Janus2: an FPGA-based Supercomputer for Spin Glass Simulations

——国防科大2020年高性能评测与优化课程小组讨论

成员：崔剑锋 王鹏程

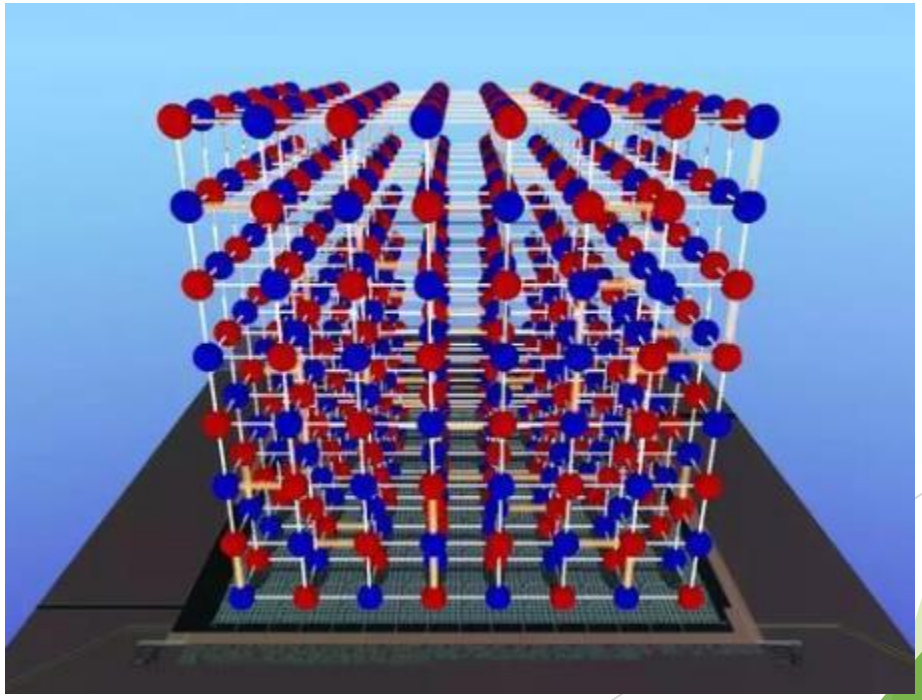
指导：龚春叶、甘新标、杨博

Janus2: an FPGA-based Supercomputer for Spin Glass Simulations

一、背景介绍
二、需求分析
三、技术方案
四、Janus的性能
五、展望与性能提升

背景介绍

自旋玻璃态是一种典型的玻璃态系统，自旋玻璃态中的晶格（如右图所示的红蓝结点）处于两种随机的自旋态，相邻节点之间以键连接，随着某些晶格的自旋状态翻转，整个晶体能量发生变化，表现出不同的性质。



背景介绍

- 自旋玻璃态的蒙特卡洛模拟法，是对每个晶格所处自旋态进行概率估计并更新，经过大量的模拟次数后，得到统计意义上的自旋平衡状态。
- 真实材料实验，总共约有10E20次自旋翻转
  - ✓每个晶格约有10E12次自旋翻转；
  - ✓为了正确反映材料特性，需要至少10E6量级的晶格数量；
  - ✓为了消除误差，需要约10E2量级的样本数量同时实验。
- 商用CPU集群模拟自旋翻转实验
  - ✓ 模拟每次自旋翻转需要1ns量级，因此一个100量级CPU的集群，完整模拟整个实验需要30年时间。

需求分析

- 3维Edwards-Anderson模型

  ✓ 系统总能量为：

$$E = -\sum_{\langle ij \rangle} \sigma_i J_{ij} \sigma_j; \qquad (1)$$

其中，σ_i是晶格i的自旋状态，取值为±1，j是i的相邻晶格（在立方体模型中，有6个），J_{ij}是i和j之间键的耦合状态，

在平衡系统中，有0.5的概率为正耦合（取值为+1），0.5的概率为负耦合（取值为-1）。

  ✓ 单个结点k的局部能量为：

$$\epsilon(\sigma_k) = -\sigma_k \phi_k; \qquad (2)$$

$$\phi_k = \sum_{j=k\pm x, k\pm y, k\pm z} J_{kj}\sigma_j, \qquad (3)$$

其中，σ_k取值为±1，ϕ_k表示与k相邻的结点和k之间的能量，共有6个相邻节点，x、y、z方向各两个，也就是公式

中的k±x，k±y，k±z。

需求分析

- 3维Edwards-Anderson模型
  - ✓给定温度T下结点自旋态的概率分布（Boltzmann-Gibbs分布）：

$$P(\sigma_k = \pm 1) = \frac{\exp\left[\pm\beta\phi_k\right]}{\exp\left[\beta\phi_k\right] + \exp\left[-\beta\phi_k\right]}, \qquad (4)$$

其中，B取值为1/T。在模拟时，一个结点的自旋态取值，通过比较P与[0，1)上均匀分布的某个伪随机数来得到。

需求分析

- 3维Edwards-Anderson模型的蒙特卡洛模拟步骤：

1. 提取完整的J_{ij}配置（等概率取值为+1或-1）；
2. 提取完整的σ_i配置（等概率取值为+1或-1，由公式4可知，此时温度T极高）；
3. 等概率随机选择一个结点k，开始自旋翻转；
4. 由公式3计算结点k的局部能量，由公式4计算结点k的自旋态P（+1）的概率p；
5. 选择一个[0,1)上均匀分布的伪随机数x；
6. 将x与4中的概率p比较，若p较大，则更新σ_k=+1,否则更新σ_k为-1；
7. 重复3-6步若干次。

需求分析

- 并行优化

  - ✓ 可以看到，上面的模拟过程中，有大量的整数乘法和加法运算，例如计算局部能量的公式2和公式3，对此进行变换：

$$\sigma_k \rightarrow S_k = (1+\sigma_k)/2\,, \tag{5}$$

$$J_{ij} \rightarrow \hat{J}_{ij} = (1+J_{ij})/2\,, \tag{6}$$

$$\phi_k \rightarrow F_k = \sum_j \hat{J}_{kj} \oplus S_j = (6-\phi_k)/2\,, \tag{7}$$

  我们把晶格的自旋状态存储在一个长整型字（64位）中，经过变换之后，就可以把整数乘法变为位级的异或运算，使得计算效率和存储效率都大大提高。

  - ✓ 我们以国际象棋棋盘的形式来划分同一平面上的晶格，可以把整个样品分为黑格和白格两类，并且黑格只与白格相邻，反之亦然。这样，我们就可以同时更新所有黑格或所有白格。

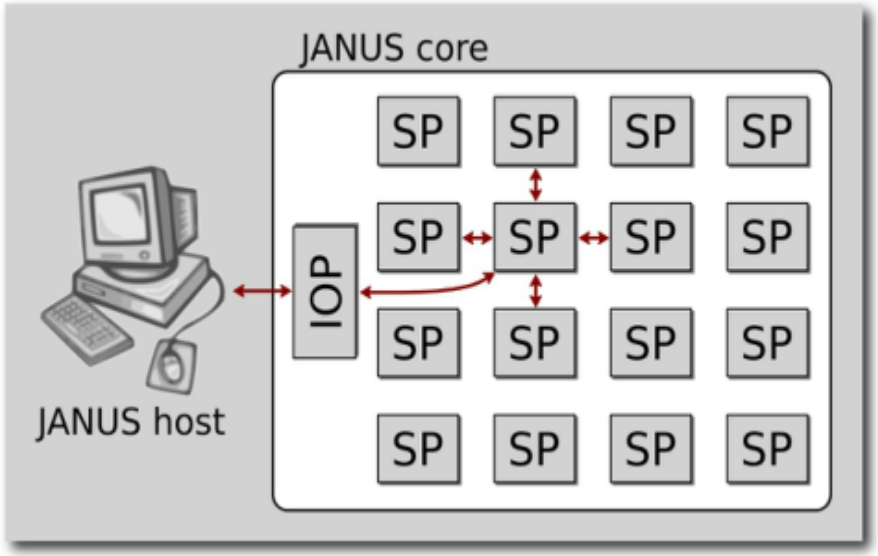
需求分析

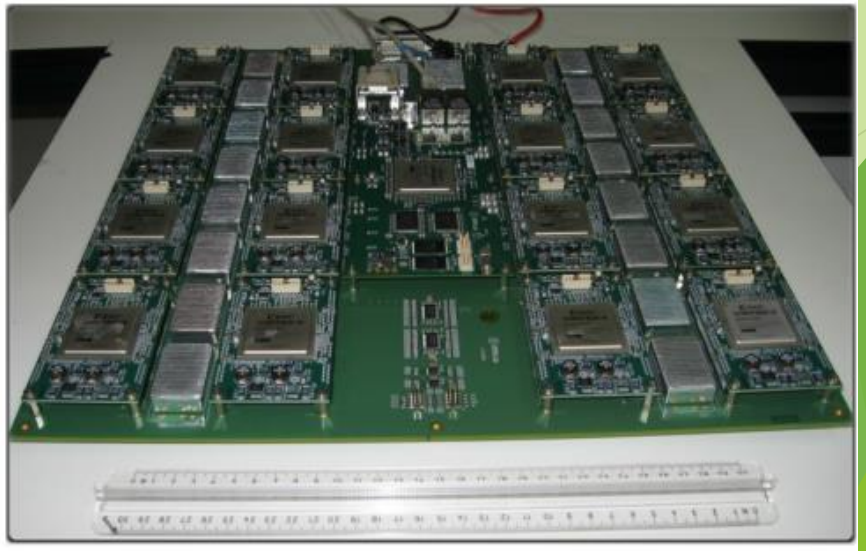
- 基于上面的方案，作者认为基于FPGA的设备非常符合需求，原因有以下几点：

1. 方案中的大部分计算，都只包含一小组离散变量的少量逻辑运算；
2. 存在大量的内部并行性，例如棋盘格方案；
3. 数据存储结构和计算过程的循环都非常规整，且数据存储不依赖于计算，通过简单的状态机就可以控制数据流；
4. 主要的计算过程可以用同一组逻辑运算来表示（如转换后的异或运算）；
5. 伪随机数生成是消耗CPU资源的一个重要部分，但其与整个算法的其他部分无关，因此可以使用一组专门的核心以需要的频率来生成伪随机数。
6. 而传统的商用CPU针对的是更复杂的基本运算（例如整型和浮点运算），并且在生成高质量的随机数方面，有比较明显的瓶颈。所以FPGA系统相比传统CPU集群，效率要高得多。

技术方案

- Janus系统由若干块Janus主板构成，每块主板上有16个模拟处理器（simulating processor）和1个输入输出处理器（input-output processor），每块主板都需要有一台主机PC来控制，如下图所示，就是一块Janus主板的示意图和实例。
- 一个完整的Janus系统由16块主板构成，共有256个模拟处理器，通过8台Linux主机控制，每台主机负责两块主板。每块主板上的16个SP通过2D（4×4）环状网络连接，IOP通过高速以太网连接到主机，负责管理Janus系统，作者选择的是62.5MHz的Xilinx Virtex-4 LX200 FPGA来作为SP和IOP。

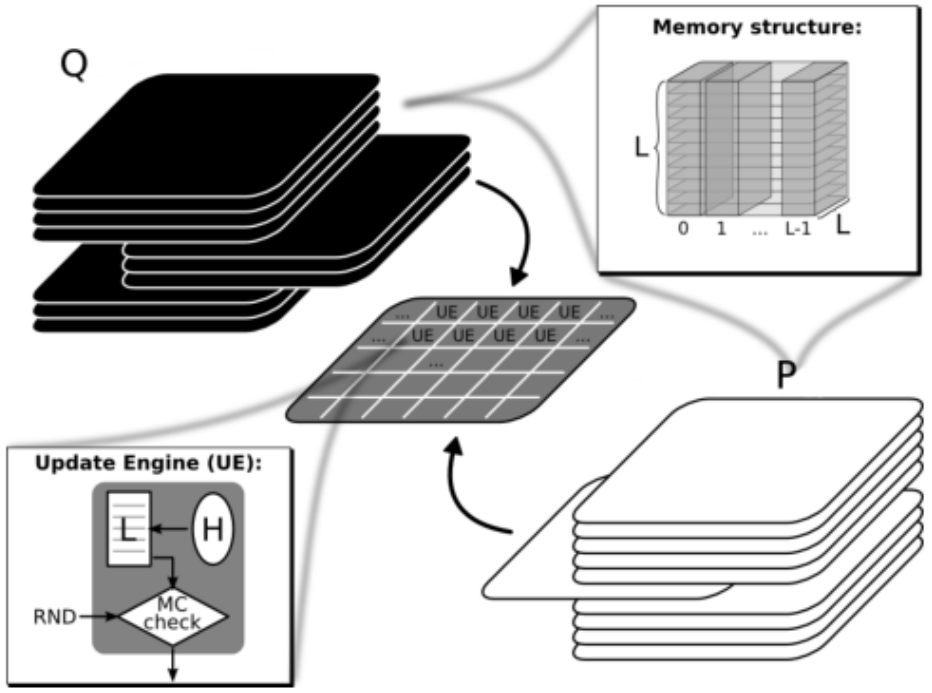




技术方案

- 棋盘格方案在Janus系统中的具体实现

  由于每个样品需要对两份相同的副本进行模拟，因此将两个样品都按照3D棋盘格划分，并将样品1的黑块和样品2的白块组合，样品2的白块和样品1的黑块组合，重新形成样品P和样品Q。



技术方案

- 棋盘格方案在Janus系统中的具体实现
  - ✓ 将晶格自旋态存储在Xilinx FPGA的大量内存块模拟而成的3维存储结构里（如上图所示）。这样，无论是P还是Q，所有晶格的相邻晶格都在另一个样品里。
  - ✓ 执行蒙特卡洛模拟时，假设先更新P，那么按照某个顺序，比如从下往上的顺序，先取出P中的第一个白色层，和Q中的3个黑色层（也就是相邻的黑块），按照之前所说的算法更新该白色层的状态，当取下一个白色层时，上一次所取的3个黑色层有2个依然是相邻层，只需要再取一个黑色层即可。每个内存块只需要一次内存读取和最多一次内存写入，即可更新完整的一层。
  - ✓ 同样的，我们用类似的结构来存储所有必需的耦合常数（即J）
  - ✓ 以此类推，更新完P后，按照同样的方式更新Q，待全部更新完成后，即可完成一次蒙特卡洛模拟。

Janus的性能

使用上文中提到的算法，在62.5 MHz时钟周期运行的应用程序，每个时钟周期执行800个更新的保守数目。将单个旋转更新时间（SUT，是对单个旋转变量执行第2节中描述的过程的步骤3、4、5和6所需的时间）作为性能单位。在每个时钟周期运行1024个更新的SP上，SUT为16 ps。在2007年，我们应用以400 ps SUT运行，并在256个独立样本上实现了完全AMSC实现。我们最好的SMSC实施以1 ns SUT运行。

Janus的性能

商用CPU在显式并行性方面已大大改善，Janus的性能差距正在缩小，后者的效率仍比任何商用解决方案高出10倍。

在多种多核系统（Cell宽带引擎，4核Nehalem Intel CPU Xeon 5560，Tesla C1060 GP-GPU）上执行了几项性能测试，并在8核Intel Sandy Bridge上进行了部分测试。处理器（Xeon E52680）。

Janus的性能

下图比较示例应用程序与某些商用处理器上的Janus性能

single-system SUT (ns/spin)						
L	Janus SP	I-NH (8 Cores)	CBE (8-SPE)	CBE (16-SPE)	Tesla C1060	I-SB (16 cores)
16	0.063	0.98	0.83	1.17	–	–
32	0.016	0.26	0.40	0.26	1.24	0.37
48	0.021	0.34	0.48	0.25	1.10	0.23
64	0.016	0.20	0.29	0.15	0.72	0.12
80	0.020	0.34	0.82	1.03	0.88	0.17
96	0.027	0.20	0.42	0.41	0.86	0.09
128	–	0.20	0.24	0.12	0.64	0.09

global SUT (ns/spin)						
L	Janus	I-NH (8 Cores)	CBE (8-SPE)	CBE (16-SPE)	Tesla C1060	I-SB (16 cores)
16	0.004 (16)	0.031 (32)	0.052 (16)	0.073 (16)	–	–
32	0.001 (16)	0.032 (8)	0.050 (8)	0.032 (8)	0.31 (4)	0.048 (8)
48	0.0013 (16)	0.021 (16)	0.030 (8)	0.016 (16)	0.27 (4)	0.015(16)
64	0.001 (16)	0.025 (8)	0.072 (4)	0.037 (4)	0.18 (4)	0.015 (8)
80	0.0013 (16)	0.021 (16)	0.051 (16)	0.064 (16)	0.22 (4)	0.011 (16)
96	0.0017 (16)	0.025 (8)	0.052 (8)	0.051 (8)	0.21 (4)	0.012 (8)
128	–	0.025 (8)	0.120 (2)	0.060 (2)	0.16 (4)	0.011 (8)

Janus的性能

功耗方面：

对L=80的三维的Edwards-Anderson模型进行蒙特卡洛步骤模拟，包含256个样本，每个样本重复两次，并将样本植入整个机器的256个SP中。

使用16个处理器场（256个样本）不间断运行17000000秒（201天；请注意，五年前在双核CPU上，挂钟时间要长10倍）才能完成上述程序。总共消耗26 GJ，每个CPU贡献95W。

The Janus Collaboration, Phys. Rev. Lett. 101,(2008) 157201

JANUS2的改进与展望

下一代Janus超级计算机将尊重其前身的通用体系结构，近几年来技术的进步使许多改进成为可能。

1. 最新的FPGA器件；
2. IOP与主机PC之间的耦合更紧密以及板内部；
3. 跨板的IOP和SP之间更快，更灵活的通信；

JANUS2的改进与展望

Janus2将再次使用集群的处理板(FPGA)。

每个板卡都是一组16个SP和一个IOP，它们作为处理模块（PB）上的背负式模块构建。

PB提供所有电气链路和连接器，以将SP网格布置成4×4×1 3D环形网络：相对于Janus板增加的第三维尺寸允许相邻板上的SP之间进行通信；

一台带有B板的Janus2机器将是4×4×B 3D环形网络。 PB还为所有设备提供125 MHz主时钟。

JANUS2的改进与展望

IOP的复杂性极大提高：

它将集成所有以前的IOP功能，并将集成主机PC。

模块化计算机（COM）快捷板插入IOP背负模块。

IOP通过Infiniband适配器与世界连接；还将有服务连接：两个千兆以太网通道和两个串行接口（COM和FPGA的每种类型之一）。 IOP当然也为SP的FPGA的即时配置提供了编程接口。每个SP都可以彼此独立地配置。

JANUS2的改进与展望

新的FPGA功能与高速互联：

可用逻辑允许每个时钟周期的更新次数增加2倍。来自系统的更快时钟（在Janus中为62.5 MHz）。这相当于将全局SUT增加4倍。另一个因素的改进可能来自于深度地定制我们的应用以适应新的可用快速FPGA。

与Janus的主要区别在于3D网格中SP之间的高速直接连接：每个链接比2D中的每个链接快大约五倍Janus板的SP网格。

JANUS2的改进与展望

功耗上：

在处理相同的样本数据时,通过软件开发工具来预测SP的能耗，而无需对正在运行的硬件进行直接测量。

我们假设每个SP的乐观数字为50 W，完成L = 150系统的仿真程序所需的总功率为30 GJ（L = 80晶格将为5 GJ）。16个商用CPU（Sandy-Bridge）的服务器场将消耗92 GJ。

请批评指正！