

doi:10.11835/j.issn.2096-6717.2019.043



基于阿里云的四维弹簧模型并行运算性能

赵高峰, 陈华

(天津大学 建筑工程学院; 水利工程仿真与安全国家重点实验室, 天津 300072)

摘要: 四维弹簧模型(Four-Dimensional Lattice Spring Model, 4D-LSM)是一种考虑额外维相互作用的新型离散数值计算方法。该方法用于岩石破坏分析需要消耗大量计算资源,不适合在普通个人电脑上运行。基于多核并行技术,在阿里云和多核工作站等多种硬件环境下对 4D-LSM 的计算极限性能及瓶颈进行详细分析,主要研究了求解规模、求解类型、线程数、硬件配置等对 4D-LSM 求解效能的影响。研究发现,内存容量决定可计算的模型规模,弹性问题的计算时间与模型规模成正比,并行计算效率受 CPU 性能和内存带宽的共同影响。在不考虑经济因素的情况下,云计算在多核匹配和内存分配方面的灵活性特别适合于四维弹簧模型的并行计算分析。结果表明:基于阿里云的 4D-LSM 最大运算规模可以达到十亿单元,由于目前的瓶颈在于前后处理,4D-LSM 目前的可分析规模仍然限制在两千万单元。最后,展示了采用极限规模的并行四维弹簧模型求解三维币形裂纹扩展的实际应用案例。

关键词: 云计算; 四维弹簧模型; 并行计算; 三维裂纹扩展

中图分类号: TU452 **文献标志码:** A **文章编号:** 2096-6717(2019)03-0001-10

Performance of the parallel four-dimensional lattice spring model using Alibaba cloud

Zhao Gaofeng, Chen Hua

(School of Civil Engineering; State Key Laboratory of Hydraulic Engineering Simulation and Safety, Tianjin University, Tianjin 300072, P. R. China)

Abstract: Four-dimensional Lattice Spring Model (4D-LSM) is a newly developed discrete numerical method considering the extra-dimensional interaction. The method needs large amounts of computing resources in three-dimensional rock failure analysis and therefore is not suitable for the conventional personal computer (PC). In this work, based on the multi-core parallel technology, the computational performance and bottleneck of 4D-LSM were analyzed in details. A variety of hardware environments, such as Alibaba cloud, multi-core PC, and multi-core workstation, were selected to investigate effects of the model size, problem type, thread number and hardware configuration on the parallel computing performance. It is found that the memory capacity determines the limit size of the computable model, and the computational

收稿日期: 2018-09-30

基金项目: 国家自然科学基金(1177020290); 国家重点研发计划(2018YFC0406800)

作者简介: 赵高峰(1982-), 男, 教授, 博士生导师, 主要从事岩石动力学、多尺度方法和高性能数值计算研究, E-mail: gaofeng.zhao@tju.edu.cn.

Received: 2018-09-30

Foundation item: National Natural Science Foundation of China (No. 1177020290); The National Key Research and Development Program of China(No. 2018YFC0406800)

Author brief: Zhao Gaofeng(1982-), professor, doctoral supervisor, main research interests: rock dynamics, multiscale computational method and high performance computing, E-mail: gaofeng.zhao@tju.edu.cn.

time of the elastic problem is proportional to the model size. The parallel computing efficiency is affected by both the CPU performance and memory bandwidth. The flexibility of cloud computing in multi-core matching and memory allocation is especially suitable for parallel computing of 4D-LSM without considering the economic factor. Through analysis, it is found that the maximum size of 4D-LSM based on Alibaba cloud can reach 1 billion particles. However, due to the bottleneck lies on the pre-processing and post-processing, the current maximum capacity of 4D-LSM is still limited to 20 million particles. Finally, as an example, 4D-LSM was used to solve a three-dimensional coin-shaped crack propagation problem.

Keywords: cloud computing; 4D-LSM; parallel computing; 3D crack propagation

由于具有可重复、经济及参数可控等优点,数值模拟已经成为理论分析和物理实验之外的第3种研究方法^[1-3]。数值模拟不但被广泛应用于隧道等地下工程在不同工况下的稳定性分析,而且被用来研究特定工程灾变问题的内在力学机理。例如,唐春安等^[1]采用 RFPA 软件对隧道洞室周边的分区破坏机理和演化规律进行了研究,吴顺川等^[2]采用三维离散元软件研究了隧道岩爆的机理。上述两个数值模拟案例分别采用了基于连续介质的有限元法和基于离散介质的离散元法。连续介质方法的基本思路是先建立求解对象的偏微分方程,然后,通过数学离散方法求解,是一种自顶向下的方法。由于其连续假设,连续介质方法在求解动态断裂等问题时具有一定的局限性。将损伤模型引入连续介质方法可增强其求解渐进动态破坏的能力,例如 RFPA 软件^[1]及 LSDYNA 软件^[4]采用了类似单元生死法来实现对材料渐进破坏的描述,并已成功应用于岩石动态破坏及实际工程的计算。然而,这种关于破坏的处理方式没有精确考虑破裂面的形态以及再接触,因此,该方法对破坏后的描述可能会有偏差^[5]。基于离散介质的数值方法则考虑了破裂面的分离和再接触,更适用于岩石动态破坏问题,其中,最为著名的是 Cundall 等^[6]提出的离散元和 Shi^[7]提出的非连续变形分析方法 DDA。作为离散数值计算方法的一种,Lattice Spring Model (LSM)由亚历山大博士于 1941 年最早提出^[8],但由于泊松比限制问题,LSM 发展一直很缓慢。为了解决该问题,Distinct Lattice Spring Model (DLSM)引入了多体剪切弹簧^[8],该模型已被成功应用于岩石与煤的动态破坏研究^[9-11]。最近,研究者提出了另一种解决方法,基于经典 LSM 引入额外维相互作用,称之为四维弹簧模型(Four-lattice spring model,4D-LSM)^[12]。4D-LSM 的基本元件是由弹簧键连接的颗粒,通过弹簧键的变形和破坏来反映固体的宏观力学响应。4D-LSM 这种自底向上的建模方式与离散元类似,其单元(颗粒)数量必须达到一定规模才能得到足够真实

的模拟结果。因此,4D-LSM 有庞大的计算需求,传统的个人电脑已经不能满足,高效的并行计算是唯一解决办法。

目前,并行计算器件主要分为多核 CPU 和 GPU,而主要计算平台是超级计算机。超级计算机拥有多个节点,每个节点一般是可以单独实现 CPU 多核并行和(或)GPU 并行的计算机,节点间通过网络连接实现消息传递,从而将计算资源整合利用,并达到超高的计算性能。全球超级计算机 Top500 中,目前排名第一的超级计算机“Summit”由 4 608 个节点组成,每个节点搭载 2 个“Power9”CPU 和 6 个“NVIDIA Tesla V100”GPU,CPU 核心数量为 202 752,GPU 流处理器数量超过 1.4 亿,其峰值性能为 200 PFLOPS^[13]。近年来,GPU 计算发展十分迅速^[14],但是 GPU 计算并不能取代 CPU 计算,比如 CPU 更擅长处理逻辑控制密集任务,CPU 多核并行仍然是一种便捷、可靠并且广泛使用的高性能计算方式。实现 CPU 多核并行主要依靠应用程序接口,例如 OpenMP(Open Multi-processing)等^[15]。OpenMP 是基于共享内存的应用程序接口,提供了对并行算法的高层抽象描述,非常适合多核 CPU 计算机的并行程序设计。OpenMP 的显著特点是精简、易用,只需要在串行代码中加入简单的 pragma 指令即可实现并行,因此,OpenMP 的使用非常普遍,例如 4D-LSM 和 DLSM 就采用 OpenMP 实现了多核并行^[12,16]。

高性能计算通常以高性能计算机为依托,但超级计算机硬件的高昂费用和固定资产属性常导致高性能计算的使用成本较高。近年来,计算领域中面向服务的“云计算”为解决这个问题提供了可能。云计算是指通过网络按需提供虚拟计算资源和解决方案的有偿服务,相对于传统的计算模式,其主要优点是配置灵活、方便快捷、管理投入少等。例如,由“阿里云”提供的弹性云服务器类型有通用性、计算型、内存型等,CPU 核数从 2 核到 160 核不等,内存从 4 GB 到 1 920 GB 不等^[17],付费方式也有按量计费、

按时间计费等不同选择。笔者主要研究多核 4D-LSM 在云计算及常规多核工作站和个人电脑上的并行运算性能,通过大量数值模拟计算来研究线程数量、硬件配置、求解问题类型等对 4D-LSM 并行计算时间的影响,进而找到 4D-LSM 在“阿里云”计算环境下的极限规模和瓶颈,最后,通过 4D-LSM 求解脆性材料的三维破裂问题来展示离散数值计算方法和并行计算相结合带来的优势。

1 四维弹簧模型

1.1 基本原理

在经典物理学中,空间是三维的,时间作为第四维,它们共同构成了四维时空。有些研究者为了统一自然界的 4 种基本力,通过引入一个额外的空间维度,提出了五维时空。4D-LSM 借鉴了五维时空理论。4D-LSM 模型的构建过程如图 1 所示。图 1(a)中,三维空间中的立方体晶格模型能够再现各向同性弹性,其泊松比固定为 0.25。该原始模型的弹簧键有两种,即正弹簧(例如 AB)和对角弹簧(例如 AC),其刚度系数均为 k 。图 1(b)所示为原始模型在第四维的“平行体”,对于给定的质点 A,其“平行体”即为 A' ，“平行体”模型的构造和弹簧刚度均与“本体”模型相同。然后,利用第四维相互作用(弹簧键)连接“本体”和“平行体”,如图 1(c),具体规则为:“本体”模型的一个弹簧(例如 A-B)产生 4 个相应的四维弹簧(A-A'、B-B'、A-B'和 A'-B)。

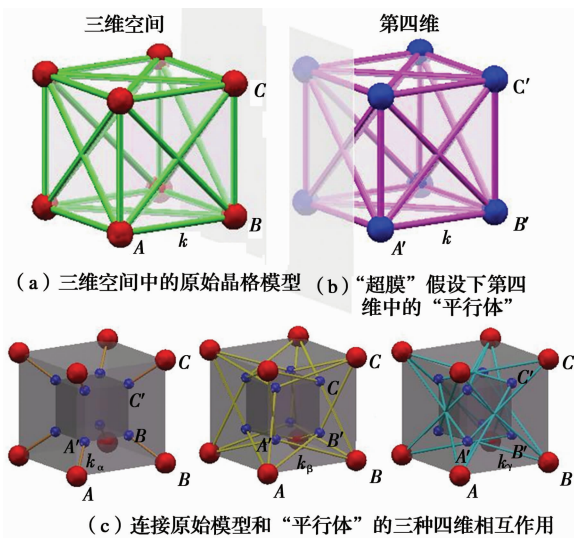


图 1 四维弹簧模型的构建过程^[12]

Fig. 1 The model building process of 4D-LSM

1.2 系统方程

在 4D-LSM 中,假定三维世界是一个四维超膜,离散的四维颗粒由弹簧键连接。4D-LSM 的描述和证明详见文献^[12],这里只关注对实现并行化必不可少

的有关方程。四维颗粒的空间位置和运动参数表示为

$$\mathbf{x}_i = (x_i \ y_i \ z_i \ \vartheta_i)^T \quad (1)$$

$$\dot{\mathbf{x}}_i = (\dot{x}_i \ \dot{y}_i \ \dot{z}_i \ \dot{\vartheta}_i)^T \quad (2)$$

$$\ddot{\mathbf{x}}_i = (\ddot{x}_i \ \ddot{y}_i \ \ddot{z}_i \ \ddot{\vartheta}_i)^T \quad (3)$$

式中: i 代表第 i 个颗粒; x, y, z 和 ϑ 是四维坐标; $\dot{x}, \dot{y}, \dot{z}$ 和 $\dot{\vartheta}$ 是速度; $\ddot{x}, \ddot{y}, \ddot{z}$ 和 $\ddot{\vartheta}$ 是加速度。利用时间中心差分法,则四维颗粒的位置更新公式为

$$\begin{cases} x_i^{t+\Delta t} = x_i^t + \dot{x}_i^{t+\frac{\Delta t}{2}} \Delta t \\ y_i^{t+\Delta t} = y_i^t + \dot{y}_i^{t+\frac{\Delta t}{2}} \Delta t \\ z_i^{t+\Delta t} = z_i^t + \dot{z}_i^{t+\frac{\Delta t}{2}} \Delta t \\ \vartheta_i^{t+\Delta t} = \vartheta_i^t + \dot{\vartheta}_i^{t+\frac{\Delta t}{2}} \Delta t \end{cases} \quad (4)$$

式中: t 为时间; Δt 为时间增量。以相同的方法,可以得到颗粒的速度公式

$$\begin{cases} \dot{x}_i^{t+\frac{\Delta t}{2}} = \dot{x}_i^{t-\frac{\Delta t}{2}} + \ddot{x}_i^t \Delta t \\ \dot{y}_i^{t+\frac{\Delta t}{2}} = \dot{y}_i^{t-\frac{\Delta t}{2}} + \ddot{y}_i^t \Delta t \\ \dot{z}_i^{t+\frac{\Delta t}{2}} = \dot{z}_i^{t-\frac{\Delta t}{2}} + \ddot{z}_i^t \Delta t \\ \dot{\vartheta}_i^{t+\frac{\Delta t}{2}} = \dot{\vartheta}_i^{t-\frac{\Delta t}{2}} + \ddot{\vartheta}_i^t \Delta t \end{cases} \quad (5)$$

颗粒 i 和颗粒 j 之间四维距离为

$$l_{ij} = \sqrt{(x_j^t - x_i^t)^2 + (y_j^t - y_i^t)^2 + (z_j^t - z_i^t)^2 + (\vartheta_j^t - \vartheta_i^t)^2} \quad (6)$$

如果这些颗粒通过刚度为 k 的弹簧连接,那么颗粒 j 对颗粒 i 的作用力为

$$\begin{cases} f_{ij}^x = \frac{x_j^t - x_i^t}{l_{ij}^0} k (l_{ij}^0 - l_{ij}^t) \\ f_{ij}^y = \frac{y_j^t - y_i^t}{l_{ij}^0} k (l_{ij}^0 - l_{ij}^t) \\ f_{ij}^z = \frac{z_j^t - z_i^t}{l_{ij}^0} k (l_{ij}^0 - l_{ij}^t) \\ f_{ij}^\vartheta = \frac{\vartheta_j^t - \vartheta_i^t}{l_{ij}^0} k (l_{ij}^0 - l_{ij}^t) \end{cases} \quad (7)$$

式中: l_{ij}^0 和 l_{ij}^t 是按式(6)计算的原始弹簧长度和当前弹簧长度。在 4D-LSM 中,假设沿第四维的体力为零,那么,若某颗粒 i 有 m 个颗粒与之相连,则其所受的合力为

$$\begin{cases} f_i^x = m_i g_x + \sum_{j=1}^m f_{ij}^x \\ f_i^y = m_i g_y + \sum_{j=1}^m f_{ij}^y \\ f_i^z = m_i g_z + \sum_{j=1}^m f_{ij}^z \\ f_i^\vartheta = \sum_{j=1}^m f_{ij}^\vartheta \end{cases} \quad (8)$$

式中: m_i 为颗粒 i 的质量; g_x 、 g_y 和 g_z 为重力加速度。在 4D-LSM 中, 假设牛顿第二定律也适用于第四维, 则颗粒 i 的加速度为

$$\begin{cases} \ddot{x}_i^t = \frac{f_i^x}{m_i} \\ \ddot{y}_i^t = \frac{f_i^y}{m_i} \\ \ddot{z}_i^t = \frac{f_i^z}{m_i} \\ \ddot{\vartheta}_i^t = \frac{f_i^{\vartheta}}{m_i} \end{cases} \quad (9)$$

式(1)~式(9)是实现 4D-LSM 并行化涉及的所有基础性计算。

1.3 模型参数选取

对于立方体四维晶格, 有 3 种类型的四维弹簧, 刚度分别为 k_α 、 k_β 、 k_γ 。对于弹性各向同性体, 它们的刚度值需满足关系^[12]

$$k_\alpha = k_\beta = \frac{4}{3}k_\gamma = \lambda^{4D}k \quad (10)$$

式中: λ^{4D} 为四维刚度系数; k 为三维弹簧的刚度, k 用式(11)计算。

$$k = \frac{6VE}{\eta \sum l_i^2} \quad (11)$$

式中: V 为三维晶格模型的代表体积; E 为弹性模量, l_i 为三维晶格模型的初始弹簧长度; η 为尺度参数。 η 可用式(12)计算^[12]。

$$\eta = -0.007\ 850\ 6\lambda_{4D}^2 + 0.416\ 136\ 15\lambda_{4D} + 1.003\ 692\ 23 \quad (12)$$

四维刚度系数 λ^{4D} 也可以由泊松比得到^[12]。

$$\lambda_{4D} = -211.134\ 937\ 79v^3 + 162.846\ 558\ 51v^2 - 55.424\ 497\ 19v + 6.929\ 022\ 11 \quad (13)$$

式中: v 是泊松比。结合式(10)和式(13)可算出式(7)所需的力学参数(弹簧刚度)。这些参数都是预先计算的, 与 4D-LSM 的计算循环无关, 因此, 参数计算部分不参与并行。更多细节和数学证明可以在 4D-LSM 的原始文献^[12]中找到。

1.4 OpenMP 多核并行

OpenMP 是多核环境下实现软件并行的有力工具, 是一个包含了编译指令、运行库例程和环境变量的应用程序接口, 能够在 FORTRAN、C 和 C++ 等编译环境下运行, 使用叉接模型实现任务的并行化, 本文中所有代码均在 Windows 系统的 Microsoft Visual Studio 环境下采用 C++ 语言编写。4D-LSM 中颗粒的运动遵从牛顿第二定律, 利用力-位移法则不断更新颗粒的运动状态, 其求解过程类似

DEM, 主要分为两部分, 一是力的计算, 二是运动状态更新, 包括位移、速度等。由于单个颗粒运动状态的计算独立于其他颗粒, 而不同颗粒的相应数据都执行同样的操作指令, 因此, 4D-LSM 的计算属于数据密集型, 非常适合并行。串行 4D-LSM 只有一个主线程, 依次计算颗粒的力和位移, 如图 2(a)所示。多核 4D-LSM 使用了叉接模型, 使多个处理器同时参与计算, 其工作流程如图 2(b)所示: 当 4D-LSM 开始运行时, 主线程被激活; 当程序执行到需要并行的位置时, 主线程通过分叉使得更多的线程参与计算。

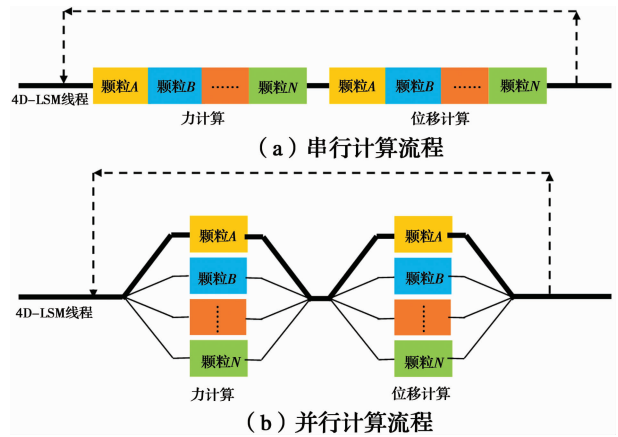


图 2 4D-LSM 的分叉并行策略

Fig. 2 Fork/Join parallel strategy of 4D-LSM

1.5 计算模型

采用两种 4D-LSM 计算模型, 分别对应弹性和破坏问题。如图 3 所示, 模型外观均为立方体, 选用的颗粒直径为 1 mm。第 1 计算模型为立方体单轴压缩试验, 底端在竖直方向被固定, 顶端施加竖直向下的位移荷载, 整个模拟过程不破坏, 属于线弹性问题。第 2 个问题是爆炸开裂模型, 模型中心有球形空洞(绿色部分), 冲击荷载施加在球体的内表面, 属于动态破坏问题。每个计算模型采用不同规模, 立方体边长分别为 20、50、100、150、200、250、300 mm, 因此, 最大的模型颗粒数达到 2 700 万 ($300 \times 300 \times 300$)。

1.6 并行性能分析

云服务器是一种虚拟的计算机, 根据客户的需求可以有不同配置。如果用户选用的云服务器操作系统与自己本地计算机的操作系统一样(如 Windows 系统或者 Linux 系统等), 那么, 云服务器的操作体验与本地计算机几乎没有区别, 能够在本地机运行的程序同样可以在云服务器上运行, 不需要做任何额外的更改, 本文涉及的基于 OpenMP 的

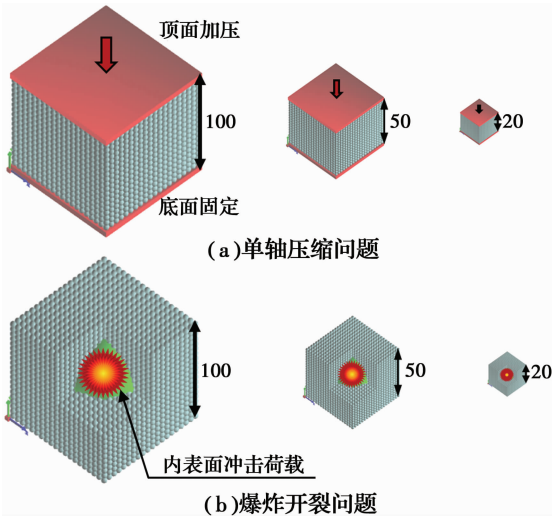


图 3 两种 4D-LSM 计算模型

Fig. 3 Two kinds of 4D-LSM computing models

4D-LSM 也是如此。如表 1 所示,选用的云服务器 (CS)具有 64 核心、128 GB 内存容量,CPU 频率是 2.5 GHz。线程测试选用的是单轴压缩模型,模型的大小有 3 种,边长为 50、100、150 mm,分别记为“Cube_50”、“Cube_100”和“Cube_150”,相应的颗粒数为 12.5 万、100 万和 337.5 万。将每一个模型在不同的计算机上采用不同的线程数进行重复计算,记录每次计算的时长,并换算出加速比,加速比定义为单线程计算时间与多线程计算时间的比值。

表 1 计算机主要参数

Table 1 Main parameters of the computers

计算机	CPU 频率/GHz	核心数	线程数	内存容量/GB	内存带宽/(GB · s ⁻¹)
CS	2.5	64	64	128	20
PC-1	3.6	4	8	8	25.6
PC-2	3.0	4	4	8	38.4
WS-1	2.6	20	40	32	34.1
WS-2	2.2	20	40	64	68.3

阿里云上的测试结果如图 4 所示,最大加速比约为 16.8×。单从最大加速比来看,云计算优于两台工作站 WS-1 和 WS-2(见图 8,最大加速比约为 9.0×),并且,云计算还有使用灵活、无需维护等优点。但是,相较于本地计算资源,云计算也有不足之处。首先是性价比的问题,测试用的云服务器按时间计费,费用约为 350 元/d,而工作站 WS-2 的一次性投入约为 5 万元,该费用只能购买该云服务器 5 个月左右,但通常情况下,一台工作站的性能至少可以在 3 年内保持相当的竞争力。数值计算方面的科

研工作,经常需要修改模型的参数,这样的重复计算是对云计算资源的浪费。因此,最合理的方式是利用本地计算资源调整数值模型,然后利用云服务一次性完成大规模计算。其次,大规模计算必定涉及到大量的数据存储问题,由于云服务的存储具有时效性,也不方便进行后处理工作,因此,如何快速将海量数据保存到本地存储空间是云服务应用于数值分析计算面临的另一个问题。

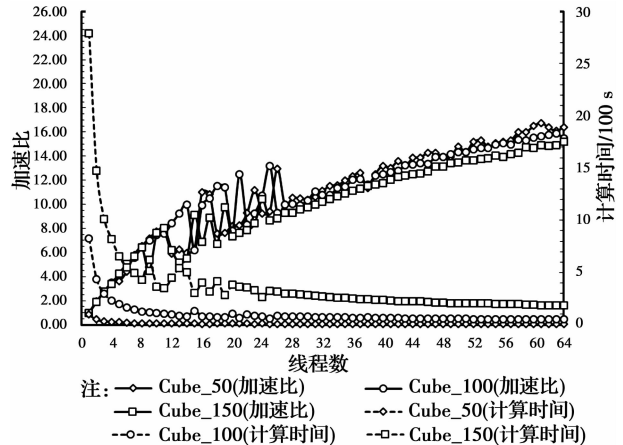


图 4 阿里云计算环境下 4D-LSM 并行效率测试结果

Fig. 4 Parallel efficiency test results of 4D-LSM in Alibaba cloud computing environment

PC-1、PC-2、WS-1 和 WS-2 的测试结果如图 5~图 8 所示,最大加速比分别为 2.6×、3.2×、10.8×和 9.1×,随着线程数的增加,计算速度总体上呈加快的趋势。PC-1 是 4 核 8 线程,由图 5 可知,当线程数超过 4 以后,加速效果明显下降,例如“Cube_150”的模型使用 2 线程、4 线程和 8 线程时的加速比分别为 1.84×、2.54×和 2.6×,意味着加速比从 2 线程到 4 线程的增幅为 38%,而从 4 线程到 8 线程的增幅仅为 15%,这说明物理核心的加速效果远远超出超线程技术的加速效果。两台工作站 WS-1 和 WS-2 是 20 核 40 线程的双路计算机,由图 7、图 8 可知,当线程数超过 20 以后,加速效果的提升即开始放缓。更值得关注的是,对于 WS-1 和 WS-2 这两台双路计算机(每个 CPU 有 10 个核心、两个 CPU 共 20 核),10 线程和 20 线程的加速效果几乎相同,例如 WS-1 上“Cube_150”模型使用 10 线程和 20 线程时的加速比分别为 7.1×和 7.7×,当线程数介于 10 和 20 之间时,加速比呈现先降后升的“凹”型曲线,计算资源的增加却适得其反。

1.7 并行性能影响因素及极限运算分析

1.7.1 求解类型的影响 4D-LSM 模型中,破坏的

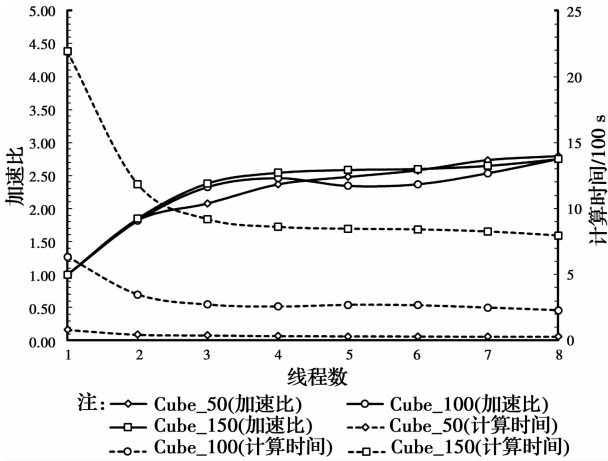


图 5 4D-LSM 在 PC-1 上并行效率测试结果
Fig. 5 Parallel efficiency test results of 4D-LSM on PC-1

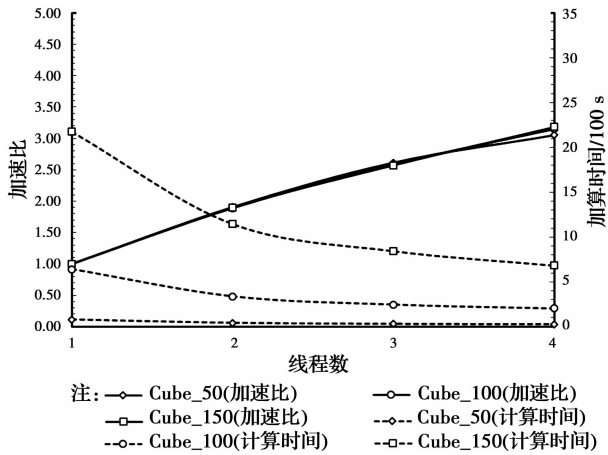


图 6 4D-LSM 在 PC-2 上并行效率测试结果
Fig. 6 Parallel efficiency test results of 4D-LSM on PC-2

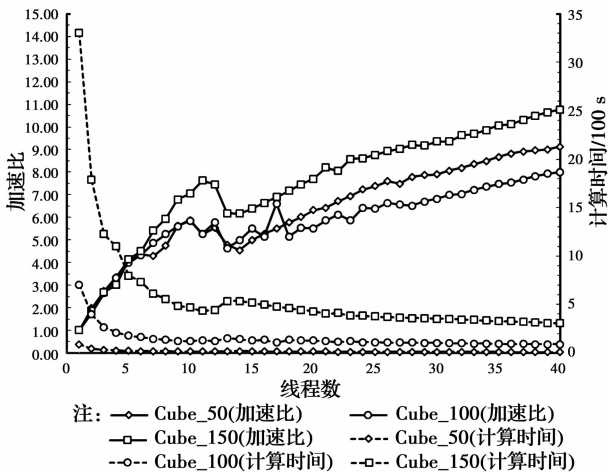


图 7 4D-LSM 在 WS-1 上并行效率测试结果
Fig. 7 Parallel efficiency test results of 4D-LSM on WS-1

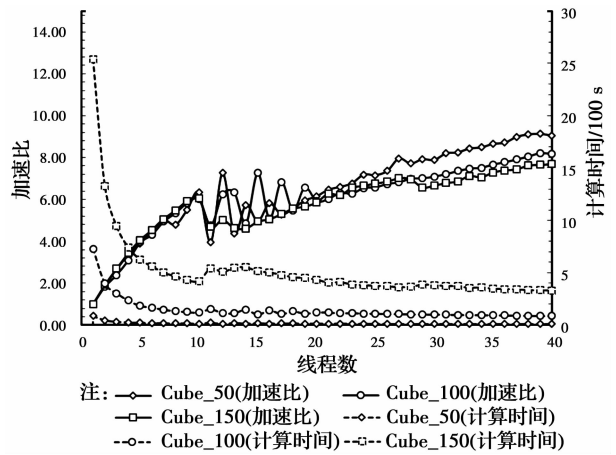


图 8 4D-LSM 在 WS-2 上并行效率测试结果
Fig. 8 Parallel efficiency test results of 4D-LSM on WS-2

受拉,其目的只是为了防止破坏颗粒由于运动而穿透其他颗粒,相对于非破坏模型,破坏模型的颗粒检索将会消耗额外的时间。为了研究加速比与求解类型的关系,选用单轴压缩模型和爆炸开裂模型进行对比,前者代表弹性(Elastic)问题,后者代表破坏(Fracture)问题,模型外观均为立方体且边长均为 100 mm,计算机选用工作站 WS-2,测试结果如图 9。对于线程数与计算效率的总体关系,破坏模型与前述弹性模型一致,但是,对比弹性模型与破坏模型的加速比发现,随着线程数的增加,两者的加速比差距越来越大,最终,使用 40 线程时弹性模型的加速比达到 9.0 \times ,而破坏模型相应的加速比为 5.8 \times ,仅为前者的 64%。总之,并行化的 4D-LSM 求解破坏问题所获得的加速效果要低于非破坏问题,使用的线程数越多,这种差距越明显。

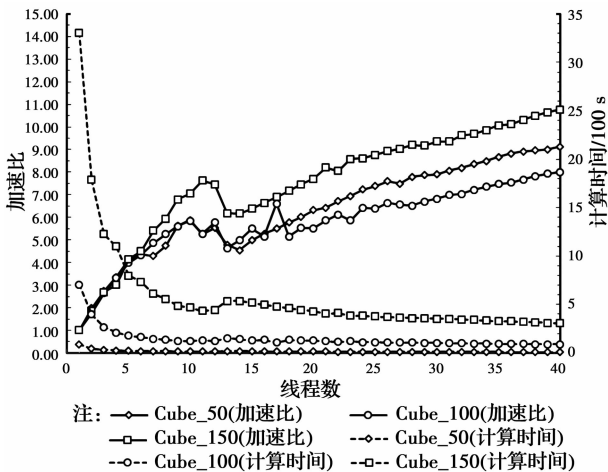


图 9 工作站 WS-2 对于弹性模型和破坏模型的并行效率测试结果
Fig. 9 Parallel efficiency test results for elastic model and failure model of 4D-LSM on WS-2

颗粒在每一步的计算过程中都会进行动态接触检索,当有其他颗粒接触到该破坏颗粒时,这两个颗粒之间会产生一个新的特殊弹簧键,该弹簧键并不能

1.7.2 计算规模的影响 由图 7、图 8 可知,对于大小不同的模型,在相同条件下,其加速比有一定的

区别。例如工作站 WS-1 使用 40 线程时,“Cube_50”“Cube_100”和“Cube_150”的加速比分别为 $9.1\times$ 、 $8.0\times$ 和 $10.8\times$,而相同情况下,工作站 WS-2 上对应的加速比分别为 $9.0\times$ 、 $8.2\times$ 和 $7.7\times$ 。若仅就这 3 种大小的模型而言,则工作站 WS-1 上“Cube_150”加速效果最好($10.8\times$),而工作站 WS-2 上“Cube_50”才是加速效果最好的($9.0\times$)。因此,模型的大小对加速效果有一定的影响,但这种影响没有普遍的规律,随计算机硬件配置的不同而不同。

同时也发现,弹性模型的计算时间与模型的规模呈正比,而破坏模型则并非如此。对于破坏模型而言,颗粒检索会消耗额外的时间,破坏的颗粒越多,每一步的计算时间就越长,但每一个破坏颗粒额外消耗多长时间还不明确,整个破坏过程目前也无法预测。由于这些“复杂性”,测试结果中破坏模型的计算时间与规模大小的关系曲线并不具有普适性,只能说明一般情况下是非线性的,从而区别于弹性模型的线性关系。

1.7.3 计算硬件的影响 CPU 的主要参数包括频率、核心数量和线程数量等,更高的 CPU 频率、更多

的核心或者线程都能够获得更快的计算速度。因此,由表 1 可知,一般情况下,拥有 8 线程 3.6 GHz CPU 的 PC-1 要比拥有 4 线程 3.0 GHz CPU 的 PC-2 更快,WS-1 也会因为更高 CPU 的频率而获得比 WS-2 更好的性能。将图 4 中关于计算时间的数据做进一步处理后得到表 2,表中 ΔPC 、 ΔWS 分别为 PC-1 与 PC-2、WS-1 与 WS-2 计算同一模型所用时间之差。从表 2 来看,虽然有几处 Δ 值为负数,但都是在模型较小、整个计算时间较短的情况下发生,不具有代表性,而绝大部分 Δ 值都为正数。因此,从统计的角度,对于同一个模型,可以认为 PC-1 比 PC-2 耗时更多,WS-1 比 WS-2 耗时更多,也就是说,PC-2 和 WS-2 计算速度更快,与之前的预测刚好相反,这说明 4D-LSM 的计算速度并非完全由 CPU 的性能决定。在表 1 中,对比 4 台计算机的硬件,PC-2 和 WS-2 唯一的优势就是拥有更高的内存带宽。由于计算时间不仅包括 CPU 处理数据的时间,也包括其他必要的时间消耗,如 CPU 和内存交换数据的时间,高内存带宽意味着数据传输更快,最终的结果是 PC-2 和 WS-2 在计算时速度更快。因此,对于 4D-LSM,若 CPU 性能差距不是很悬殊,则内存带宽成为计算速度非常重要的影响因素。

表 2 使用最大线程数计算不同大小模型的时间消耗表

Table 2 Calculating time of different size models using maximum thread number

求解类型	模型边长/mm	计算时间/s					
		PC-1	PC-2	ΔPC	WS-1	WS-2	ΔWS
单轴压缩	20	1.326	1.277	0.049	0.721	1.125	(0.404)
	50	26.894	23.214	3.68	12.356	13.235	(0.879)
	100	218.571	178.606	39.965	129.664	113.18	16.484
	150	753.637	642.34	111.297	378.563	363.981	14.582
	200				938.604	825.472	113.132
	250				1 943.45	1 630.89	312.56
爆炸开裂	20	1.372	1.376	(0.004)	0.795	1.187	(0.392)
	50	31.075	26.469	4.606	17.464	16.735	0.729
	100	275.371	221.807	53.564	178.568	160.856	17.712
	150	1 106.14	996.65	109.49	894.115	822.454	71.661
	200				3 069.74	2 954.74	115
	250				9 598.24	8 950.96	647.28

1.7.4 并行计算量瓶颈分析 将两组模型(单轴压缩、爆炸开裂)按从小到大的顺序依次运行,记录其计算时间以及消耗的物理内存,测试时,每台计算机都使用最大线程,例如,PC-1 使用 8 线程,而两台工作站 WS-1、WS-2 均使用 40 线程。从小到大的立方体模型的边长为 20、50、100、150、200、250、300、

350、400 mm 等,在此序列下,PC-1、PC-2、WS-1、WS-2 能计算的最大模型边长分别是 150、150、250、300 mm,对应的颗粒数分别为 337.5 万、337.5 万、1 562.5 万和 2 700 万。由此可见,不论是弹性模型还是破坏模型,模型大小(颗粒数)与消耗的物理内存呈同一个线性关系。事实上,经过更进一步的数

据分析发现,内存消耗与使用哪台计算机也没有联系,即模型的颗粒数量与内存消耗存在一一对应的关系(如图 10 所示),每 100 万颗粒约需要 1.8 GB 内存,目前来看,4D-LSM 的计算量由计算机的内存容量决定。例如,WS-1 的内存容量是 32 GB,当计算边长为 250 mm 的模型时,4D-LSM 消耗的内存约为 28 GB,而下一个模型边长是 300 mm,颗粒数量 2 700 万,按前述标准约需要 48 GB 的内存,因此,在 WS-1 上无法计算,最终,该模型在拥有 64 GB 内存的工作站 WS-2 上运行,而 2 700 万颗粒也几乎是 WS-2 的极限计算量。然而,对于阿里云来讲,则可以最大运行 10 亿单元的计算模型。从这点上来讲,云计算为一些对颗粒规模要求十分庞大的问题提供了除传统超级计算集群之外的可行解决途径。相比传统超级计算集群,云计算无需对代码进行修改,也无需进行昂贵的硬件投资。然而,4D-LSM 是采用自建的图形交互界面建模,能够建立多大的模型受制于显存。例如,WS-2 配备的“NVIDIA Quadro M5000”具有 8 GB 的显存,其构建的最大模型是 $450 \times 450 \times 450$ (约 9 000 万颗粒)。如果假设建模所需显存与颗粒数成正比,构建 10 亿颗粒的模型则至少需要大约 88 GB 的显存,因此,目前 4D-LSM 大规模并行的瓶颈在于前处理。

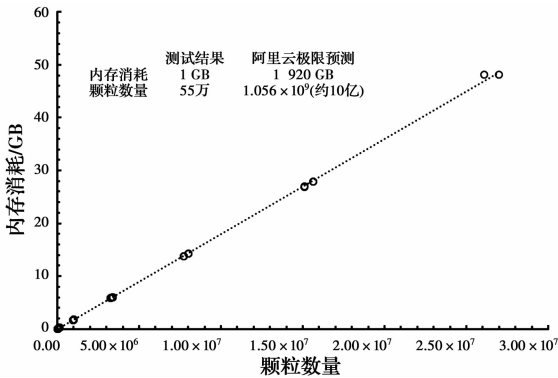


图 10 模型大小与内存消耗的关系

Fig. 10 The relationship between model size and memory consumption

2 应用案例

采用 4D-LSM 进行三维裂纹扩展分析。币型裂纹试样的尺寸及荷载条件如图 11(a) 所示,裂纹形状为圆形,直径 18 mm,厚度 1 mm,中心位置与整个立方体试样中心位置重合,裂纹面与试样底面夹角 $\theta = 30^\circ$ 。建立两个 4D-LSM 模型,一个解析度为 $110 \times 110 \times 110$ (约 130 万颗粒),另一个解析度为 $220 \times 220 \times 220$ (颗粒数大约为 1 060 万),除此之

外,两个模型并无其他任何差别。图 11 给出了针对三维币形裂纹的计算模型,采用并行 4D-LSM 进行了求解。图 12 展示了三维币形裂纹模型在不同解析度下的裂纹发展过程。通过对比,低解析度模型虽然能大致展现裂纹的扩展过程,但裂纹形态相对比较粗糙,裂纹扩展的对称性远不如高解析度模型。由此可见,更高解析度的 4D-LSM 模型对精准模拟三维裂纹扩展问题非常关键。基于云计算的并行计算技术可以求解更高解析度的计算模型,非常适合于求解三维裂纹扩展的计算。

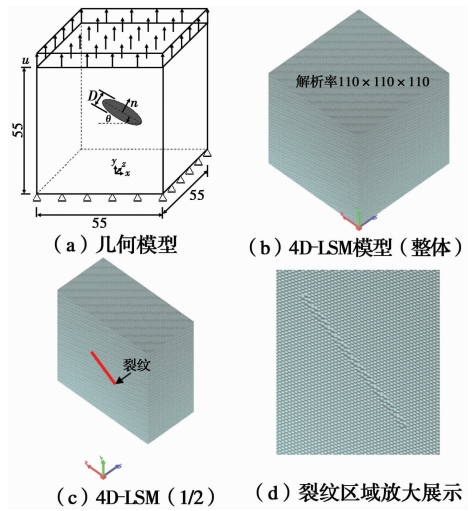


图 11 三维币型裂纹扩展模型

Fig. 11 The model of three-dimensional coin-shaped crack propagation

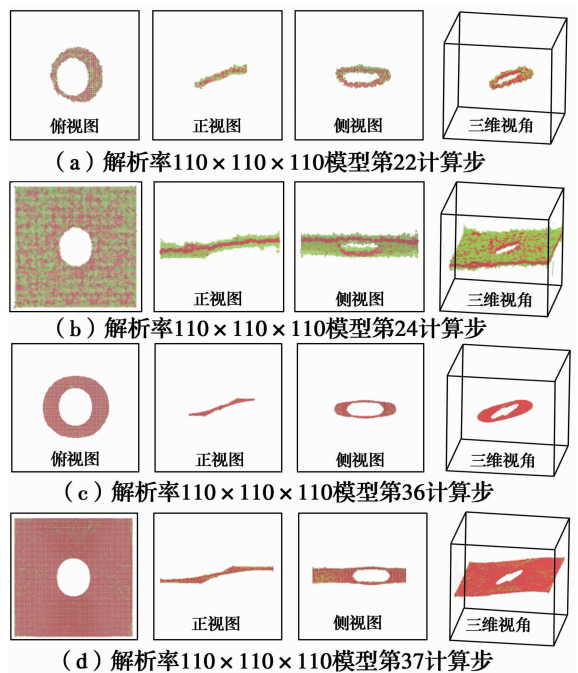


图 12 裂纹扩展过程

Fig. 12 The crack propagation process

3 结论

主要研究了4D-LSM在云计算环境下的并行性能,考虑了线程数量、硬件资源、模型大小、求解类型等因素。得到如下主要结论:

1)4D-LSM具有较好的并行性能,在20核的双路计算机上的最大加速比接近 $11.0\times$,而在64核的云服务器上的加速比接近 $17\times$ 。

2)4D-LSM模型的规模对加速效果有一定的影响,并因使用的计算机不同而不同。

3)4D-LSM求解弹性问题的加速效果优于求解破坏问题,使用的线程数量越多,这种差别表现得越明显。

4)非破坏模型的计算时间与颗粒数呈正比关系,而破坏模型由于其“复杂性”,通常情况下不是正比关系。

5)4D-LSM模型的颗粒数量与内存消耗呈正比,计算机的极限计算量由内存容量决定,每100万颗粒大约需要1.8GB的内存,若要求解10亿颗粒的模型,理论上至少需要1.8TB的内存。

6)对于双路计算机应当注意,当线程数量介于单颗CPU的物理核心数和双CPU的总物理核心数时,计算效率会下降,并且造成计算资源的浪费。

7)虽然云计算非常灵活且能提供强大的高性能计算能力,但其性价比也值得商榷,使用时应当综合考虑,有的放矢。

另外,需要说明:

1)在测试极限计算量时只考虑了物理内存,实际上有些4D-LSM模型在内存需求超过计算机的物理内存时也可以计算,比如边长为200mm的单轴压缩模型,颗粒数量是800万,大约需要14.4GB的内存,却可以在内存容量8GB的PC-2上运行,这是因为系统自动启用了虚拟内存(此处虚拟内存是相对物理内存而言,并非编程模式下所指的虚拟地址空间),但此时计算速度非常缓慢,不在可接受的范围,因此,未予以考虑。

2)对于双路计算机,当使用的线程数量介于单颗CPU核心数和双CPU总核心数时,不仅计算效率会下降,而且多次重复计算的结果表明:在此区间计算时间的离散程度也急剧增加,即计算效率不稳定,计算效率不稳定的情况与求解类型无关。相差40%的结论是因为统计了WS-2在10线程和20线程之间重复计算100次同一个模型的计算时间,求

出了计算时间的变异系数,该变异系数最高为20%左右。因此,当线程数量处于该区间时,同样模型的两轮计算时间有可能相差40%。

3)对于本地计算机,当使用的线程数量超出计算机的最大线程数时,其计算效率会下降20%左右,但对于云计算的虚拟服务器而言,超出最大线程后,计算效率并不会下降,而是保持在同一水平线上。

参考文献:

- [1] 唐春安,张永彬. 岩体间隔破裂机制及演化规律初探[J]. 岩石力学与工程学报, 2008, 27(7): 1362-1369. TANG C N, ZHANG Y B. Discussion on mechanism and evolution laws of fracture spacing in rock mass [J]. Chinese Journal of Rock Mechanics and Engineering, 2008, 27(7): 1362-1369. (in Chinese)
- [2] 吴顺川,周喻,高斌. 卸载岩爆试验及PFC^{3D}数值模拟研究[J]. 岩石力学与工程学报, 2010, 29(Sup2): 4082-4088. WU S C, ZHOU Y, GAO B. Study of unloading tests of rock burst and PFC^{3D} numerical simulation [J]. Chinese Journal of Rock Mechanics and Engineering, 2010, 29(Sup2): 4082-4088. (in Chinese)
- [3] 李利平,李术才,李树忱,等. 松散承压含水层下采煤的流固耦合模型试验与数值分析研究[J]. 岩土工程学报, 2013, 35(4): 679-690. LI L P, LI S C, LI S C, et al. Numerical analysis and fluid-solid coupling model tests of coal mining under loose confined aquifer [J]. Chinese Journal of Geotechnical Engineering, 2013, 35(4): 679-690. (in Chinese)
- [4] 鞠杨,夏昌敬,谢和平,等. 爆炸荷载作用下煤岩巷道底板破坏的数值分析[J]. 岩石力学与工程学报, 2004, 23(21): 3664-3668. JU Y, XIA C J, XIE H P, et al. Numerical analysis on rupture of subfloor of coal Mine tunnel subjected to blasting loads [J]. Chinese Journal of Rock Mechanics and Engineering, 2004, 23(21): 3664-3668. (in Chinese)
- [5] 李世海,高波,燕琳. 三峡永久船闸高边坡开挖三维离散元数值模拟[J]. 岩土力学, 2002, 23(3): 272-277. LI S H, GAO B, YAN L. 3-D simulation of the excavation of high steep slope of Three-Gorges permanent lock by distinct element method [J]. Rock and Soil Mechanics, 2002, 23(3): 272-277. (in Chinese)

- [6] CUNDALL P A, STRACK O D L. A discrete numerical model for granular assemblies [J]. *Géotechnique*, 1979, 29(1): 47-65.
- [7] SHI G H. Discontinuous deformation analysis: A new numerical model for the statics and dynamics of block systems [D]. Berkeley: University of California, 1988.
- [8] ZHAO G F, FANG J N, ZHAO J. A 3D distinct lattice spring model for elasticity and dynamic failure [J]. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2011, 35(8): 859-885.
- [9] ZHAO G F, XIA K W. A study of mode-I self-similar dynamic crack propagation using a lattice spring model [J]. *Computers and Geotechnics*, 2018, 96: 215-225.
- [10] WANG Y B, YANG R S. Study of the dynamic fracture characteristics of coal with a bedding structure based on the NSCB impact test [J]. *Engineering Fracture Mechanics*, 2017, 184: 319-338.
- [11] ZHAO Y X, ZHAO G F, JIANG Y D, et al. Effects of bedding on the dynamic indirect tensile strength of coal: Laboratory experiments and numerical simulation [J]. *International Journal of Coal Geology*, 2014, 132: 81-93.
- [12] ZHAO G F. Developing a four-dimensional lattice spring model for mechanical responses of solids [J]. *Computer Methods in Applied Mechanics and Engineering*, 2017, 315: 881-895.
- [13] Top500 List-November 2018[EB/OL]. [2018-07-29]. <https://www.top500.org/>
- [14] ZHAO G F, KHALILI N. Graphics processing unit based parallelization of the distinct lattice spring model [J]. *Computers and Geotechnics*, 2012, 42: 109-117.
- [15] Home-Open MP [EB/OL]. [2018-07-29]. <https://www.openmp.org/>.
- [16] ZHAO G F, FANG J N, SUN L, et al. Parallelization of the distinct lattice spring model [J]. *International Journal for Numerical and Analytical Methods in Geomechanics*, 2013, 37(1): 51-74.
- [17] 阿里云 [EB/OL]. [2018-07-29]. <https://www.aliyun.com/>.

(编辑 胡英奎)