Gaussian98 Performance Guide on a Heterogeneous Environment

Carlos P. Sosa
Cray Inc
and
Michael Frisch
Gaussian, Inc.

Agenda

- Motivation
- *Gaussian* architecture
- Parallel versions
- *Gaussian* Usage
- Cray SV1 Architecture
- Performance
- Summary
- Future Work

Motivation

- *Gaussian* takes into account resources available at run time and tries to choose the best algorithm for a particular platform
- *Gaussian* cannot choose the number of processors for any type of input nor it can select a machine in a heterogeneous environment

Gaussian Architecture

- *Gaussian 98* is a connected system of programs for performing a variety of semiempirical, density functional theory and *ab initio* calculations
- It consists of more than 4000 subroutines organized into programs which communicate through disk files. Each subprogram is a link. Links are organized in overlays.
  - Overlay 0 is responsible to start the program, including reading the input file.
  - Once the route card is read, the proper set of overlays/options/links is selected for a particular run.
  - Overlay 99 (L9999) terminates the run.
- Currently one source code supports all versions

Shared and Distributed Memory Parallel Versions

- *Gaussian* is parallelized either through shared memory or Linda
- Parallel processing in *Gaussian* is transparent to the user through subroutines that implement parallelism

Link 0 Prallel Command

%nproc = *N*

Requests that the job use up to *N* processors. On parallel machines, the numbers of processors to use in production can be set in the <u>Default.Route</u> file. If %Nproc nor the <u>Dafault.Route</u> file is used, *Gaussian* will use only 1 processor

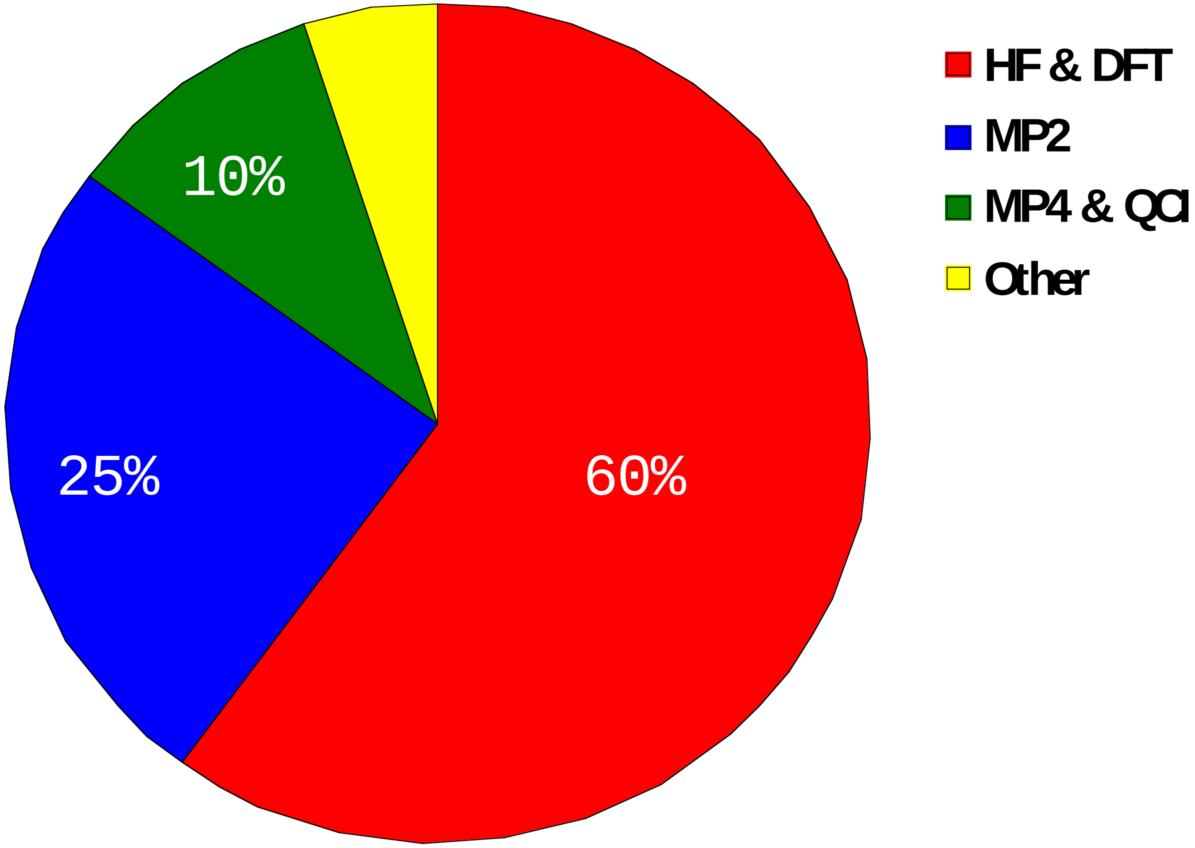
Shared Memory Parallel Input

```
%mem=16MW
%nproc=2
#p hf/6-311++G**

H2O test

0 1
o
h 1 r
h 1 2 2 a

r = 0.98
a = 109.
```

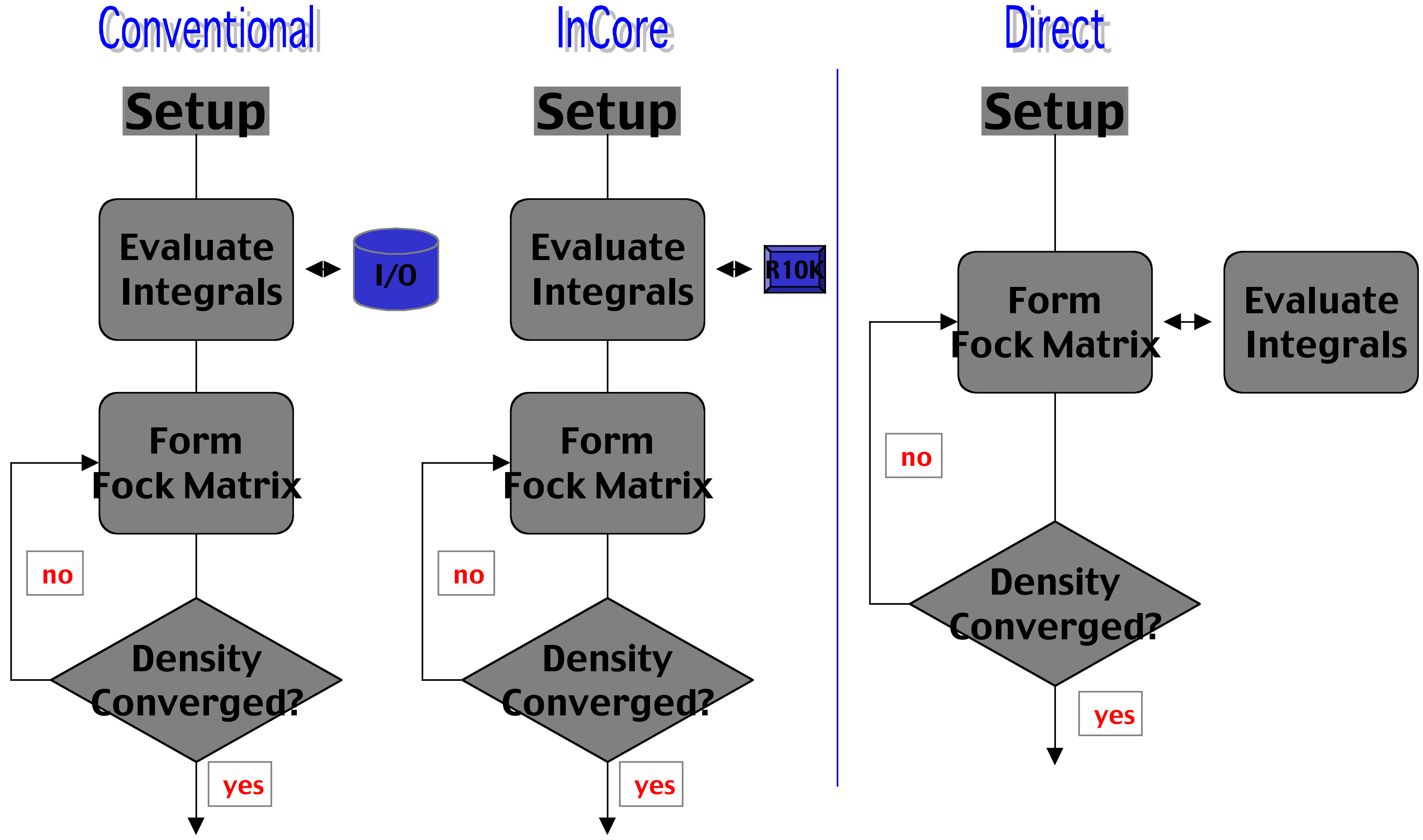
Distributed Memory Parallel Input

```
%mem=16MW
%nproclinda=2
#p hf/6-311++G**

H2O test

0 1
o
h 1 r
h 1 2 2 a

r = 0.98
a = 109.
```

Gaussian Usage



Algorithm


Conventional
Setup
Evaluate Integrals
I/O
Form Fock Matrix
no
Density Converged?
yes
InCore
Setup
Evaluate Integrals
R10K
Form Fock Matrix
no
Density Converged?
yes
Direct
Setup
Form Fock Matrix
Evaluate Integrals
no
Density Converged?
yes

System Resources

Conventional

- $O(N^{3.5})$CPU and Disk
- limited by disk capacity
- I/O bound

- integrals computed once

InCore

- $O(N^{3.5})$CPU
- $N^4/8$ – $N^4/4$ memory
- single CPU
- memory needed
  - 100 basis = 100 MB
  - 200 basis = 200 MB
  - 300 basis = 8100MB

- integrals computed once
- fast

Direct

- $O(N^{2.3})$CPU
- integrals computed
- many times

- modest memory
- large systems

Parallelization of two-electron Integrals

```
loop over $N_{proc}$
  loop over total angular momentum
    loop over degrees of bra and ket contraction
      do integrals for $1/N_{proc}$ of shell quartets
    end loop
  end loop
  add integral contributions to partial Fock matrix
end loop
loop over $N_{proc}$ (serial code)
  add $1/N_{proc}$ Fock matrix contributions
endloop
```

Parallelized Links

Link	Description
502	Closed and open shell SCF solution
506	GVB solution
508	Quadratically convergent SCF solution
510	MCSCF solution
602	One-electron properties
703	Two-electron integrals and derivatives
906	MP2 energies and derivatives
913	CI, CC, QCI, MP3 and MP4
914	Excited states
1002	CPHF solution
1014	Coupled-Perturbed CI singles
1110	Two-electron contributions to Fock derivatives
1112	Forms most terms in MP2 2nd derivatives

Turner, Trucks, and Frisch, ACS Symposium Series 592, 1995
Sosa, Ochterski, Carpenter, and Frisch, J. Comp. Chem. 19, 1053(1998)
Sosa, Scalmani, Gomperts, and Frisch, Theochem., Parallel Comp. 2000

CRAY SV1 Architectural Features

Processor:
300 MHz, 32 CPUs
Cache:
256 KB, 4-way set associative
Memory Size:
4GW
Mainframe:
SN3202, ICE

Memory Allocations for Parallel Runs

Total:
mem = (mem_1_cpu) + (nproc – 1) * (mem_1_cpu) * 3/4

For each additional CPU:
mem_add = 0.75 * (mem_1_cpu)

mem_1_cpu: *Memory required for single CPU run*

Performance

Speedup: $S = t_s/t_p$

Efficiency: $e = S/N_{proc}$

Extrapolated speedup: $s = 1/[(p/N_{proc}) + (1-p)]$

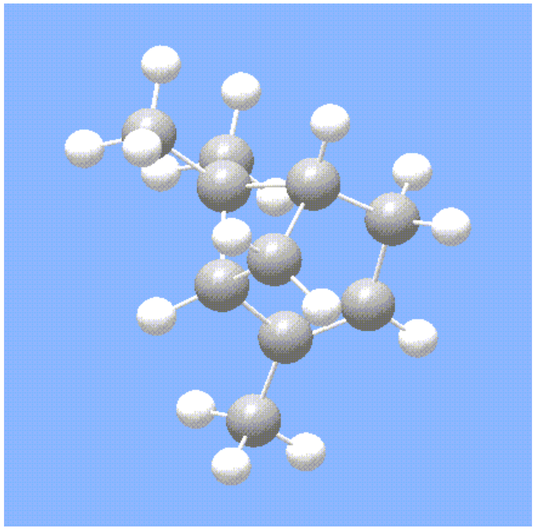
Percentage of parallel code: $p = a / b$

$a = S_{Nproc} - S_{mproc}$

$b = (1 - 1/N_{proc})xS_{Nproc} - (1 - 1/M_{proc})xS_{Mproc}$

Single CPU performance as a Function of Basis Sets

Basis Sets	Mflop/s
STO-3G	81.73
6-311G	129.09
6-311G(d)	133.07
6-311G(d,p)	131.91
6-311+G(d,p)	136.11
6-311++G(d,p)	129.35
6-311++G(2d,2p)	123.25



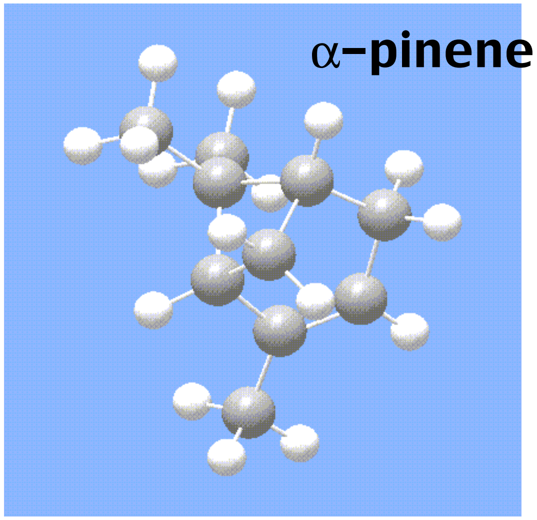
α-pinene

B3-LYP

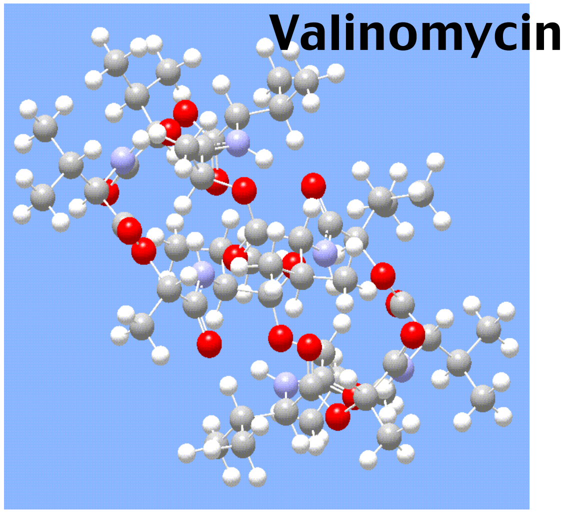
Single Point Energy

$C_{10}H_{16}$

Single CPU performance as a Function of the Number of Atoms



B3-LYP/STO-3G

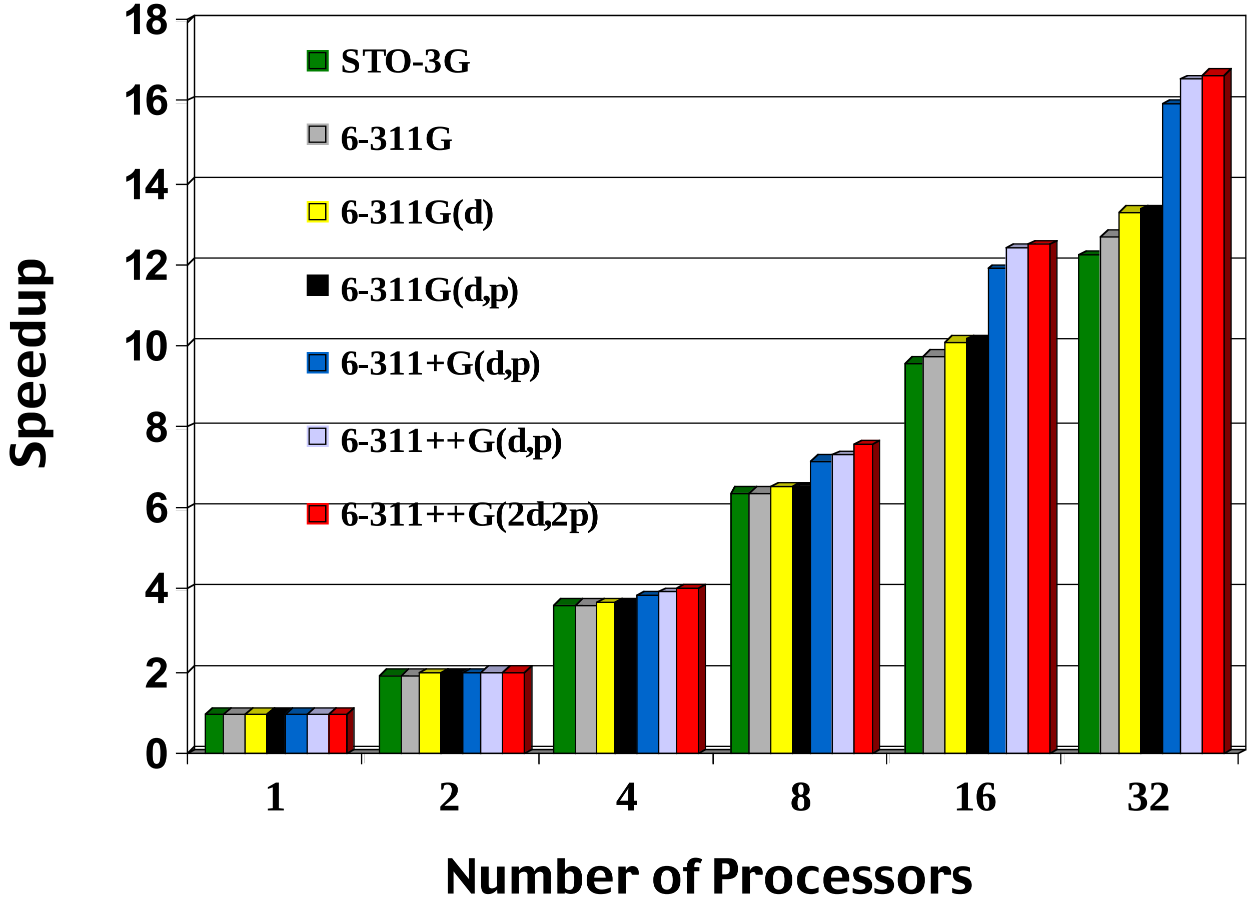


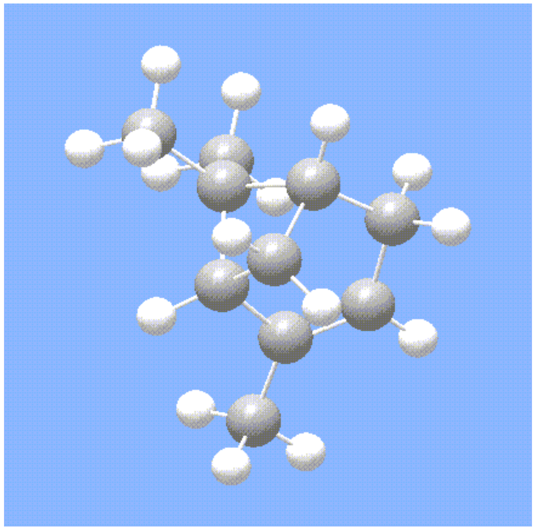
81.75 Mflop/s

100.06 Mflop/s

α–pinene: Scalability as a Function of Basis Sets

Link 502

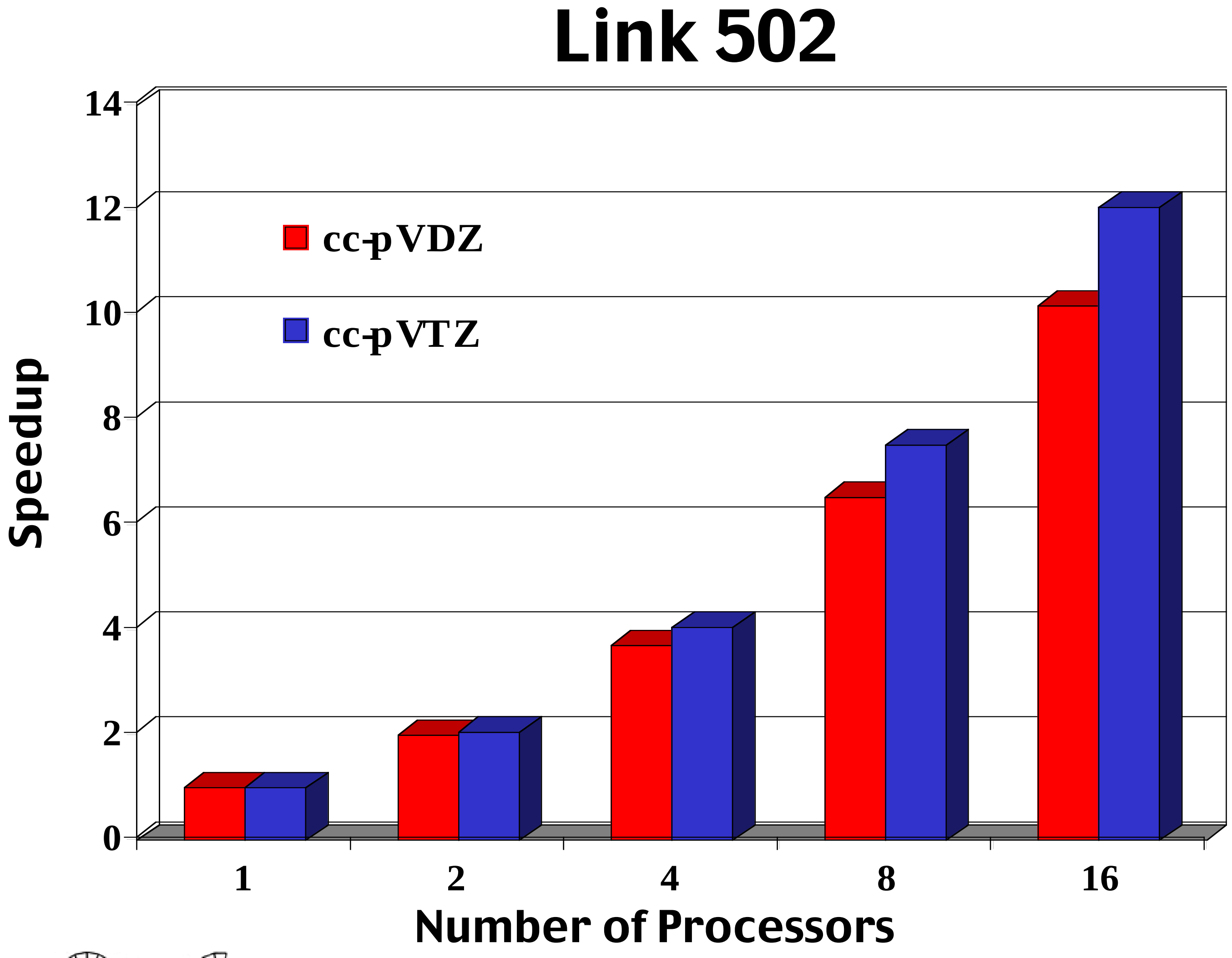


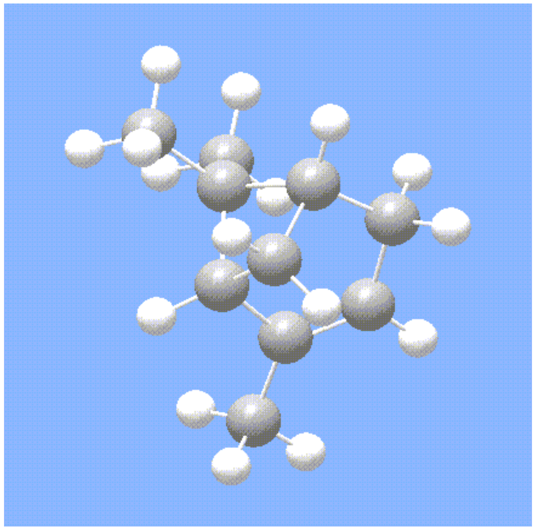


α–pinene
B3–LYP
Single Point Energy
$C_{10}H_{16}$

CUG SV1 Workshop

α–pinene: Scalability as a Function of Dunning’s Basis Sets

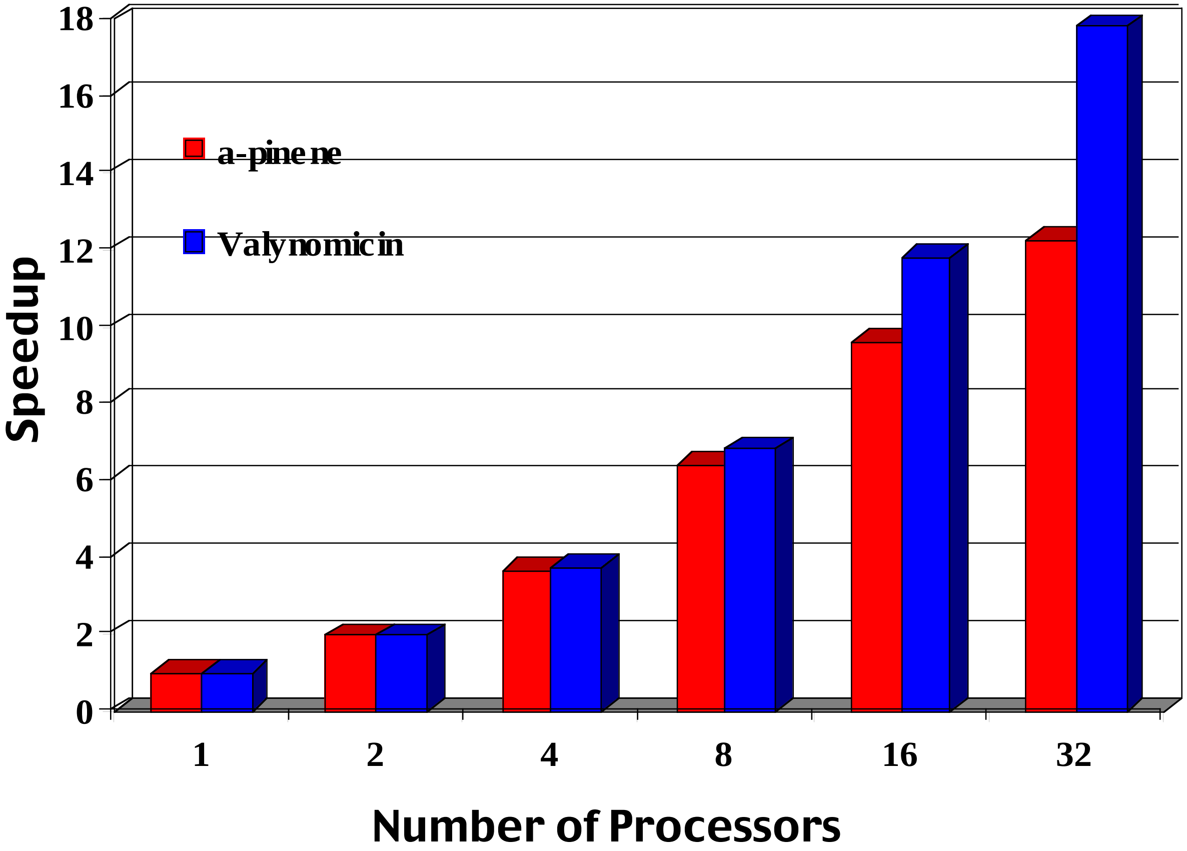


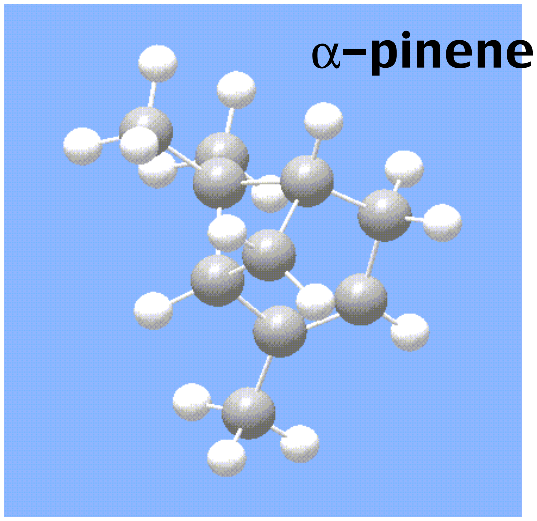


α–pinene
B3–LYP
Single Point Energy
$C_{10}H_{16}$

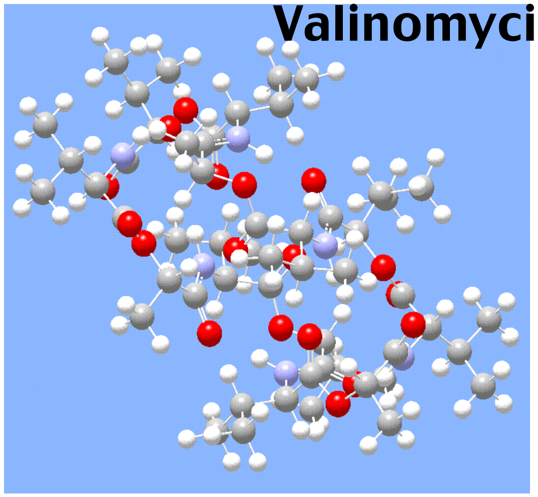
Scalability as a Function of the Size of the System

Link 502

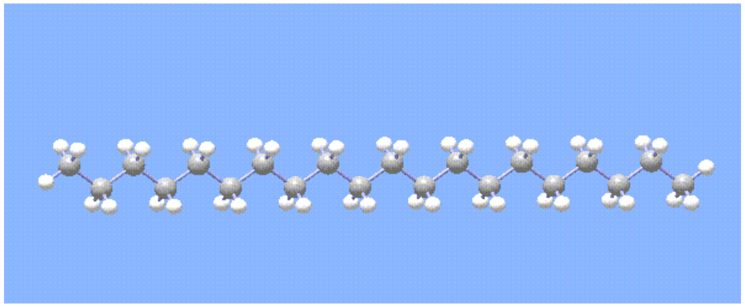


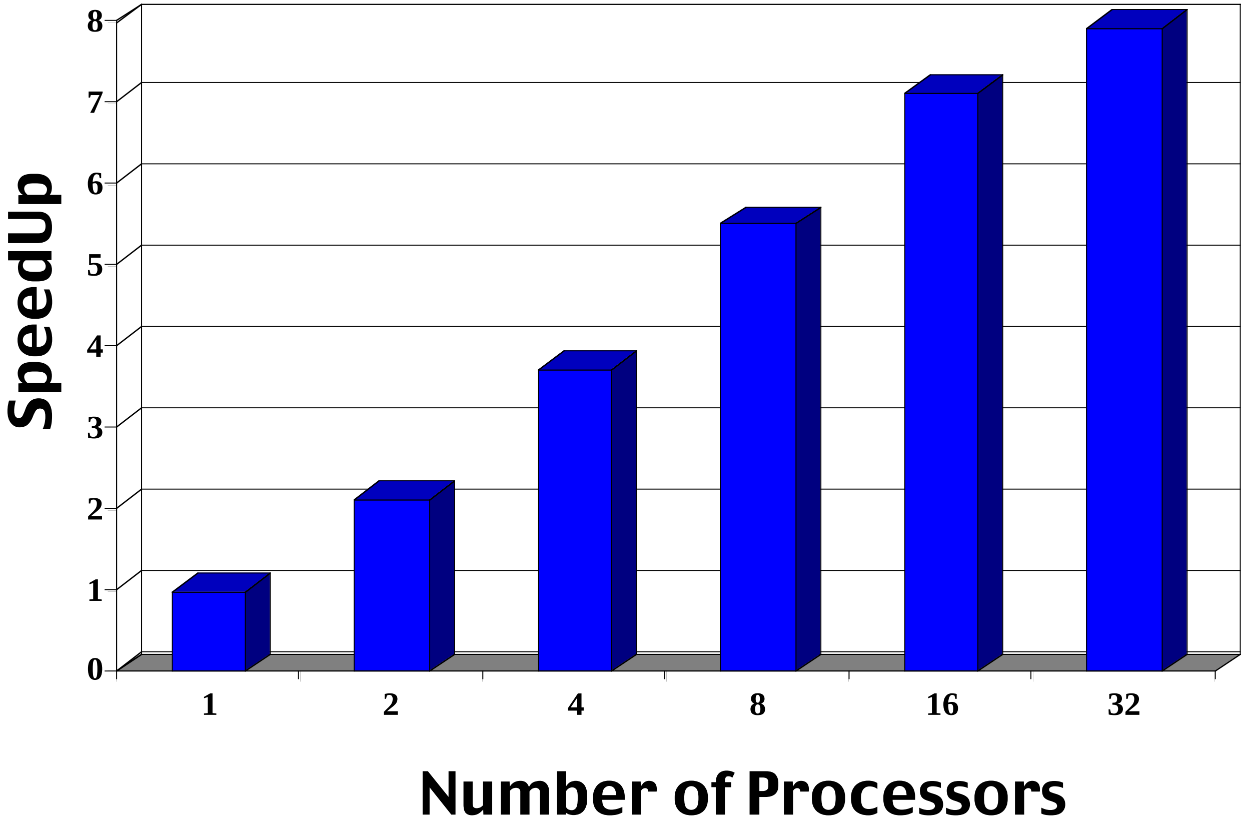

α−pinene

B3−LYP SP


Valinomycin

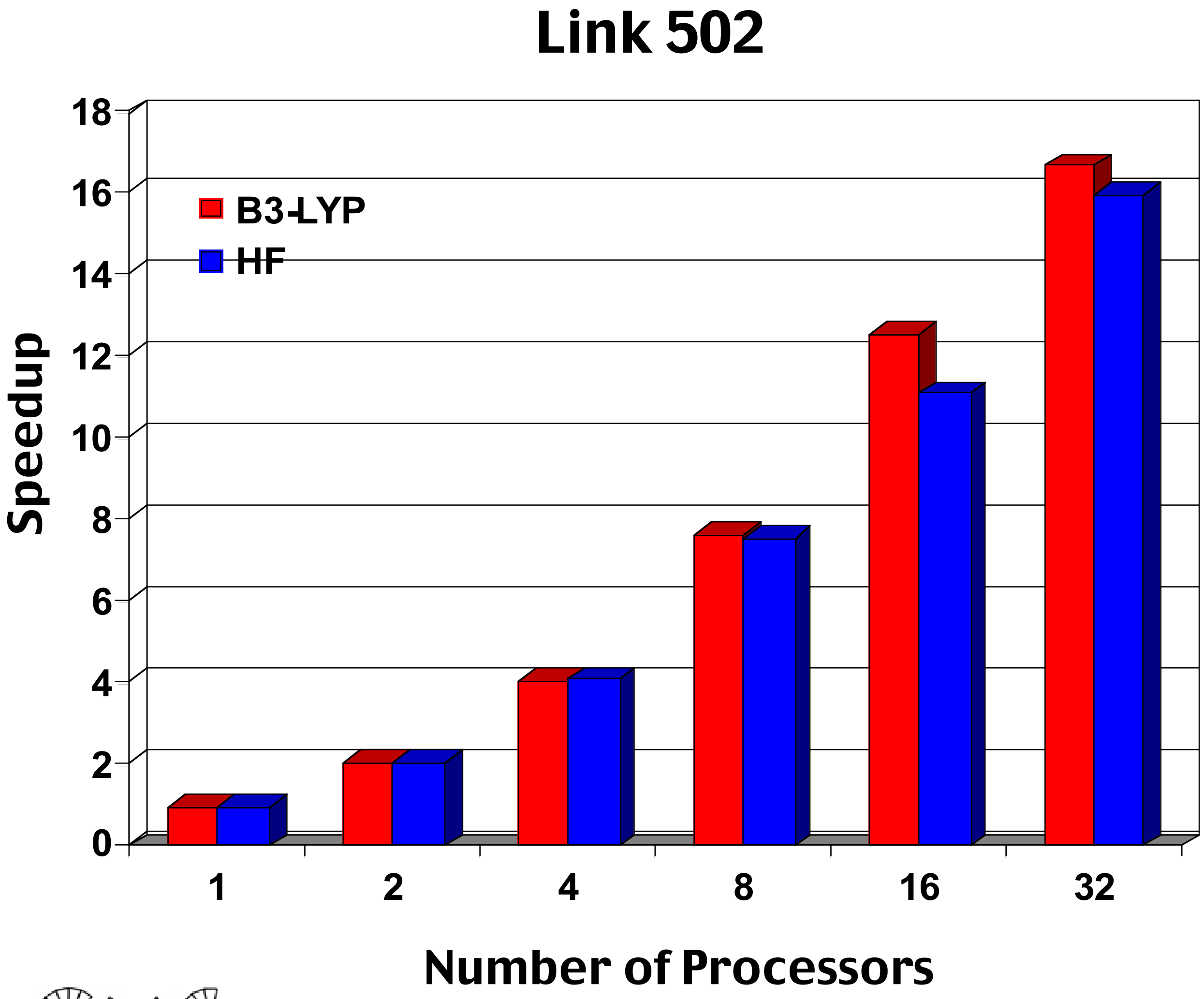
$C_{20}H_{42}$: Scalability as a Function of Molecular Symmetry

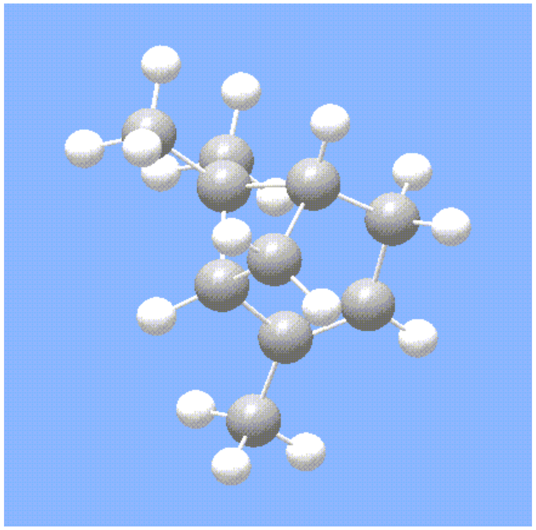




B3-LYP/6-311G(d,p)
612 Basis Functions
Point Group C2H

Differences in Scalability Between HF and B3-LYP





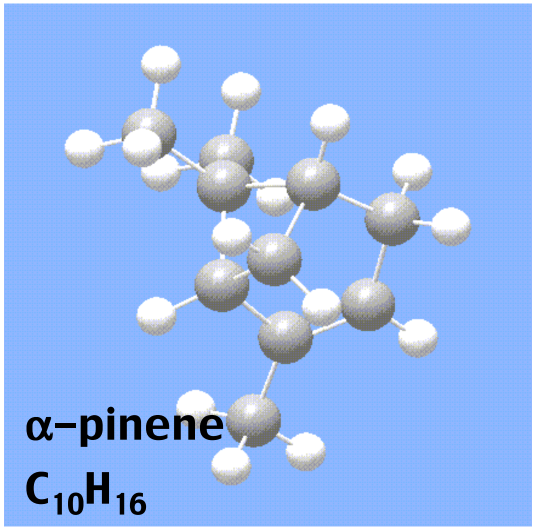
α-pinene

6-311++G(2d,2p) scf=direct

$C_{10}H_{16}$

Total versus Individual Links Speedups

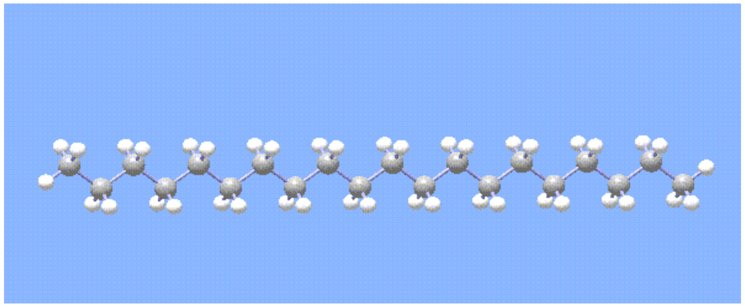
G98->
l1->l101->l202->l301-l302->l303->l401->l502->l601->
l9999



α-pinene
$C_{10}H_{16}$

Case	SpeedUp					
Ideal	1	2	4	8	16	32
STO-3G[L502]	1	1.96	3.65	6.42	9.64	12.27
STO-3G[total]	1	1.88	3.28	5.04	5.88	6.13
6-311++G(2d,2p)[L502]	1	2.04	4.04	7.62	12.54	16.70
6-311++G(2d,2p)[total]	1	2.03	4.00	7.45	12.06	15.71

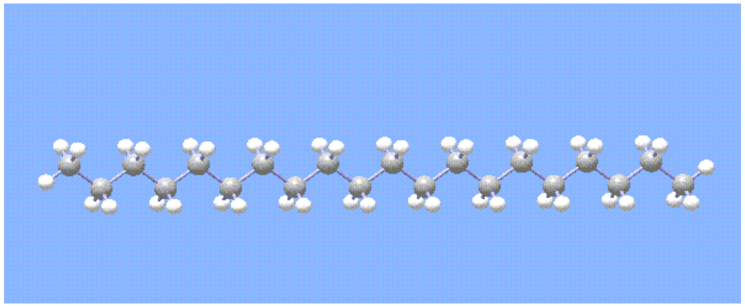
$C_{20}H_{42}$: MP2 Single CPU Performance



Point Group C2H

Basis Set	Mem.	Algorithm	Mem. Usage	MFlops
STO-3G	8 MW	Disk Based	OVN mem.	122.95
	64 MW	Fully Direct	OVN mem.	143.15
	300 MW	Incore		355.67

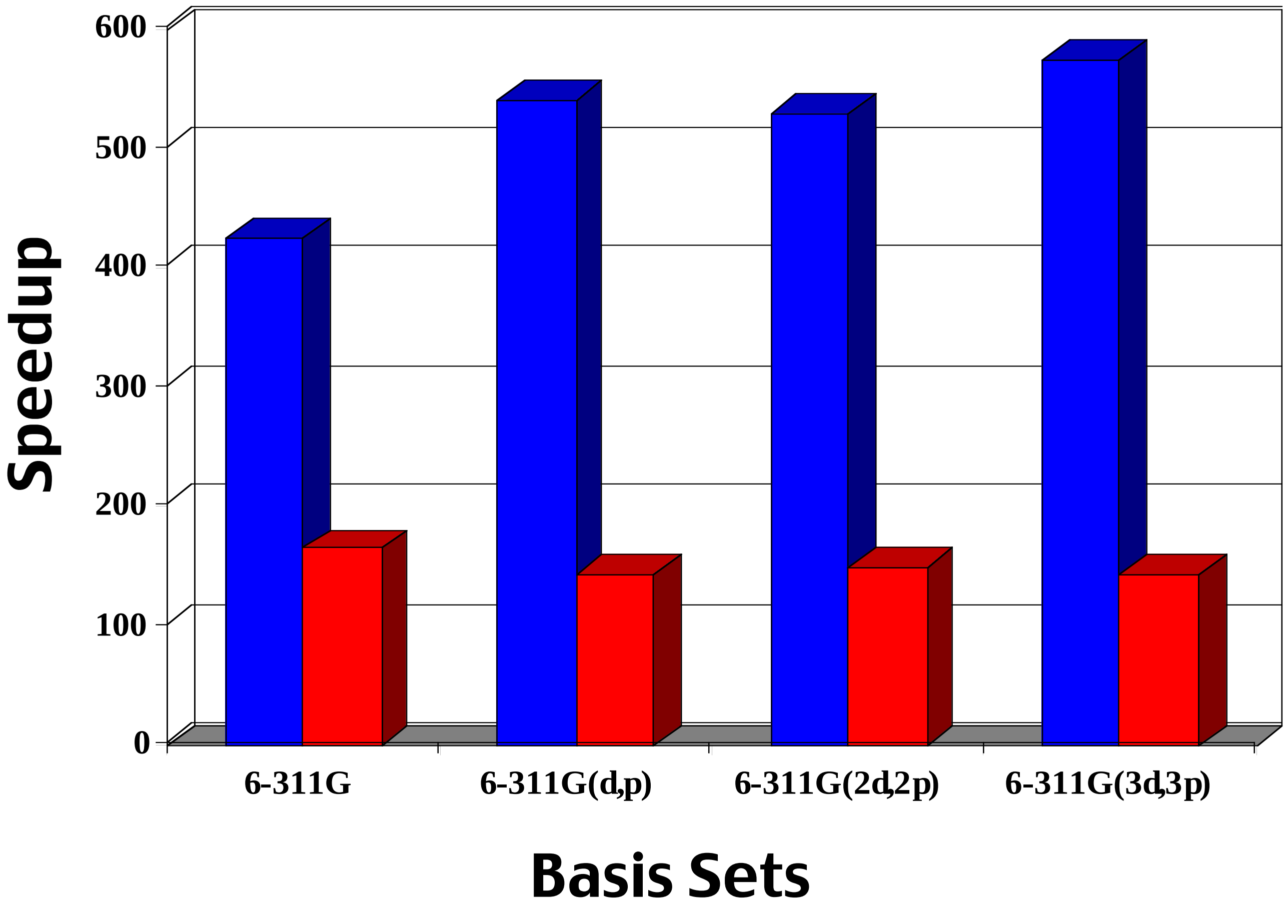
MP2 as a Function of Basis Sets

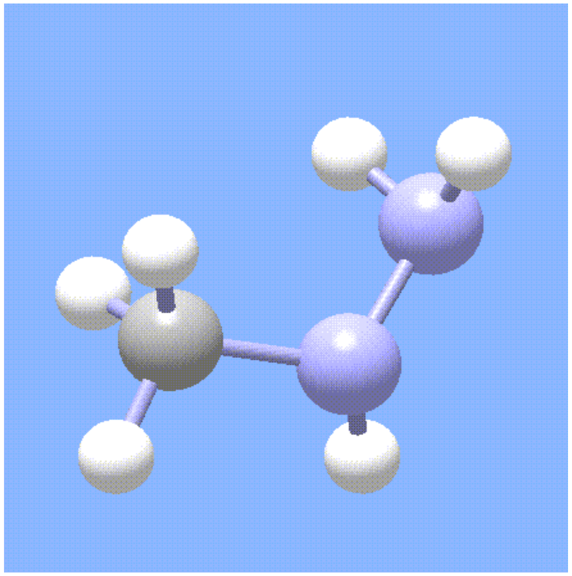


Point Group C2H

Basis Sets	Mflop/s	Mem. Used	Mbytes	B.F.
STO-3G	122.95	OVN,61 occ.	209	142
6-311G	112.14	OVN,61 occ.	4459	386
6-311G(d)	205.66	ON^2,31 occ.	5363	486
6-311G(d,p)	192.98	ON^2,31 occ.	9025	612
6-311+G(d,p)	157.86	ON^2,31 occ	15730	692
6-311++G(d,p)	148.28	ON^2,31 occ.	19072	734

MP4 Calculations: Mflops as a Function of the Basis Sets





Methyl hydrazine
C1 Symmetry
MP4sdtq and MP4sdq

Summary

- It is evident from these simple calculations that clear (performance) patterns can be established based on different *Gaussian* parameters
- Parameters considered in this study were: basis sets, number of atoms, symmetry and level of theory
- Single CPU performance shows a directly proportional dependency on the size of the parameters
- Multiple CPU performance also shows a dependency on these parameters

Future Work

- Extend this type of analysis to include more options (keywords) within *Gaussian*
- Write a fairly general document with this information
- Use this information to prioritize options that require optimization (vectorization or scalability)

Gaussian Information

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

CRAY Gaussian site: http://home.cray.com/~cpsosa/

Gaussian News@ CRAY

gaussian@cray.com

References:

C.P. Sosa, J. Ochterski, J. Carpenter, M. J. Frisch, "Ab Initio on the CRAY T3E MPP Supercomputer . II", J. Comp. Chem., 19, 1053(1998).

C. P. Sosa and J. Carpenter, "Running Gaussian in a Combined CRAY PVP and T3E Environment", CUG, Spring 1997, San Jose, CA