Gaussian Tutorial: Estimating Resource Requirements

Carlos P. Sosa
IBM
and
Patton Fast
Supercomputing Institute
Contents

- IBM SP Overview
- Review common methods
- Alternative algorithms and why the program selects them
- Estimating resource usage
- Allocating memory and disk resources for good performance
- Running Gaussian
IBM SP Overview

78 WinterHawkII, 375 MHz
17 NightHawk, 222 MHz
4 Silver nodes, 333 MHz

Nodes are 4-way
Silver nodes are 333 MHz, 604e, 32-bit
Gaussian Design

- *Gaussian98* has been designed to work efficiently given a variety of computer configurations.

- The program attempts to select the most efficient algorithm given the memory and disk constraints imposed upon it.

- *Gaussian98* offers a wide variety of algorithms.

- An understanding of the possibilities and tradeoffs can help you to achieve optimal performance.
Gaussian Input

%chk=h2o
%nproc=1
%mem=8MW
#p hf/sto-3g opt

water optimization

0 1
 0
 h 1 oh
 h 1 oh 2 a

oh = 0.89
 a = 105.

Control files & system resources
Choice of computational model
Type of calculation
Charge and multiplicity
Coordinates
Link 0 Commands

%mem=N  Sets the amount of dynamic memory used to N works (8N bytes). The default is 6MW. N may be followed by a units designation: KB, MB, GB, KW, MW or GW

%nproc(l)=N  Requests that the job use up to N processors

%chk=file  Locates and names the checkpoint file

%rwf=file  Locates and names a single, unified Read-Write file

%KJob LN [M]  Tells the program to stop the run after the Mth occurrence of link N

%save  Causes Link 0 to save scratch files at the end of the run

%subst LN dir  Tells Link 0 to the executable for a link from alternate directory
**Sequence of Programs**

- **L1**
  - Read and parse route section

- **L101**
  - Read in molecule specification

- **L202**
  - Determine molecular symmetry

- **L301**
  - Set up basis set, compute one-electron integrals

- **L302**
  - Generate initials orbitals

- **L303**

- **L401**
  - Solve SCF equations

- **L502**

- **L601**
  - Assign orbital and wavefunction symmetries, print orbitals, and perform Mulliken population analysis

- **L9999**
Hartree-Fock Energies

\[ E_{HF} = \frac{\langle \Psi_o | H | \Psi_o \rangle}{\langle \Psi_o \Psi_o \rangle} ; \quad \frac{\partial E_{HF}}{\partial C_{\mu i}} = 0 \]

\[ \sum_v F_{\mu \nu} C_{vi} = \varepsilon_i \sum_v S_{\mu \nu} C_{vi} \]

\[ F_{\mu \nu} = h_{\mu \nu} + \sum_{\lambda \sigma} [(\mu \nu \parallel \lambda \sigma) - (\mu \sigma \parallel \lambda \nu)] P_{\lambda \sigma} \]

\[ P_{\lambda \sigma} = \sum_i C_{\lambda i}^* C_{\sigma i} \]

\[ E_{HF} = \sum_{\mu \nu} P_{\mu \nu} h_{\mu \nu} + \frac{1}{2} \sum_{\mu \nu \lambda \sigma} \left[ P_{\mu \nu}^T P_{\lambda \sigma}^T - P_{\mu \sigma}^a P_{\lambda \nu}^a \right] (\mu \nu \parallel \lambda \sigma) + V_{nuc} \]
Two-electron Integrals

Traditional approach:

- Formally $O(N^4)$; often less in practice
- Atomic Orbital (AO) basis:
  - Integrals in AO basis stored on disk in random order
  - Traditional approach for SCF
  - Sorting into standard order involves substantial extra storage
- Molecular Orbital (MO) basis:
  - Integrals transformed from AO to MO
  - Stored on disk in addition to AO integrals
  - Traditional approach beyond SCF
Two-electron Integrals

- **Incore:**
  - AO integrals are stored in main memory
  - Canonical order, including zeros
  - No I/O
  - Ordering facilitates optimization

- **Direct:**
  - Recompute integrals as needed
Direct SCF

- Traditional approach:
  - Integrals are expensive
  - Compute integrals once and store
  - Read integrals once each SCF iteration
- Almlof:
  - Integrals aren't that expensive
  - I/O can be slow
  - Amount of disk limits size of calculations
  - Recompute integrals each SCF iteration
- Can be clever about neglecting integrals if their use is known:

\[ \Delta F^{(n)} = F^{(n)} - F^{(n-1)} = \sum \Delta P_{\lambda\sigma}^{(n)} \langle \mu \lambda \| v\sigma \rangle \]
**Conventional SCF**

1. Setup
2. Evaluate Integrals
3. Form Fock Matrix
   - *no*
   - New Density: Converged?
     - *no*
     - Evaluate Integrals
     - Form Fock Matrix
     - I/O
     - Disk
     - scf=conventional
   - *yes*
   - New Density: Converged?
     - *yes*
     - Populations,...
Direct SCF

Setup

Form Fock Matrix

Evaluate Integrals

New Density: Converged?

no

yes

Populations,...

scf=direct (default)
Incore SCF

Setup

Evaluate Integrals

Form Fock Matrix

New Density: Converged?

no

yes

Populations,...

scf=incore

MEMORY

- $N^4/8$ memory ($N^4/4$ for open-shell)
- Fast
- Memory needed:
  - 100 basis functions = 100 MB
  - 200 basis functions = 1600 MB
  - 300 basis functions = 8100 MB
Direct versus Conventional SCF

Almlof and Alrichs: SCF is not $N^4$!
- Direct SCF is faster than conventional for large cases
\( C_nH_{n+2} \) Hydrocarbons

<table>
<thead>
<tr>
<th>n</th>
<th>Conventional (Sec.)</th>
<th>InCore (Sec.)</th>
<th>Direct (Sec.)</th>
<th>Basis Functions</th>
<th>File Sizes(C)</th>
<th>File Sizes(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.8</td>
<td>31.6</td>
<td>5.3</td>
<td>23</td>
<td>22</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>4.5</td>
<td>32.4</td>
<td>7.4</td>
<td>42</td>
<td>24</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>7.1</td>
<td>34.1</td>
<td>12.8</td>
<td>61</td>
<td>36</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>12.8</td>
<td>38.1</td>
<td>22.2</td>
<td>80</td>
<td>66</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>23.5</td>
<td>43.7</td>
<td>35.8</td>
<td>99</td>
<td>116</td>
<td>24</td>
</tr>
<tr>
<td>6</td>
<td>43.0</td>
<td>52.9</td>
<td>54.8</td>
<td>118</td>
<td>198</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>76.5</td>
<td>66.1</td>
<td>79.2</td>
<td>137</td>
<td>312</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>127.3</td>
<td>83.7</td>
<td>111.5</td>
<td>156</td>
<td>466</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>207.1</td>
<td>-</td>
<td>149.6</td>
<td>175</td>
<td>646</td>
<td>36</td>
</tr>
<tr>
<td>10</td>
<td>343.4</td>
<td>-</td>
<td>194.3</td>
<td>194</td>
<td>862</td>
<td>36</td>
</tr>
</tbody>
</table>

Timings on an IBM WinterHawkII, 375 MHz
Gaussian98 Rev. A10
Incore memory: 900MB
Conv. & Direct memory: 48MB
$C_nH_{n+2}$ Hydrocarbons CPU (C & D)

![Graph showing comparison between Conventional and Direct methods for $C_nH_{n+2}$ hydrocarbons on IBM Power3 G98 A.10. The graph plots the time in seconds on the y-axis against n on the x-axis. The Conventional method shows a higher time trend compared to the Direct method.]
$C_nH_{n+2}$ Hydrocarbons CPU (C, D, & I)

IBM Power3
G98 A.10
$C_nH_{n+2}$
$C_nH_{n+2}$ Hydrocarbons File Sizes

IBM Power3 G98 A.10 $C_nH_{n+2}$
Hartree–Fock Gradients

\[ E_{\text{HF}}^x = \sum_{\mu\nu} P_{\mu\nu} h_{\mu\nu}^x + \frac{1}{2} \sum_{\mu\nu\lambda\sigma} P_{\mu\nu} P_{\lambda\sigma} (\mu \lambda \parallel \nu \sigma) + \sum_{\mu\nu} W_{\mu\nu} S_{\mu\nu}^x + V_{\text{nuc}}^x \]

where,

\[ W_{\mu\nu} = -\sum_{\lambda\sigma} P_{\mu\lambda} F_{\lambda\sigma} P_{\sigma\nu} \]
SCF Algorithms

- **Direct:**
  - $O(N^{2.3})$ CPU
  - Modest memory - 4 MW
  - Faster than conventional
  - Faster than InCore for very large jobs, but can't use InCore for these anyway

- **Forces and Optimizations:**
  - Integral derivatives can be used as computed
  - No new storage issues
  - Energy + Gradient only 20-30% more CPU than energy
**MP2 Energy**

\[ E_{MP2} = E_{HF} + E^{(2)} = E_{HF} + \frac{1}{4} \sum a_{ij}^{ab}(ij \parallel ab) \]

where,

\[ a_{ij}^{ab} = \frac{(ij\parallel ab)}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \]

Sum for \( E^{(2)} \) is \( O(O^2V^2) \), so expensive step is forming \((ij\parallel ab)\).
Traditional MP2 Method

Traditional method: disk-based integral transformation
Conventional MP2 Energy

Setup

Evaluate and Store Integrals

Solve SCF

Transform Integrals

Antisymmetrize, form $E(2)$

I/O → Disk
Direct MP2

- Compute integrals while transforming
- Double integral evaluation permits full vectorization
- No external storage or I/O
- OVN memory minimum
- Do $O^2$VN/Memory integral evaluations, up to $O$ total
Semi-Direct MP2

- Use memory and disk to minimize CPU time
- Sort (ia|\lambda\sigma) into (\lambda\sigma|ia) on disk
- As little as $O(N^2)$ memory and $N^3/2$ disk
- Do $(1/2)OVN^2/\text{MaxDisk}$ integral evaluations
- $OVN^2/2$ disk for one pass
Semi-Direct MP2 Energy

1. Setup
2. Direct SCF
3. Form $(ia|\lambda\sigma)$ for current batch of $i$'s
4. Transpose to $(\lambda\sigma|ia)$
5. Form $<ij||ab>$
6. Update $E(2)$, More $i$'s?
7. Evaluate Integrals
8. Disk I/O

Integrals Evaluation Flowchart
InCore MP2

- Keep AO integrals in main memory
- Need double-length list
- $N^4/4$ memory for closed or open shell
MP2 Gradients

Traditional algorithm:
- Disk for derivatives and energy terms
- I/O time for sorting

Direct algorithm:
- $N^3$ memory for each $i$ in batch
- Size of system limited by memory

Semi-direct algorithm:
- Almost always preferred
- Minimum $O(N^2)$ memory, $N^3/2$ disk
- 6-8 MW for spdf
MP2 Frequencies

- Only semi-direct algorithm
- 8MW for f functions, 12MW for g functions
- MP2=stingy option default for better disk re-use
- MP2=NoStingy uses more disk, is slightly faster
- Minimum disk
  - $N^4/4 + OVN^2/2$ words
  - MaxDisk obeyed
  - Tries calculation in minimum disk regardless
### $C_nH_{n+2}$ Hydrocarbons-MP2 Calculations

<table>
<thead>
<tr>
<th>n</th>
<th>Semi-direct (Sec.)</th>
<th>Fully-direct (Sec.)</th>
<th>Basis Functions</th>
<th>File Sizes (SD)</th>
<th>File Sizes (FD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.7</td>
<td>6.2</td>
<td>23</td>
<td>28</td>
<td>20</td>
</tr>
<tr>
<td>2</td>
<td>5.6</td>
<td>8.7</td>
<td>42</td>
<td>29</td>
<td>20</td>
</tr>
<tr>
<td>3</td>
<td>8.5</td>
<td>15.5</td>
<td>61</td>
<td>42</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>14.9</td>
<td>28.4</td>
<td>80</td>
<td>71</td>
<td>24</td>
</tr>
<tr>
<td>5</td>
<td>51.3</td>
<td>49.2</td>
<td>99</td>
<td>126</td>
<td>24</td>
</tr>
<tr>
<td>6</td>
<td>85.0</td>
<td>80.4</td>
<td>118</td>
<td>221</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>130.9</td>
<td>141.6</td>
<td>137</td>
<td>361</td>
<td>28</td>
</tr>
<tr>
<td>8</td>
<td>194.3</td>
<td>205.9</td>
<td>156</td>
<td>548</td>
<td>32</td>
</tr>
<tr>
<td>9</td>
<td>274.5</td>
<td>324.5</td>
<td>175</td>
<td>779</td>
<td>36</td>
</tr>
<tr>
<td>10</td>
<td>368.0</td>
<td>474.7</td>
<td>194</td>
<td>1072</td>
<td>36</td>
</tr>
</tbody>
</table>

Timings on an IBM WinterHawkII, 375 MHz
Gaussian98 Rev. A10
Semi-direct & Fully Direct memory: 48MB
$C_nH_{n+2}$ Hydrocarbons CPU (SD & FD)

IBM Power3
G98 A.10
$C_nH_{n+2}$
$C_nH_{n+2}$ Hydrocarbons File Sizes-MP2

IBM Power3
G98 A.10
$C_nH_{n+2}$
# MP2 - Frequency

<table>
<thead>
<tr>
<th>n</th>
<th>Single-Point (Sec.)</th>
<th>Frequency (Sec.)</th>
<th>Basis Functions</th>
<th>File Sizes(SP)</th>
<th>File Sizes(F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.7</td>
<td>22.1</td>
<td>23</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>2</td>
<td>5.6</td>
<td>48.0</td>
<td>42</td>
<td>29</td>
<td>53</td>
</tr>
<tr>
<td>3</td>
<td>8.5</td>
<td>169.1</td>
<td>61</td>
<td>42</td>
<td>143</td>
</tr>
<tr>
<td>4</td>
<td>14.9</td>
<td>544.0</td>
<td>80</td>
<td>71</td>
<td>351</td>
</tr>
<tr>
<td>5</td>
<td>51.3</td>
<td>1464.6</td>
<td>99</td>
<td>126</td>
<td>751</td>
</tr>
<tr>
<td>6</td>
<td>85.0</td>
<td>3369.7</td>
<td>118</td>
<td>221</td>
<td>1435</td>
</tr>
</tbody>
</table>
MP2 Frequency & SP CPU Comparison

IBM Power3
G98 A.10
CₙHₙ₊₂
MP2 Freq & SP Disk Usage Comparison

File Size (MB)

Single-Point

Frequency

IBM Power3
G98 A.10
C_{n}H_{n+2}
Integral Transformation

Traditionally used for everything after SCF:

$$(pq \mid rs) = \sum_{\sigma} C_{\sigma s} \sum_{\lambda} C_{\lambda r} \sum_{v} C_{v q} \sum_{\mu} C_{\mu p}(\mu v \mid \lambda \sigma)$$

Gaussian uses semi-direct algorithm:
- Fixed minimum memory for integral evaluation
- Better behavior for large systems and limited memory
- Generate $<pq||rs>$ during transformation
- Can make $<ij||ab>$ using only $O(O^2N^2)$ disk
MAXDISK

Specifies the amount of disk storage available for scratch data, in 8-bytes words

The units can be: KB, MB, GB, KW, MW or GW

example:
maxdisk=8MB
## Size Dependence of Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Formal CPU</th>
<th>Formal Memory</th>
<th>Formal Disk</th>
<th>Actual CPU</th>
<th>Actual Disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv. SCF</td>
<td>$N^4$</td>
<td>$N^2$</td>
<td>$N^4$</td>
<td>$N^{3.5}$</td>
<td>$N^{3.5}$</td>
</tr>
<tr>
<td>Incore SCF</td>
<td>$N^4$</td>
<td>$N^4$</td>
<td>-</td>
<td>$N^4$</td>
<td>$N^2$</td>
</tr>
<tr>
<td>Direct SCF</td>
<td>$N^4$</td>
<td>$N^2$</td>
<td>-</td>
<td>$N^{2.3}$</td>
<td>$N^2$</td>
</tr>
<tr>
<td>Conv. MP2</td>
<td>$ON^4$</td>
<td>$N^2$</td>
<td>$N^4$</td>
<td>$ON^4$</td>
<td>$N^4$</td>
</tr>
<tr>
<td>Dir MP2 SP</td>
<td>$ON^4$</td>
<td>OVN</td>
<td>-</td>
<td>$O^2N^3$</td>
<td>$N^2$</td>
</tr>
<tr>
<td>SD MP2 SP</td>
<td>$ON^4$</td>
<td>$N^2$</td>
<td>$VN^2$</td>
<td>$O^2N^3$</td>
<td>$VN^2$</td>
</tr>
<tr>
<td>Conv. MP2 Force</td>
<td>$ON^4$</td>
<td>$N^2$</td>
<td>$N^4$</td>
<td>$ON^4$</td>
<td>$N^4$</td>
</tr>
<tr>
<td>Dir MP2 Force</td>
<td>$ON^4$</td>
<td>$N^3$</td>
<td>-</td>
<td>$O^2N^3$</td>
<td>$N^2$</td>
</tr>
<tr>
<td>SD MP2 Force</td>
<td>$ON^4$</td>
<td>$N^2$</td>
<td>$N^3$</td>
<td>$O^2N^3$</td>
<td>$N^3$</td>
</tr>
<tr>
<td>MP3, CISD, QCISD</td>
<td>$O^2N^4$</td>
<td>$N^2$</td>
<td>$N^4$</td>
<td>$O^2N^4$</td>
<td>$N^4$</td>
</tr>
<tr>
<td>MP4, QCISD(T)</td>
<td>$O^3V^4$</td>
<td>$N^2$</td>
<td>$N^4$</td>
<td>$O^3V^4$</td>
<td>$N^4$</td>
</tr>
</tbody>
</table>

$O$: Number of occupied orbitals  
$V$: Number of virtual orbitals  
$N$: Number of basis functions
Parallel Gaussian

Efficiency Considerations
Amdahl's Law

Although a code contains parallel constructs, the serial processing in the code will dominate its overall performance.

To estimate expected parallel speedups:

$$S(N) = \frac{1}{f_S + \frac{f_P}{N}}$$

- $S(N)$: Maximum expected speedup from parallelization
- $N$: Number of processors available for parallel execution
- $f_P$: Fraction of a program that can execute in parallel
- $f_S$: Fraction of a program that is serial = $1 - f_P$
Amdahl's Law Example

![Graph showing Amdahl's Law example]

- **X-axis:** Number of Processors
- **Y-axis:** Parallel Fraction

The graph illustrates the impact of adding processors to a system, showing how the parallel fraction decreases as the number of processors increases. The graph highlights the diminishing return on performance improvement as more processors are added.
Estimating Memory Requirements

**Single processor memory requirement** = \( M + 2N^2 \)

- \( M = \) Required value for a job type
- \( N = \) Number of Basis Functions

<table>
<thead>
<tr>
<th>SCF Energies</th>
<th>f functions</th>
<th>g functions</th>
<th>h functions</th>
<th>i functions</th>
<th>j functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>~6 MW</td>
<td>~6 MW</td>
<td>~12 MW</td>
<td>~25 MW</td>
<td>~64 MW</td>
<td></td>
</tr>
<tr>
<td>SCF Gradients</td>
<td>~6 MW</td>
<td>~7 MW</td>
<td>~19 MW</td>
<td>~40 MW</td>
<td></td>
</tr>
<tr>
<td>SCF Frequencies</td>
<td>~6 MW</td>
<td>~11 MW</td>
<td>~30 MW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP2 Energies</td>
<td>~6 MW</td>
<td>~7 MW</td>
<td>~14 MW</td>
<td>~30 MW</td>
<td>~74 MW</td>
</tr>
<tr>
<td>MP2 Gradients</td>
<td>~6 MW</td>
<td>~8 MW</td>
<td>~18 MW</td>
<td>~40 MW</td>
<td></td>
</tr>
<tr>
<td>MP2 Frequencies</td>
<td>~8 MW</td>
<td>~12 MW</td>
<td>~30 MW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 MW = 1,048,576 Words = 8,388,608 bytes

Example: 300 basis functions HF geometry optimization using g functions would require about 7.2 MW ( ~60MB)
FreqMem Utility

- FreqMem utility:
  - Returns minimum memory size for optimal performance

- Example:
  - `freqmem N_A N R/U C/D SP/SPD/SPDF`

  - $N_A$ = number of atoms
  - $N$ = number of basis functions
  - R/U = restricted/unrestricted
  - C/D = conventional/direct
  - SP/SPD/SPDF = functions in basis set
Memory Allocation Empirical Formula

Parallel calculations with more than one processor on shared-memory systems require additional memory.

\[ \text{total_mem} = \text{sp_mem} + (n - 1) \times 0.75 \times \text{sp_mem} \]

**total_mem** = total memory required for the parallel run
**sp_mem** = single processor memory required
**n** = number of processors
Parallel SCF

*Deck PRSMsu

subroutine PRSMsu

\[ \text{loop over } N\text{processors} \]

call PRISM

\[ \text{end loop} \]

\[ \text{loop over } N\text{processors ( serial code )} \]

add $1/N\text{processors}$ Fock Matrix contributions

\[ \text{end loop} \]
Parallel Speedup & Efficiency

Speedup (S) is defined as the ratio of the serial run time (elapsed, $t_s$) over the time that it takes to do the same problem in parallel (elapsed time, $t_p$)

$$S = \frac{t_s}{t_p}$$

$$e = \frac{S}{N_{processors}}$$
Extrapolated Speedup

\[ S = \frac{1}{\left(\frac{p}{N_{\text{processors}}}\right) + (1-p)} \]

\[ p = \frac{S_{N_{\text{processors}}} - S_{M_{\text{processors}}}}{(1-1/N_{\text{processors}}) \times S_{N_{\text{processors}}} - (1-1/M_{\text{processors}}) \times S_{M_{\text{processors}}}} \]
# Parallel Links in Gaussian98

<table>
<thead>
<tr>
<th>Link</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L302</td>
<td>Overlap integrals</td>
</tr>
<tr>
<td>L303</td>
<td>One-electron properties integrals</td>
</tr>
<tr>
<td>L502</td>
<td>Closed-and open-shell SCF solution</td>
</tr>
<tr>
<td>L506</td>
<td>GVB solution</td>
</tr>
<tr>
<td>L508</td>
<td>Quadratically convergent SCF solution</td>
</tr>
<tr>
<td>L510</td>
<td>Multiconfiguration SCF solution</td>
</tr>
<tr>
<td>L602</td>
<td>One-electron properties</td>
</tr>
<tr>
<td>L703</td>
<td>Two-electron integral first or second derivative evaluation</td>
</tr>
<tr>
<td>L906</td>
<td>Direct and semi-direct MP2 energies and gradients</td>
</tr>
<tr>
<td>L914</td>
<td>Calculates excited states using CI with single excitations</td>
</tr>
<tr>
<td>L1002</td>
<td>CPHF solution and contribution of coefficient derivatives to Hartree-Fock second derivatives</td>
</tr>
<tr>
<td>L1014</td>
<td>Coupled perturbed CI singles</td>
</tr>
<tr>
<td>L1110</td>
<td>Two-electron contributions to Fock matrix derivatives with respect to nuclear coordinates</td>
</tr>
<tr>
<td>L1112</td>
<td>Forms most of the terms in MP2 second derivatives</td>
</tr>
</tbody>
</table>

Linda links
# Crown ether Example

| Processors       | Elapsed Time (Sec)
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4549</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ethernet¹</td>
<td>2271</td>
</tr>
<tr>
<td>switch¹</td>
<td>2268</td>
</tr>
<tr>
<td>shared-memory²</td>
<td>2365</td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>ethernet¹</td>
<td>652</td>
</tr>
<tr>
<td>switch¹</td>
<td>610</td>
</tr>
<tr>
<td>shared-memory²</td>
<td>626</td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>ethernet¹</td>
<td>442</td>
</tr>
<tr>
<td>switch¹</td>
<td>372</td>
</tr>
<tr>
<td>shared-memory²</td>
<td>386</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>12</td>
</tr>
</tbody>
</table>

¹ 16X(4-way nodes), Power3-II, 375 MHz, 8MB L2

² 1X(16-way node), Power3-II, 375 MHz, 8MB L2

³ Gaussian98 Rev. A.7, xlf 5.1.1 Compiler
Crown ether Parallel Speedup

(OCH₂)$_7$, Crown ether
HF/6-31G* FOPT OPTCYC
test178

test178: RHF/6-31G** SCF=DIRECT POP=NPA PROP=FIT
300 Basis Functions
Full Point Group D_{3h}

<table>
<thead>
<tr>
<th>Processor</th>
<th>Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>153.39</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>109.86</td>
<td>1.39</td>
</tr>
<tr>
<td>4</td>
<td>82.90</td>
<td>1.85</td>
</tr>
<tr>
<td>8</td>
<td>90.01</td>
<td>1.70</td>
</tr>
<tr>
<td>16</td>
<td>84.48</td>
<td>1.82</td>
</tr>
</tbody>
</table>

Gaussian 98 Rev. A.7
Shared-memory
Symmetry reduces the total number of integrals

test178: RHF/6-31G** SCF=DIRECT POP=NPA PROP=FIT
300 Basis Functions
Full Point Group $D_{3h}$
## α-pinene SP Scalability

<table>
<thead>
<tr>
<th>Processors</th>
<th>Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HF</td>
<td>2880.54</td>
<td>1.00</td>
</tr>
<tr>
<td>B3-LYP</td>
<td>4022.71</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HF</td>
<td>1463.07</td>
<td>1.97</td>
</tr>
<tr>
<td>B3-LYP</td>
<td>2036.24</td>
<td>1.98</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HF</td>
<td>726.13</td>
<td>3.97</td>
</tr>
<tr>
<td>B3-LYP</td>
<td>1031.06</td>
<td>3.90</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HF</td>
<td>360.90</td>
<td>7.98</td>
</tr>
<tr>
<td>B3-LYP</td>
<td>515.95</td>
<td>7.80</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HF</td>
<td>194.39</td>
<td>14.82</td>
</tr>
<tr>
<td>B3-LYP</td>
<td>285.37</td>
<td>14.10</td>
</tr>
</tbody>
</table>

α-pinene HF/6-311G(df,p) & B3-LYP/6-31G(df,p)  
346 Basis Functions  
$C_{10}H_{16}$  
Distributed-memory
α-pinene: Hf & DFT Scalability

HF/6-31G(df,p) & B3-LYP/6-31G(df,p)
346 Basis Functions
C$_{10}$H$_{16}$
Distributed-memory
## \(\alpha\)-pinene Frequency Calculation

B3-LYP/6-31G* FREQ
182 Basis Functions
G98 Rev. A.7
shared-memory

![Chemical Structure of \(\alpha\)-pinene](image)

<table>
<thead>
<tr>
<th>Processors</th>
<th>L502</th>
<th>S(^a)</th>
<th>L1110</th>
<th>S(^a)</th>
<th>L1002</th>
<th>S(^a)</th>
<th>L703</th>
<th>S(^a)</th>
<th>Total</th>
<th>S(^a)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1076</td>
<td>1.0</td>
<td>2802</td>
<td>1.0</td>
<td>3144</td>
<td>1.0</td>
<td>3702</td>
<td>1.0</td>
<td>10738</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>515</td>
<td>2.0</td>
<td>1402</td>
<td>2.0</td>
<td>1602</td>
<td>2.0</td>
<td>1827</td>
<td>2.0</td>
<td>5362</td>
<td>2.0</td>
</tr>
<tr>
<td>4</td>
<td>254</td>
<td>4.0</td>
<td>700</td>
<td>4.0</td>
<td>879</td>
<td>4.0</td>
<td>911</td>
<td>4.0</td>
<td>2764</td>
<td>3.9</td>
</tr>
<tr>
<td>8</td>
<td>136</td>
<td>7.9</td>
<td>359</td>
<td>7.8</td>
<td>580</td>
<td>7.9</td>
<td>471</td>
<td>7.9</td>
<td>1576</td>
<td>6.8</td>
</tr>
<tr>
<td>16</td>
<td>78</td>
<td>13.8</td>
<td>187</td>
<td>15.0</td>
<td>437</td>
<td>14.8</td>
<td>251</td>
<td>14.8</td>
<td>998</td>
<td>10.1</td>
</tr>
</tbody>
</table>

\(^a\) Speedup
\( \alpha \)-pinene Speedups

![Graph showing speedup with number of processors]

- L502
- L1110
- L1002
- L703
- Total

B3-LYP/6-31G* Frequency
182 Basis Functions
**CIS Calculation**

**CIS=direct, 6-31++G, scf=direct, force**

154 Basis Functions

**Distributed-memory**

**G98 Rev. A.7**

<table>
<thead>
<tr>
<th>Processors</th>
<th>L502</th>
<th>S&lt;sup&gt;a&lt;/sup&gt;</th>
<th>L914</th>
<th>S&lt;sup&gt;a&lt;/sup&gt;</th>
<th>L1002</th>
<th>S&lt;sup&gt;a&lt;/sup&gt;</th>
<th>L703</th>
<th>S&lt;sup&gt;a&lt;/sup&gt;</th>
<th>Total</th>
<th>S&lt;sup&gt;a&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>821</td>
<td>1.00</td>
<td>1455</td>
<td>1.00</td>
<td>701</td>
<td>1.00</td>
<td>193</td>
<td>1.00</td>
<td>3182</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>441</td>
<td>1.86</td>
<td>776</td>
<td>1.88</td>
<td>377</td>
<td>1.86</td>
<td>97</td>
<td>1.99</td>
<td>1703</td>
<td>1.87</td>
</tr>
<tr>
<td>4</td>
<td>234</td>
<td>3.51</td>
<td>411</td>
<td>3.54</td>
<td>199</td>
<td>3.52</td>
<td>50</td>
<td>3.86</td>
<td>906</td>
<td>3.51</td>
</tr>
<tr>
<td>8&lt;sup&gt;b&lt;/sup&gt;</td>
<td>112</td>
<td>7.33</td>
<td>212</td>
<td>6.86</td>
<td>98</td>
<td>7.15</td>
<td>29</td>
<td>6.66</td>
<td>480</td>
<td>6.63</td>
</tr>
<tr>
<td>16</td>
<td>69</td>
<td>11.90</td>
<td>129</td>
<td>11.28</td>
<td>63</td>
<td>11.1</td>
<td>15</td>
<td>12.87</td>
<td>292</td>
<td>10.90</td>
</tr>
</tbody>
</table>

<sup>a</sup> Speedup

<sup>b</sup> Shared-memory

![Chemical structure](image)
CIS Scalability

Speedup

Number of Processors

CIS = direct, 6-31++G, scf = direct force
154 Basis Functions
Distributed-memory
G98 Rev. A.7
**MCSCF Calculation**

*CAS(6,6), 6-31+G(3df), guess=cards, NOSYM*

240 Basis Functions

Shared-memory

G98 Rev. A.7

<table>
<thead>
<tr>
<th>Processors</th>
<th>L510</th>
<th>( S^a )</th>
<th>Total</th>
<th>( S^a )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>476</td>
<td>1.00</td>
<td>483</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>238</td>
<td>2.00</td>
<td>246</td>
<td>1.96</td>
</tr>
<tr>
<td>4</td>
<td>123</td>
<td>3.87</td>
<td>134</td>
<td>3.60</td>
</tr>
<tr>
<td>8</td>
<td>67</td>
<td>7.10</td>
<td>83</td>
<td>5.82</td>
</tr>
<tr>
<td>16</td>
<td>40</td>
<td>11.90</td>
<td>65</td>
<td>7.43</td>
</tr>
</tbody>
</table>
MCSCF Scalability

![Graph showing MCSCF Scalability](image)

- Speedup vs. Number of Processors
  - L510
  - Total

CAS(6,6), 6-31+G(3df), guess=cards, NOSYM
240 Basis Functions
Shared-memory
G98 Rev. A.7
Summary (for an n-way system)

HF: SP, Gradients, Freq
DFT: SP, Gradients, Freq
CIS: SP, Gradients, Freq
MCSCF: SP and Gradients
MP2: SP, Gradients
MP3 & MP4: SP
MP4SDTQ: SP
QCI & CC: SP

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Shared-Memory</th>
<th>Distributed-Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>HF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MCSCF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP3 &amp; MP4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP4SDTQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QCI &amp; CC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Efficiency: 0 0.2 0.4 0.6 0.8 1
Information

Gaussian official site:
http://www.gaussian.com

Institute-IBM Gaussian site:
http://www.msi.umn.edu/user_support/compchem/ gaussian_tech/

contact:
help@msi.umn.edu