SGI ISV & Developer Technology Brief

Parallel Scaling Performance of Gaussian 09 on Shared Memory & Cluster Parallel Systems

Introduction

Gaussian 09 [1] is the latest in the Gaussian series of programs. It provides state-of-the-art capabilities for electronic structure modeling. G09 is capable of predicting many properties of atoms, molecules, and reactive systems such as energies, structures and frequencies, using *ab initio*, density functional theory, semi-empirical, molecular mechanics, and hybrid methods.

Gaussian performance, for a given chemical problem, depends on the system hardware, G09 installation and user choices. The hardware system configuration, such as the type and number of processors per node, memory per node, interconnection network, disk resources, and the size of the processor cache all impact performance. The available resources, G09 site installation and user options determine what G09 algorithms are used.

The *All Linux/UNIX* version of Gaussian 09 can run on single CPU systems and is one of the few commercially available programs in the discipline that has been parallelized using OpenMP [2] for shared-memory multiprocessor (SMP) systems. SMP nodes may consist of the typical commodity nodes with 2-4 sockets of 2-16 cores/socket or be a cache-coherent shared memory machine like the SGI® UV™ 2000 with 100's -1000's of cores being able to access system memory in a single node. For cluster and network parallel execution, the TCP Linda [3] parallel computing environment is used for parallelism across nodes while OpenMP is used on a node. Given the diverse set of machine hardware that G09 is targeted to run on, a primary goal in the implementation of parallelism in Gaussian is to have a single set of routines which implement an algorithm and which can be used for serial, SMP parallel, and cluster parallel computations.

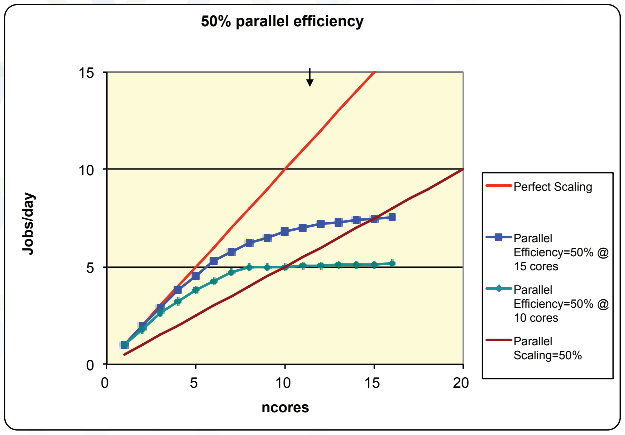


Figure 1

Performance Benchmarks

There are many ways that charts can be used to show the performance of a computer application. Although absolute time to solution matters, many times in a production data center, the balance of effective use of resources and good time to solution gives the best productivity choice. At SGI, we plot the performance as a rate, jobs/day and then use the 50% Parallel Efficiency point as a guideline as a reasonable choice for deciding on how may cores to run on in production.

% Parallel Efficiency = (t1-core / tn-core) / n * 100
Where tn-core is the elapsed time for an n-core run

Figure 1 shows two model problems, the blue with a 50% parallel efficiency at 15 cores and the green at 10 cores. The 50% parallel efficiency is marked with a black arrow for the G09 scaling presented in this paper.

The two hardware systems used for this technology brief are the shared memory SGI® UV™ 2000 and SGI® ICE™ X cluster. Below are the specifications of each system.

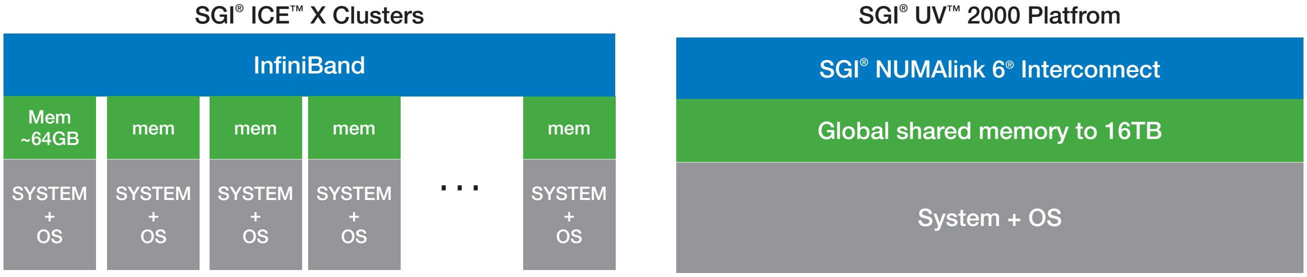
Shared Memory — SGI UV 2000

- 1 node – 512 cores
- 32 – 2 socket boards HUB 3.0; 8 cores / socket
- Intel® Xeon® E7-4627v2 (Ivy Bridge)
  – 3.3GHz 16MB L3 per 8 core socket
- Total Mem: 4 TB Speed: 1867 MHz
- SLES11 SP3 OS, SGI ProPack 7 SP2
- SGI NUMALink® 6 interconnect
- Hyper Threading not available
- Turbo Boost enabled but not used

Cluster — SGI ICE X

- 144 nodes- 2304 cores
- 144 - 2 socket x 10- cores/socket
- Intel® Xeon® E5-2690v2 (Ivy Bridge)
  – 3.0GHz 25MB L3 per 10 core socket
- Mem: 128 GB/node Speed: 1867 MHz
- SLES11 SP3 OS, SGI ProPack 7 SP2
- InfiniBand - FDR
- Hyper Threading enabled but not used
- Turbo Boost enabled but not used

Figure 2 shows a schematic of the characteristics of a cluster compared to an SMP system.



- Each system has its own memory and OS
- Nodes communicate over commodity interconnect
- Inefficient cross-node communication creates bottlenecks
- Coding required for parallel code execution

- All nodes operate on one large shared memory space
- Eliminates data passing between nodes
- Big data sets fit entirely in memory
- Less memory per blade required
- Simpler to program
- High Performance, Low Cost, Easy to Deploy

Figure 2

This technology brief presents SMP vs. clusters performance data for these 3 chemical systems. Figure 3 a,b,c give the 3D chemical structure.

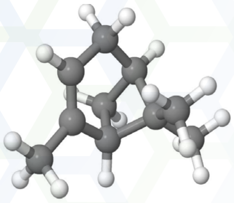


Figure 3a

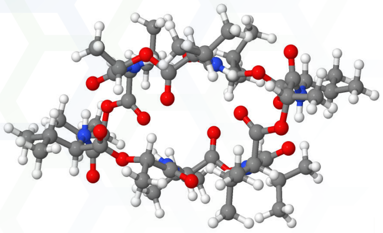


Figure 3b

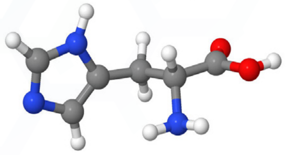


Figure 3c

α-pinine HF, DFT, FREQ

- $C_{10}H_{16}$ 26 atoms / 10 heavy
- b3lyp/6-311++G(3df,3pd) scf=tight
  - 678 basis functions
- HF (16mw/core), DFT (16mw/core), FREQ(32mw/core)

Valinomycin DFT Force

- $C_{54}H_{90}N_6O_{18}$ 168 atoms / 78 heavy
- rb3lyp/3-21g force scf=novaracc
  - 882 basis functions
- test397 in Gaussian QA datasets

Histidine-H+ complex: MP4

- $C_6H_9N_3O_2$ 20 atoms / 11 heavy
- MP4/6-31G(d) scf=noIncore
  - 183 basis functions
- Output from test310 of Gaussian QA datasets

Looking at the performance for a simple α-pinine HF calculation, we present it two ways, Jobs per day by core and Jobs per day by node. For the SMP UV 2000 system a blade is used for the “node” count. In Figure 4 a and b you can see that the UV 2000 SMP parallel scaling is better than the ICE X Cluster. This is due to the different methods of dividing up the problem. For the SMP all the cores have access to all the memory requested and for the clusters using Linda, the per node memory is replicated on each node and only the cores on that node share the memory. Different problems show different response to these two methods of dividing the problem.

Often technical users of Gaussian talk about per core scaling but sales departments or computer centers talk about per node performance. Machines are generally purchased and allocated to the users through queuing systems on a per node basis. Users should realize that the chart can look different when comparing systems with different resources per node and decide what presentation is best for their purpose. In this case UV 2000 has 16 cores per node and a clock speed of 3.3GHz compared to the ICE-X 20 cores per node and 3.0GHz clock speed. Comparing the α-pinine HF per core and per node scaling plots in Figure 4 shows that up 32 cores the performance is basically the same for the two systems. The SMP continues to remain above 50% parallel efficiency to 144 cores – 9 blades whereas the 50% parallel efficiency is hit for the cluster at 80 cores or 4 nodes.

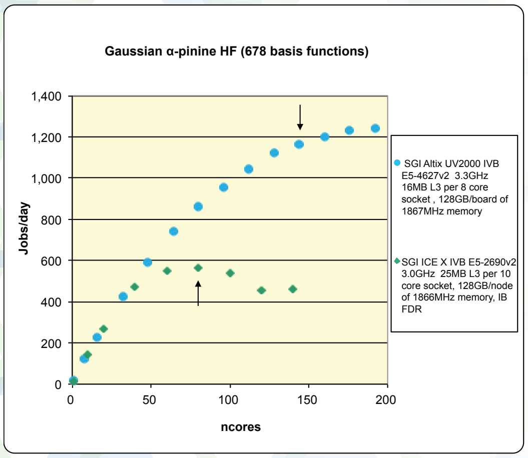


Figure 4a: Jobs per day by core

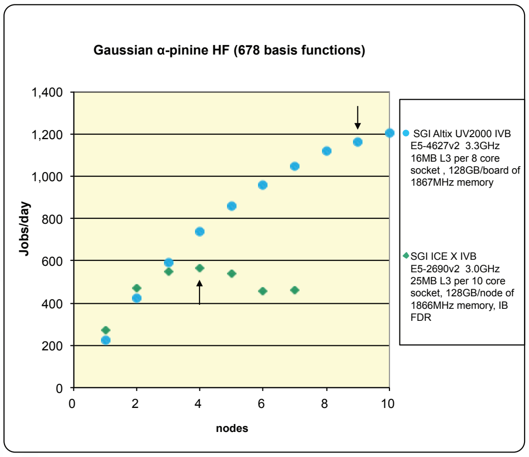


Figure 4b: Jobs per day by node

Figure 5 a and b show two more complex chemical problems, single point α-pinine DFT and Valinomycin DFT with forces. You can see that as complexity is added into the calculation, the parallel scaling is reduced.

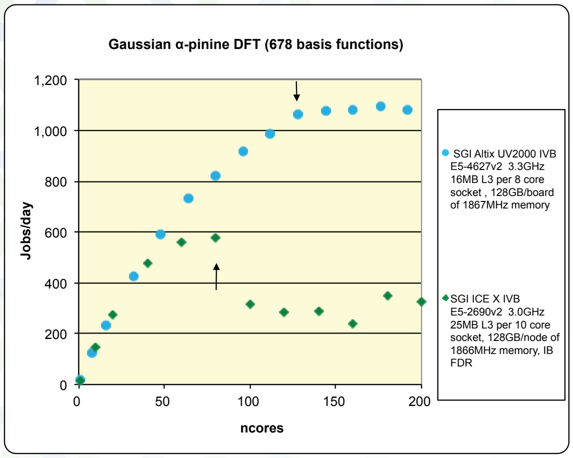


Figure 5a

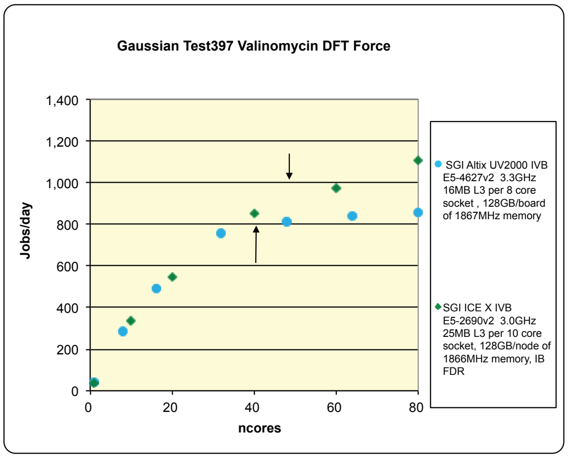


Figure 5b

Figure 6a shows the parallel scaling for the histidine-H+ complex MP4 calculation and you see that for SMP, the performance improves with increasing number of cores and has a 50% parallel efficiency at 40 cores. The clusters/Linda computation does not scale above one node, 20 cores. This is due to a lack of parallel algorithm for the primary computation using Linda.

For the Histidine-H+ complex MP4 calculation, Figure 6b shows the impact of memory on performance. G09 scales back the number of processors used in different parallel sections if enough memory is not available for their workspace. A user needs to judge the required memory for a single core calculation and then scale it up for the number of cores to be used in shared memory.

The memory needed per node, %mem, for parallel scaling should be set to be:
Mem=Mem0 + sf * (ncores – 1) sf ~ ½ to 1
Where Mem0 is the memory to run effectively on 1 processor

For SMP, there is only one node and the %mem is shared by all the cores used. For clusters, Linda parallel used %mem per node and the memory is replicated on each node.

Disk usage can impact G09 performance for problems that use large scratch files assigned to GAUSS_SCRDIR. The user should choose the fastest disk available for the size problem.

- Memory disk or local node disk if possible
- Not global file system if at all possible
- For MP2 calculations, all nodes read scratch files of the same name, either one mounted file system or local files with same name on each node. These can be defined in the .cshrc or .profile.

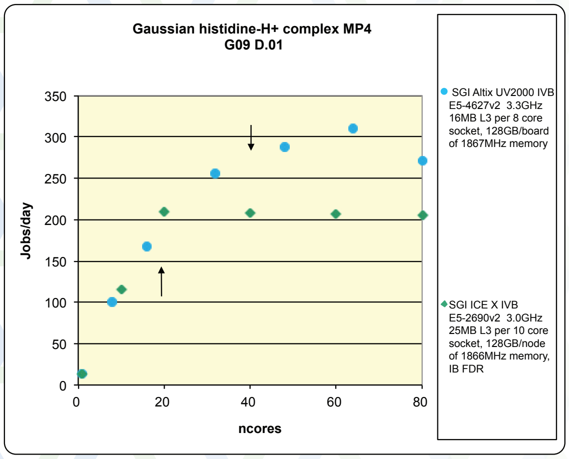


Figure 6a

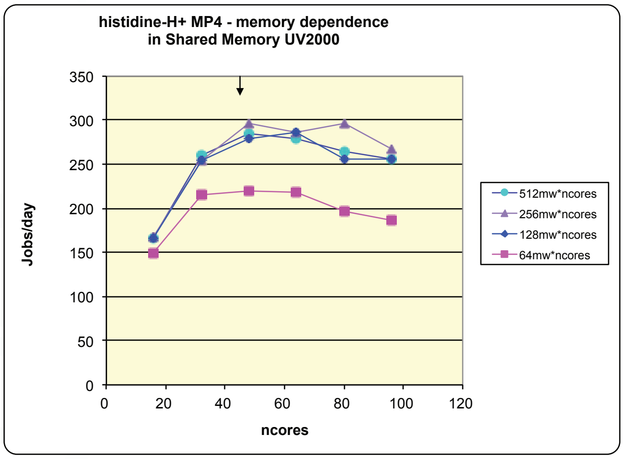


Figure 6b

Conclusions

Gaussian 09 is a research tool that incorporates diverse theoretical methods into an efficient computer program to calculate structure and a wide variety of properties for chemical systems. It is one of the most widely used applications in high performance computing centers around the world. This SGI technology brief demonstrates the scalability of Gaussian on both clusters and SMP platforms. Some primary use pathways through Gaussian scale on 50 to 150 processor cores. The performance charts above highlight how large SMP architectures, like the SGI UV 2000, can be a very effective platform for running Gaussian jobs.

Generalizations on parallel scaling for SMP and clusters machines show less scaling with:

- Increase in chemical model sophistication HF, DFT, MP2, MP4, CISD,…
- Increase in the basis set size
- Use of incore rather than recalculated integrals (direct) algorithms
- Decrease in hardware cache size/core, when using cache “chunking” algorithms.
- Decrease in interconnection network speed, Proprietary, InfiniBand, GigE, Ethernet

Gaussian 09 users make many choices that impact performance like the algorithm through input keywords, kind and size of parallelism, memory usage, and file placement. They need to use their resources wisely to get the best productivity possible.

User support for Gaussian is available from:

- Computer center and site support staff
- Gaussian manuals, http://www.gaussian.com/
- Gaussian helpdesk help@gaussian.com
- Hardware vendor applications staff like Dr. Jan Fredin, Applications Engineer, SGI jfredin@sgi.com

References:

[1] Gaussian 09, Revision D.01, M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox, Gaussian, Inc., Wallingford CT, 2009.
[2] OpenMP Application Program Interface, Version 3.0 May 2008, http://www.openmp.org/mp-documents/spec30.pdf
[3] TCP-Linda with Gaussian http://www.gaussian.com/g_prod/linda.htm
[3] SGI ICE X product guide http://www.sgi.com/products/servers/ice/x/
[4] SGI UV 2000 product guide http://www.sgi.com/products/servers/uv/

Global Sales and Support: sgi.com/global

©2015 Silicon Graphics International Corp. All rights reserved. SGI and the SGI logo are registered trademarks or trademarks of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries. All other trademarks are property of their respective holders. 13082012 4371 03022015