

曙光 TC4600 百万亿次超级计算系统新系统测试

Gaussian16

中国科学技术大学[超级计算中心](#) 张运动

一、软件介绍：

Gaussian 软件是目前计算化学领域内最流行、应用范围最广的综合性量子化学计算程序包。Gaussian 软件基于量子力学而开发，它致力于把量子力学理论应用于实际问题，它可以通过一些基本命令验证和预测目标体系几乎所有的性质。此外，可视化软件 GaussView 的发布及计算机的快速发展更是大大降低了理论计算的门槛，使得各领域研究者能够轻松使用 Gaussian 研究和分析各种科学问题。目前为止最新的版本是 [Gaussian 16 A.03](#)。

校超级计算中心统一购买安装的软件版本为 Gaussian Available Binary Versions for Rev. A.03, AMD/Intel Vendor, X86_64 CPU, Nvidia Tesla-K40/K80 GPU, Linux, 不含 Linda（即只能单节点内并行）。

二、测试平台：

曙光 TC4600 百万亿次超级计算系统新系统：[平台介绍](#)

三、软件版本使用：

系统使用 `module` 进行软件版本管理，查看安装软件版

本、配置、加载使用：

```
[pirate@tc4600v4 ~]$ module avail gaussian
----- /opt/Modules/app -----
gaussian/g09/D01          gaussian/g16/A03/AVX2    gaussian/g16/A03/SSE4
gaussian/g16/A03/AVX     gaussian/g16/A03/legacy
[pirate@tc4600v4 ~]$ module show gaussian/g16/A03/AVX2
-----
/opt/Modules/app/gaussian/g16/A03/AVX2:

module-whatis      Gaussian 16, version A03
setenv             G16BASIS /opt/gaussian16/avx2/g16/basis
setenv             GAUSS_ARCHDIR /opt/gaussian16/avx2/g16/arch
prepend-path      PATH /opt/gaussian16/avx2/g16
append-path       PATH /opt/gaussian16/avx2/g16/bsd
append-path       PATH /opt/gaussian16/avx2/g16/local
append-path       PATH /opt/gaussian16/avx2/g16/extras
setenv            _DSM_BARRIER SHM
setenv            GAUSS_LEXEDIR /opt/gaussian16/avx2/g16/linda-exe
setenv            GAUSS_EXEDIR /opt/gaussian16/avx2/g16
prepend-path      LD_LIBRARY_PATH /opt/gaussian16/avx2/g16/extras
prepend-path      LD_LIBRARY_PATH /opt/gaussian16/avx2/g16/local
prepend-path      LD_LIBRARY_PATH /opt/gaussian16/avx2/g16/bsd
setenv            PGI_TERM trace,abort
setenv            GAUSS_BSDDIR /opt/gaussian16/avx2/g16/bsd
setenv            GAUSS_SCRDIR /tmp
-----

[pirate@tc4600v4 ~]$ module load gaussian/g16/A03/AVX2
[pirate@tc4600v4 ~]$ which g16
/opt/gaussian16/avx2/g16/g16
```

设置 `setenv GAUSS_SCRDIR /tmp`，计算时临时文件生成在计算节点的 `/tmp` 目录下，减少存储读写，使用计算节点本地硬盘，可有效减少计算耗时。

若在 `~/.bashrc` 等环境变量文件中设置有多个高斯版本时，实际计算时调用的为最后一个高斯版本的可执行文件。

四、指令集测试：

G16 软件根据 CPU 指令集的不同，安装了不同参数的编译版本，测试对比：

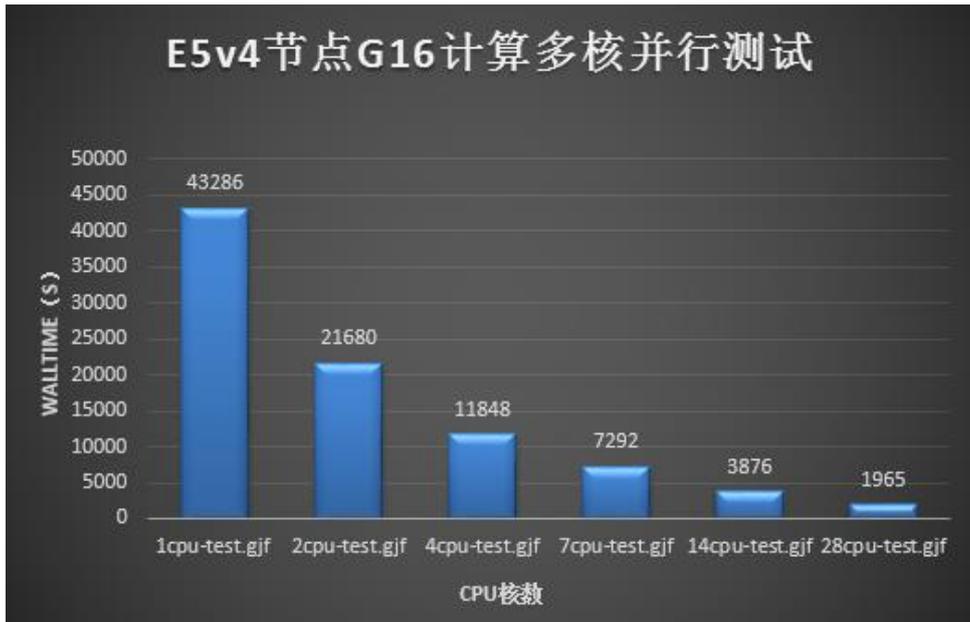


结果与讨论:

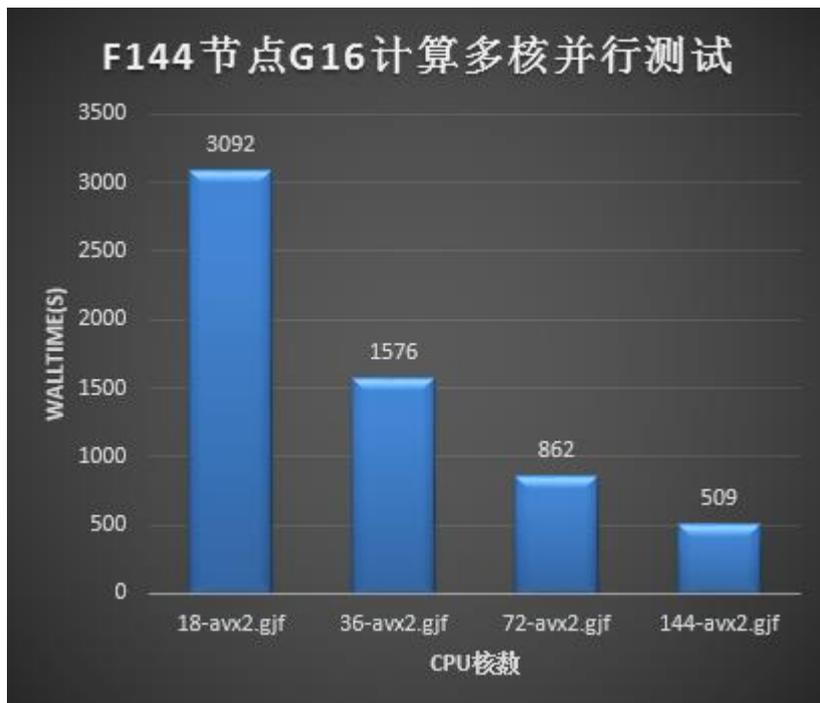
- 1) 计算效率: $avx2 > avx > sse4 > legacy$
- 2) 优先选择: TC4600 系统所有节点的 CPU 均支持 avx2 指令集, 因此在该系统上应优先选择使用 avx2 版本 G16。

五、CPU 节点相关测试:

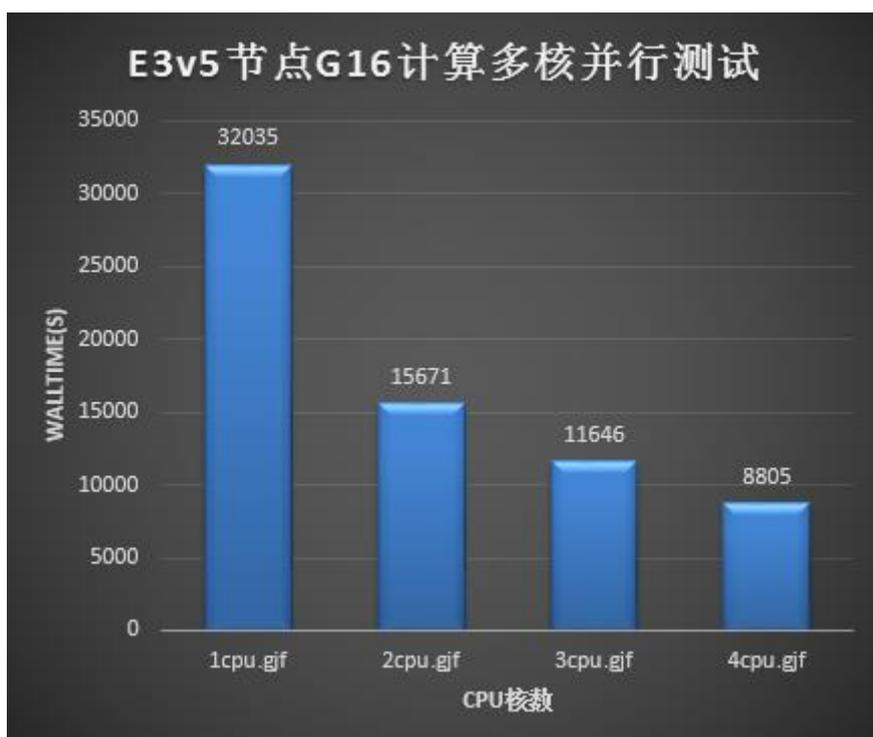
5.1: E5v4 节点使用 1、2、3、4、7、14、28 核并行计算



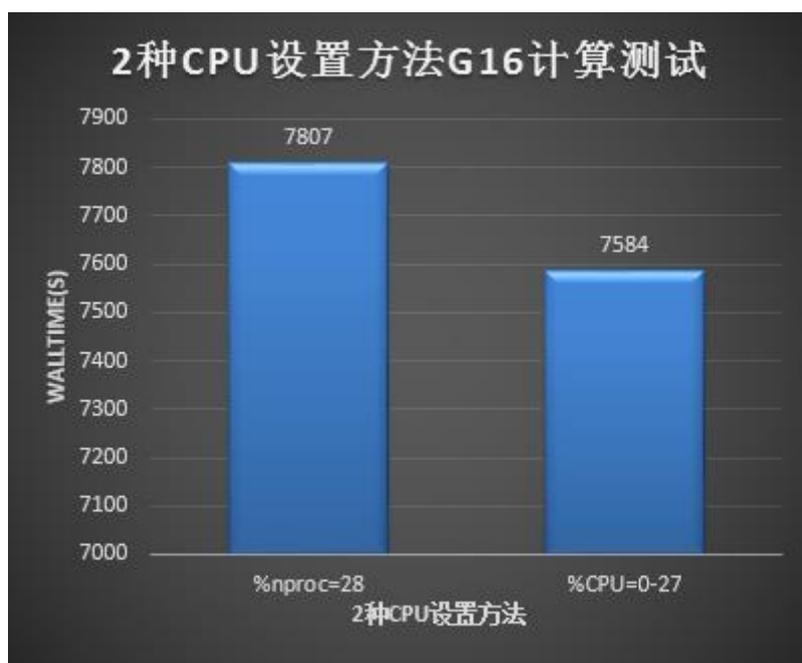
5.2: F144 胖节点使用 18、36、72、144 核并行计算



5.3: E3v5 节点使用 1、2、3、4 核并行计算



5.4: nproc 与 CPU 设置方法比较



结果与讨论:

- 1) 多类型 CPU 配置的节点测试中, G16 在单节点内并行计算时均具有很好的并行扩展性, 使用进程核数增加一倍时, 计算效率提高近 1 倍。

- 2) 在 F144 胖节点上，并行 144 核时仍具有很好的扩展性潜力，受限于单节点 CPU 核数，更多进程数的节点内并行没有进行测试。
- 3) 使用相同进程核数时，E3v5 节点计算效率优于 E5v4 节点，分析与其 CPU 主频较高有关，E3v5 CPU 主频 3.5GHz，而 E5v4 CPU 主频 2.4GHz。且 CPU 核数用满时具有睿频现象，E3v5 CPU 主频最大可达 4.0GHz，比较适合单节点并行的计算密集型应用。
- 4) 输入文件中 Link 0 部分设置使用 %CPU=** 方式，具有更好的计算效率，分析此种设置方式，进程与分配的 CPU 核绑定，可以减少在计算过程中的缓存损失。

六、GPU 节点相关测试：

6.1 设备支持：

TC4600 系统上配置 K80 GPU 卡节点为 k801-k804，驱动版本：375.39，CUDA 版本 8.0.61。查看 GPU 节点配置：

- 单节点 2 块 k80 gpu 卡
- 单卡片 2 个 GK210 核心
- 单卡片内存：2*12GB

注意：计算时 GPU 运算进程使用内存不可超过其使用的计算核心上内存（12GB），否则计算效率明显下降。

```

[pirate@K801 ~]$ nvidia-smi
Wed Apr 19 16:08:57 2017
+-----+
| NVIDIA-SMI 375.39                Driver Version: 375.39          |
+-----+-----+-----+-----+-----+-----+
| GPU  Name          Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf   Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+-----+
|  0   Tesla K80          Off      | 0000:05:00.0  Off    |          0          |
| N/A   40C    P0      88W / 149W | 8944MiB / 11439MiB | 100%      Default   |
+-----+-----+-----+-----+-----+-----+
|  1   Tesla K80          Off      | 0000:06:00.0  Off    |          0          |
| N/A   53C    P0     135W / 149W | 8944MiB / 11439MiB | 100%      Default   |
+-----+-----+-----+-----+-----+-----+
|  2   Tesla K80          Off      | 0000:85:00.0  Off    |          0          |
| N/A   52C    P0     120W / 149W | 8944MiB / 11439MiB | 100%      Default   |
+-----+-----+-----+-----+-----+-----+
|  3   Tesla K80          Off      | 0000:86:00.0  Off    |          0          |
| N/A   45C    P0     134W / 149W | 8944MiB / 11439MiB | 100%      Default   |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
| Processes:                                 GPU Memory |
|  GPU       PID    Type   Process name                               Usage      |
+-----+-----+-----+-----+-----+-----+
|    0         3136   C   /opt/gaussian16/avx2/g16/1502.exe         8940MiB   |
|    1         3136   C   /opt/gaussian16/avx2/g16/1502.exe         8940MiB   |
|    2         3136   C   /opt/gaussian16/avx2/g16/1502.exe         8940MiB   |
|    3         3136   C   /opt/gaussian16/avx2/g16/1502.exe         8940MiB   |
+-----+-----+-----+-----+-----+-----+

```

6.2 nvidia-smi topo -m 查看硬件架构

下图所示中最重要的参数显示是 CPU 亲密度（CPU Affinity）。此例中显示 0 号 GPU 卡的 GPU0、GPU1 核心与 0 号 CPU 的 0-13 号核心较近，1 号 GPU 卡的 GPU2、GPU3 核心与 1 号 CPU 的 14-27 号核心较近。

当使用 GPUs 时，每个 GPU 进程必须由一个特定的 CPU 进程控制。该 CPU 应该在物理架构上更靠近所控制的 GPU，并且 GPUs 不能共享 CPUs 控制。另外需要注意，用于控制 GPU 的 CPUs 不能再用于做进程计算核心。

```

K801@tc4600: ~# nvidia-smi topo -m

```

	GPU0	GPU1	GPU2	GPU3	CPU Affinity
GPU0	X	PIX	SOC	SOC	0-13
GPU1	PIX	X	SOC	SOC	0-13
GPU2	SOC	SOC	X	PIX	14-27
GPU3	SOC	SOC	PIX	X	14-27

```

Legend:
  X    = Self
  SOC  = Connection traversing PCIe as well as the SMP link
        between CPU sockets (e.g. QPI)
  PHB  = Connection traversing PCIe as well as a PCIe Host
        Bridge (typically the CPU)
  PXB  = Connection traversing multiple PCIe switches (without
        traversing the PCIe Host Bridge)
  PIX  = Connection traversing a single PCIe switch
  NV#  = Connection traversing a bonded set of # NVLinks

```

6.3 GPU-CPU 参数设置

G16 使用 GPU 的参数设置与 CPU 相似，在输入文件的 Link 0 部分使用 %GPU-CPU 参数：%GPU-CPU=gpu-list=control-cpus，其中 gpu-list 为 GPU 列表(用逗号隔开)，也可以是数值范围(例如，0-4,6)；control-cpus 为相似格式的控制 CPU 列表，这两列的内容是 GPU 及控制 CPU，例如

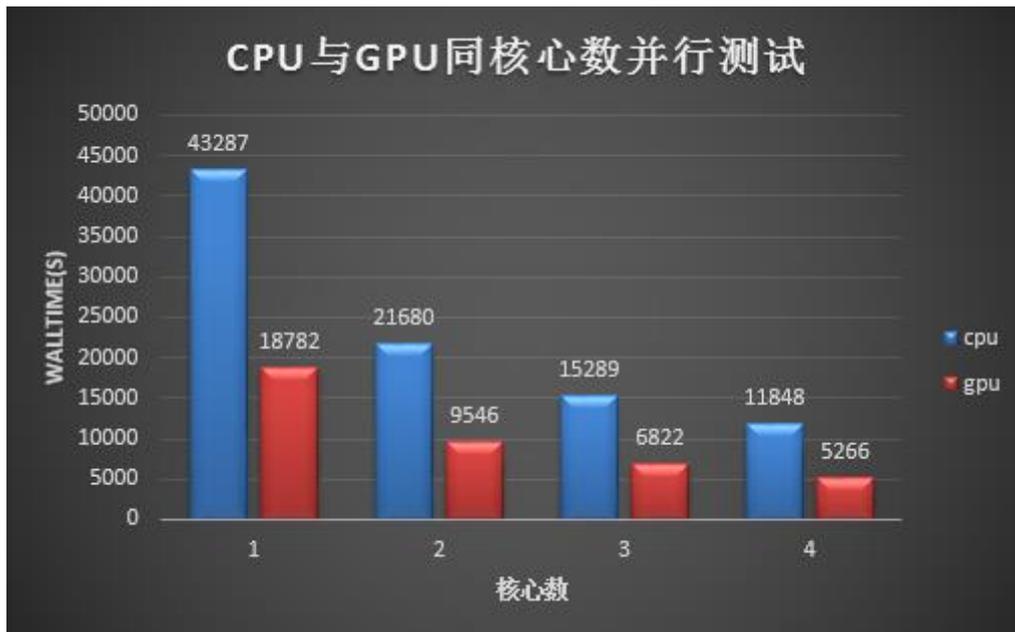
```

%CPU=0-27 或 %nproc=28    #使用 0-27 号 cpu 核
%GPU-CPU=0-3=0-3         #使用 0-3 号 GPU 卡和 0-3
                           号 CPU 核控制 GPU 进程

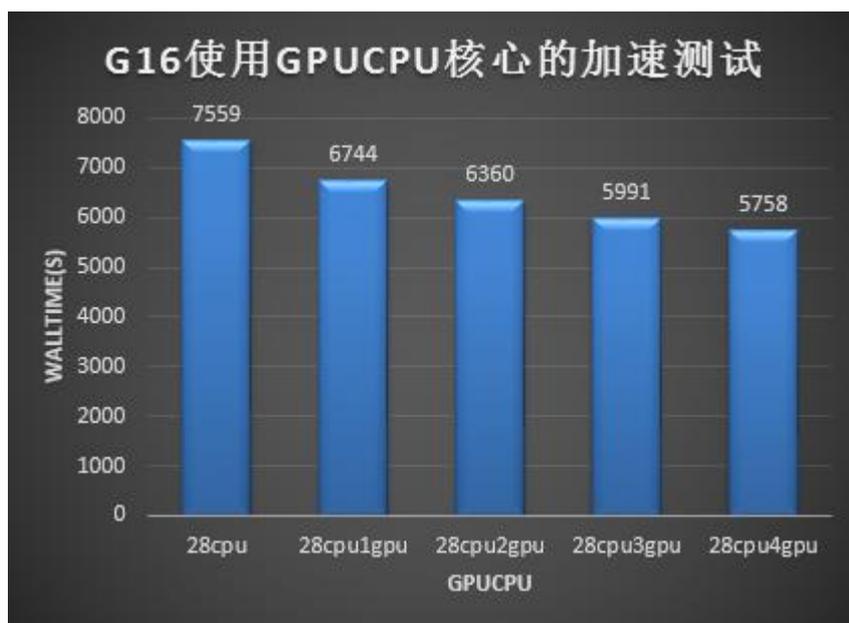
```

6.4 G16 使用 gpu 测试

6.4.1 CPU 与 GPU 同核心数并行测试



6.4.2 CPU 与 GPU CPU 加速效率测试



结果与讨论:

- 1) 仅使用 GPU 核心计算时，核数增加一倍，计算效率提升近一倍
- 2) 相同数量的 CPU 与 GPU 核心计算时，GPU 计算效率是 CPU 约 2.3 倍

3) 使用 `gpucpu` 设置计算时较 `cpu` 计算提高约 23.83% 的计算效率。

七、参考资料:

[1] <http://gaussian.com/products/>

[2] <http://www.emoltech.com/?p=51>