

Estado Gaussian

- Actualizado (16.12.2005)

Gaussian 03

A execución de Gaussian (tanto na súa versión 03 como 98) nos servidores do CESGA, representa preto dun 70% do consumo de horas de computación no período 1 de xaneiro de 2004 a 30 de setembro de 2004, principalmente desde a súa instalación no novo HP Superdome. A incorporación de Gaussian 03 (<http://www.gaussian.com>), con novas funcionalidades, máis adaptada ás arquitecturas existentes no CESGA, representa un cambio tecnolóxico importante na aplicación. Nesta nova versión cámbiase o sistema de paralelización, pasando á utilización de fíos (threads en inglés) a través do estándar OpenMP (<http://www.openmp.org>). Isto significa, por exemplo, que xa non utiliza memoria compartida, xa que todos os fíos poden acceder á memoria reservada polo programa directamente. Mantense a posibilidade de execución de multiproceso a través de LINDA (<http://www.lindaspaces.com/products/linda.html>), pero esta funcionalidade non está dispoñible en ningún servidor do CESGA.

Fig. 1. Porcentaxe de horas de uso de CPU.

Dada a importancia que ten a utilización deste software no CESGA, é importante coñecer adecuadamente a forma de mellorar o rendemento, de tal forma que se optimice o uso conxunto e individual dos servidores. Por iso analizouse o comportamento do Gaussian para algúns dos problemas tipo habituais, en función dos parámetros máis importantes que poden ser seleccionados polo usuario: a memoria a utilizar e o número de fíos (ou procesadores). Esta análise soa realizase para Gaussian 03. Se se necesita a mesma información para Gaussian 98, consulte a páxina Web <http://www.cesga.es/ga/CalIntensivo/gaussian98.html>.

1. Utilización da memoria

A selección de memoria virtual a utilizar na execución dun problema fíxase a través da directiva %MEM. Esta ha de estar situada antes de cada link que se inclúa no ficheiro. En caso de que non se fixe, Gaussian reservará por defecto 6 MegaWords (6 millóns de palabras, onde cada palabra corresponde a 8 Bytes). A sintaxe deste comando é

```
%mem=<numero><unidade>
```

onde número é a cantidade de memoria a utilizar (no exemplo 6) e unidade pode ser KB, MB, GB (cando a cantidade se expresa en Bytes) ou KW, MW, GW (cando se expresa en palabras. Unha palabra equivale a 8 bytes no HPC320 e SUPERDOME). Por exemplo:

```
%mem=6mw  
#P HP/ST0-3G FORCE
```

En función do tipo de problema e da memoria seleccionada, Gaussian escolle o modo de funcionamento de forma automática que considera máis adecuado, a menos que se lle especifique o contrario. Os modos de funcionamento poden ser:

Directo. Todas as integrais necesarias para a resolución son calculadas cando se necesitan. É o modo por defecto.

Convencional. As integrais calcúlase unha vez e acumúlanse en disco. Este sistema é máis lento que o directo, polo cal está desaconsellado

InCore. As integrais necesarias acumúlanse en memoria (coidado, se se selecciona este método de cálculo, se non

hai suficiente memoria o cálculo non se executará).

Semi-directo. Fundamentalmente utilizado en cálculos MP. É unha combinación do modo directo co convencional. Para facer unha estimación da memoria mínima necesaria para un cálculo, hase de utilizar a fórmula

$$M + 2 Nb^2$$

Que dará a cantidade de memoria en palabras. Nb é o número de bases do problema, mentres que M é unha cantidade dependente do mesmo que se ha de seleccionar da táboa dada no manual (http://www.gaussian.com/g_ur/m_eff.htm):

Job Type

Highest Angular Momentum Basis Function

f functions

g functions

h functions

i functions

j functions

SCF Energies

4 MW

4 MW

9 MW

23 MW

~60 MW

SCF Gradients

4 MW

5 MW

16 MW

38 MW

SCF Frequencies

4 MW

9 MW

27 MW

MP2 Energies

4 MW

5 MW

10 MW

28 MW

~70 MW

MP2 Gradients

4 MW

6 MW

16 MW

38 MW

MP2 Frequencies

6 MW

10 MW

28 MW

Para o caso dun problema de cálculo de frecuencias, existe ademais a utilidade `freqmem` que calcula a memoria mínima en función do tipo de problema e número de átomos.

Fig.2 . Tempos de cálculo versus a memoria dispoñible para 188 bases.

Porén, existe unha forte dependencia da velocidade do cálculo co tipo de método que depende fundamentalmente da memoria dispoñible. Por exemplo, pódese apreciar na fig. 2 que para un cálculo DFT ou HF, incrementar a reserva de memoria por encima do mínimo necesario non axuda a mellorar a velocidade do mesmo (de feito en Gaussian 98, a eficiencia do mesmo diminúe de forma apreciable), polo que se estarían a desperdiciar recursos innecesariamente. Porén, esta situación cambia cando a cantidade reservada é suficiente para acumular as integrais en memoria; cando Gaussian cambia o método de resolución a InCore. A mellora no rendemento é substancial neste caso (próximo ó 100%).

Para poder calcular a memoria necesaria en Gaussian 03 para realizar o problema utilizando o método InCore, pódense utilizar as fórmulas:

$Nb4/8 + 500.000$ para o caso de Close-Shell

$Nb4/4 + 500.000$ para o caso de UHF ou ROHF

Que tamén se poden aplicar para os métodos GVB e MCSCI.

Fig. 3. Consumo de memoria de DFT versus o número de bases.

Na figura 3 pódese ver o consumo de memoria para un problema DFT en relación ás dúas fórmulas. Apréciase que a primeira fórmula é un límite máximo para todos os casos. Para se asegurar de que se utiliza o cálculo en forma Incore, pódense utilizar os comandos SCF=Incore para o caso DFT ou HF, mentres que para outros é necesario especificalo directamente (véxase o manual para cada caso). Por exemplo, para MP2 o cálculo realizaríase InCore co comando MP2=Incore.

2. Execución en paralelo.

Como xa se dixo anteriormente, Gaussian 03 utiliza a tecnoloxía OpenMP (única posibilidade existente no CESGA) ou memoria distribuída (é dicir, varios procesos que non comparten memoria, a través de LINDA, non dispoñible no CESGA). Na versión actual, entre os enlaces (links) paralelizados están:

Link

Descrición

L502

Solución de la ecuación SCF iterativa

L703

Derivadas primera y segunda (spdf)

L914

Estados excitados de CI-singles, RPA y Zindo. Estabilidad SCF

L1002

Solución de las ecuaciones CPHF; cálculo de propiedades (como NMR)

L1110

Contribución de las integrales de $2e$ a F_x

En xeral, tendo en conta isto, todos os cálculos de enerxía, gradientes e frecuencia a través de SCF (isto é, HF, DFT e CIS) xunto con TD-DFT están paralelizados. Outros métodos, como Möller-Plesset, non melloran suficientemente cando se executa con máis dun procesador utilizando fíos, polo que se desaconsella o seu uso neses casos.

Para executar Gaussian en paralelo, hase de utilizar a nova clave

`%NprocShared`

Por exemplo, para a execución con catro fíos, utilizaríase

`%NProcShared=4`

É importante notar que para que isto sexa eficiente e que cada fío se poida executar de forma independente nun procesador dedicado, é necesario solicitalo tamén na petición do traballo a través da directiva `num_proc` do comando `qsub`. Se non se fixese así, Gaussian crearía igualmente 4 fíos, competindo eles mesmos polo uso do procesador e polo tanto degradando o rendemento.

Para executar con máis dun procesador é necesario ter en conta que:

1. A memoria mínima necesaria cando o cálculo se realiza de forma directa se incrementa (Gaussian recomenda multiplicar a memoria mínima calculada da forma anterior polo número de fíos a utilizar).
2. Cando o cálculo se pode facer InCore, a memoria adicional necesaria é insignificante, aínda que é necesario aumentala un pouco en relación á prevista para o cálculo.

Fig. 4: Tempos de cálculo para un problema DFT e HF directo e incore. O escalado defínese como a razón entre o tempo consumido para 1 procesador dividido entre o tempo para n procesadores. A eficiencia é o escalado dividido polo número de procesadores.

Na figura 4 pódese ver o comportamento paralelo de dous métodos de uso habitual. Pódese observar que en ambos os dous casos se reduce en tempo de cálculo, mellorando sensiblemente no caso directo, alcanzando para catro procesadores un rendemento case idéntico. Existe unha dependencia entre a eficiencia da paralelización co número de bases (en realidade co número de primitivas); a maior número de bases, mellor eficiencia da paralelización. Isto é debido a que o programa está máis fracción do tempo executando zonas de código paralelizadas.

3. Paso de integrais ó cálculo de frecuencias.

O cálculo de frecuencias pode mellorar sensiblemente coa memoria. Para calcular a memoria necesaria pódese utilizar o comando `freqmem` comentado anteriormente (que proporciona un valor exacto) ou estimala utilizando a fórmula seguinte para calcular o número de palabras

$3NANb^2 + 500.000$

onde NA é o número de átomos e Nb o número de bases.

Se o problema admite a execución incore, é posible pasar o resultado deste cálculo ó módulo de frecuencias a través do comando `SFC=(InCore, pass)`, a conta de consumir máis espazo en disco (da orde de $Nb^4/8$).

4. Matrices Sparse

Cando o problema é demasiado grande para acumular as integrais completas en memoria, é posible utilizar o método de matrices sparse, a conta de eliminar do cálculo as interaccións cun valor inferior a un dado (10-10 por defecto). Neste caso, só os valores por encima do de corte se manteñen en memoria. Gaussian recomenda utilizar só este método cando o problema ten máis de 400 átomos (200 para cálculos semiempíricos como o AM1). Porén, dado que o consumo de memoria depende das interaccións calculadas, non é posible a priori calcular a memoria que será necesaria. En caso de que se queira utilizar este método, é necesario indicalo na descrición do problema coa clave sparse (http://www.gaussian.com/g_ur/k_sparse.htm).

5. Reinicio do traballo.

Dadas a limitacións de tempo de execución existentes, é posible que algún cálculo non acabe adecuadamente por esa limitación. Para reiniciarlo desde o último link que se executou é necesario facer varias accións. A primeira é que se ha de solicitar ó final do traballo a copia dos ficheiros ó directorio local. Por defecto, estes créanse no directorio temporal (e por cuestións de eficiencia, procure non cambiar os ficheiros rwf fose desta localización). Para iso, ó enviar o traballo ó sistema de colas inclúa a opción ó comando qsub:

```
-v COPIA=$HOME/destino
```

Deste modo, a copia dos ficheiros do directorio temporal realizarase ó finalizar o traballo de forma inesperada. Por exemplo, supoñamos que se envía o traballo seguinte:

```
#p rhf/6-311g* opt freq test geom=modela scf=direct
```

```
water with restart
```

```
0,1
```

```
ou h h
```

coas seguintes opcións de qsub

```
qsub -l num_proc=1,s_rt=00:00:03,s_vmem=500M,h_fsize=1G -cwd -v COPIA=$PWD -ou 1.log -e 1.err
```

Unha vez pasados tres segundos de execución (tempo límite marcado na petición), o traballo abortaría no link onde estase a executar de forma abrupta. Ó ter solicitado a copia dos ficheiros xerados no scratch, terase no directorio de envío os seguintes ficheiros:

```
total 38658
```

```
drwxr-xr-x 2 agomez cesga 1024 Oct 6 13:39 .
```

```
drwxr-xr-x 32 agomez cesga 2048 Sep 29 11:45 ..
-rw-r--r-- 1 agomez cesga 107 Oct 6 13:38 1.com
-rw-r--r-- 1 agomez cesga 0 Oct 6 13:39 1.err
-rw-r--r-- 1 agomez cesga 27571 Oct 6 13:39 1.log
-
-rw----- 1 agomez cesga 111 Oct 6 13:38 Gau-7855.inp
-rw----- 1 agomez cesga 532480 Oct 6 13:38 Gau-7860.chk
-rw----- 1 agomez cesga 0 Oct 6 13:38 Gau-7860.d2e
-rw----- 1 agomez cesga 0 Oct 6 13:38 Gau-7860.int
-rw----- 1 agomez cesga 18653184 Oct 6 13:38 Gau-7860.rwf
-rw----- 1 agomez cesga 532480 Oct 6 13:38 Gau-7860.scr
```

Para rearrancar o traballo, hai que indicar a Gaussian que utilizamos o ficheiro RWF correspondente, para o cal se definirá o seguinte ficheiro de comandos de Gaussian:

```
%rwf=Gau-7860
```

```
#p restart
```

Impulsando o traballo con esta opción, Gaussian continuará a partir do último link que se executase completamente antes da interrupción. Por exemplo, no caso de exemplo, o último link executado no primeiro envío era o I301.exe, estando en execución o I302. Ó rearrancar, Gaussian reinicia no link I302 que non acabara.

6.Recomendacións

Cos resultados previos, pódense obter as seguintes conclusións:

1. Para cálculos HF, DFT, TD-DFT e CIS, se a memoria necesaria para facer o cálculo utilizando o método InCore é inferior ou igual a 16GB (aínda que G03 poida utilizar máis, xa que non existe a limitación que había en G98), tente utilizar este método. Se ademais é posible paralelizar o cálculo, calcule o número de procesadores como a razón entre a memoria necesaria en GB dividida por 4, cun límite de 4 CPUs para Incore. Teña en conta que cantos máis procesadores solicite, o tempo de espera para executar o traballo se incrementa, xa que é necesario que estean dispoñibles ó mesmo tempo.

2. Non utilice varios procesadores para métodos combinados como G3 ou para cálculos Möller-Plesset, xa que a ganancia é insignificante (se necesita facer cálculos MP en paralelo, avalíe a posibilidade de utilizar outros programas, como NWCHEM, que si teñen paralelizada esta opción).

3. Ó enviar o traballo a cola, asegúrese de seleccionar os parámetros adecuados. Na opción `s_vmem` do comando `qsub`, inclúa a memoria que calculase máis 200MB (para o código do programa).

4. Se o seu problema necesita máis de 16GB para calcular utilizando o método Incore, comprobe se é paralelizable e utilice preferentemente esta opción cun cálculo directo.

5. Tente enviar sempre a execución coa opción de copia de ficheiros. En caso de que o traballo funcione correctamente non obterá ningunha saída adicional, xa que Gaussian elimina os ficheiros temporais cando acaba correctamente mentres que se acaba de forma incorrecta, poderá reiniciar o traballo desde o último link executado correctamente.

En calquera caso, se ten calquera dúbida ou suxestión, non dubide en contactar connosco a través .