

我们列举**五种有代表性的数值方法**来说明超大模型的求解。后续将分篇详解给出每种方法的求解策略。

1.有限元

有限元是目前工业软件仿真领域最常用的数值方法，没有之一。其特点是对求解对象划分网格，对每个网格单元构建刚度矩阵，然后将所有网格单刚矩阵组装整体刚度矩阵，将各种边界，激励，荷载加入其中，最后形成一个线性方程组，其中系数矩阵稀疏，有些对称，有些不对称。显式动力学没有线性方程组。该方法优点是适应性好，通用性强，没有太多前提要求。参考（[点击链接](#)）

[一篇文章入门多物理场有限元（全篇）](#)

后续求解器计算中，将介绍

MKL,PETSc,MUMPS,OpenBlas,HYPRE等各种迭代数值计算库在超大模型求解中的应用。这些库分别在硬件和软件层面都做了不少优化。

2.时域有限差分

时域有限差分(FDTD)可以理解有限差分的一种，由于求解思想天然符合电磁场传播特性，是求解电磁问题的一种有效方法。在求解空间分布各向正交连续点网格(Grid)，通过时间上电磁节点不断交替前进求解每个节点数值。该方法主要计算量集中在时域上每个节点的迭代，整体框架上不要求解线性方程组。（[点击链接查看](#)）[一篇文章入门时域有限差分方法（FDTD）](#)

3.边界元(BEM)/矩量法(MOM)

边界元是一种将网格划分在边界上的求解方法，也就是三维计算只需面网格，二维计算只需线网格。边界元是一种数值解和解析解结合的方法，计算需满足一定条件。借助于解析解，可以大幅减少最终线性方程组的规模。但最方程组的系数矩阵**一般满秩**，需要借助快速多极子，多层快速多级子等方法提升求解效率，加大了开发难度。矩量法也是一种半解析半数值解法，最终形成的也是满秩矩阵。

[一篇文章入门边界元方法](#)

4. 有限体积(FVM)

有限体积是CFD计算最常用的方法，在大规模计算中对网格划分子块的方法支持比较友好。

5.格子玻尔兹曼(LBM)

LBM是最近十几年逐渐在工程领域流行的求解CFD问题的方法。该方法在介观尺度离散玻尔兹曼方程，不需要划分网格，并且易于并行化，前处理效率高，在一些非常规CFD计算领域有较高的计算精度，是一种典型的**无网格方法**。

[一篇文章入门格子玻尔兹曼方法（LBM）](#)

这五种数值计算方法基本涵盖了目前常用的大模型求解场景，即超大规模稀疏矩阵，稀疏对称矩阵，满秩矩阵，有网格无需解线性方程组和无网格无线性方程组迭代计算。

之前简单介绍过HPC，即高性能计算，超大模型的计算和HPC有关联，但并不等价([点击链接参考](#)):

[HPC高性能计算知识: 计算架构演进](#)

[HPC简介（软件）](#)

本文简单介绍了超大模型求解器求解的一些特点，可以看出在现有技术条件下，超大模型求解对硬件资源依赖程度较高，在算法层面也不是简单依靠数值计算方法，更多的是[针对实际数据特点](#)，[对软硬件资源](#)，[数值方法](#)，[整体框架](#)，[计算策略](#)等综合性考量。

阅读: null

在看: null