

Gaussian Performance Analysis on the IBM® pSeries POWER4™ Systems

A Comparison Between p690, p670, p655, p650, and p630

Carlos P. Sosa, IBM, pSeries High-Performance Computing, Chemistry and Life Sciences Development Solutions and University of Minnesota Supercomputing Institute, Minneapolis, MN

John D. McCalpin, IBM, eServer Hardware Performance, Austin, TX

Patton L. Fast, University of Minnesota Supercomputing Institute, Minneapolis, MN

Micheal J. Frisch, Gaussian, Inc., New Haven, CT

January 6, 2003

©2003 International Business Machines Corporation, all rights reserved

Abstract

We compare the performance of the new family of IBM POWER4 processors, eServer pSeries™, such as the p630, p650, p655, p670 and p690. The pSeries 690 scales up to 32-way with 1.30 GHz POWER4 processors. The p670 can scale up to 32-way POWER4 processor at 1.10 GHz. The p630, 650, and p655 are considered entry to medium level systems. The p630 runs at 1.00 GHz and 1.45 GHz with 1 to 4 processors, the p650 with a 1.45 GHz micro-processor can be configured with 1 to 8 processors. Finally, the p655 running at 1.30 GHz has 1 to 4 processors and the p655 1.10 GHz can include up to 8 processors. The *Gaussian98* Revision A.11 and A.11.3 series of electronic structure programs were used to compare the performance of the various eServers. *Gaussian98* models molecular systems from basic laws of quantum mechanics. We have run calculations with methods such as Hartree-Fock and density functional theory, including first and second derivatives, to analyze and compare the performance of the various pSeries eServers. In addition, we also look at excitation calculations via single-excitation configuration interaction. Single-point Moller-Plesset second order energy calculations were used to test the distributed-memory of the servers. The set of benchmarks were selected to be comparable with the results from previous studies.

Introduction

The combined effort between software and hardware is what makes it possible to carry out calculations that a few years ago were either impossible or too time consuming. An RHF/6-31G** (restricted Hartree-Fock with 6-31G** basis set) single point energy calculation on triamino-trinitro-benzene (TATB) using *Gaussian88* on an IBM RS/6000® model 550 in 1993 took 4.5 hours [1]. The same calculation using *Gaussian92* took approximately 1 hour [1]. Today, this calculation takes less than 1 minute using *Gaussian98* on the latest IBM POWER4 systems.

In addition to faster processors, chemists have long realized that effective parallel computing requires scalable software [2] and hardware [3]. Scalable hardware can be classified in terms of how memory is organized [4,5]: *distributed*-memory, in which each node has its own local memory (e.g., IBM SP nodes); and *shared*-addressable memory among all the processors (shared memory). The IBM pSeries 690 is a major step that IBM has taken toward providing customers with shared memory machines.

The next generation of the IBM POWER series was introduced in October 2001. This new machine scales up to 32-way with 1.30 GHz POWER4 processors (pSeries 690) or 32-way with 1.10 GHz POWER4 processors (pSeries p670), each available with up to 256 GB of memory. The IBM pSeries 690 uses multi-chip modules (MCM), which are equivalent to a mainframe's central processing module. This approach optimizes chip-to-chip communications, boosting performance of the overall system. The module-to-module interconnect runs at half of the processor's frequency [6].

Gaussian98 can take advantage of both memory approaches, distributed or shared. Previous studies have reported the performance of *Gaussian*[®] running on different architectures [3,7,8,9], including pSeries; however, it is beyond the scope of this work to review the *Gaussian* parallel implementation in detail. In the next section we present the design features of the systems tested on this study. We also provide a brief review section that describes how *Gaussian* has been parallelized on distributed and shared memory architectures. This work is summarized in Ref. [9]. In the last four sections we describe the benchmarks and their performance. Although we try to maintain a constant set of benchmarks, it is important to note that from time to time we update this set to reflect new functionality in *Gaussian* or particular features where performance might play an important role and has not been exposed with our current set of benchmarks. We conclude this work with a summary.

Design Features

Several different types of IBM pSeries servers were used in this study: 1.00 GHz and 1.45 GHz p630 Model 6E4 multiprocessor (1-, 2-, and 4-way SMP), 1.45 GHz p650 Model 6M2 multiprocessor (1-, 2-, 4-, and 8-way SMP), 1.10 GHz p655 multiprocessor (1-, 2-, 4-, and 8-way SMP), 1.30 GHz p655 multiprocessor (1-, 2-, and 4-way SMP), 1.10 GHz p670 multiprocessor (up to 32-way SMP), and a p690 multiprocessor (up to 32-way SMP) 1.30 GHz POWER4 pSeries 690 Turbo.

The pSeries 690 server is the latest UNIX[®] server from IBM, with the POWER4 MCM at the core of this latest architecture [6,10]. The building blocks for the systems utilized here are an 8-way MCM Turbo running at 1.1 GHz and 1.30 GHz. The Turbo system (8-way MCM) has two cores per L2 cache; hence, one MCM is 8-way. A full description of the POWER4 architecture is beyond the scope of this work; however, in this section we will provide an overview of the most important features of this architecture. Further details are given in Refs. [6] and [10].

Each processor chip on the pSeries 670 and 690 consists of two microprocessors, an L2 cache that runs at the same speed as the microprocessor, the microprocessor interface unit (i.e. the interface for each microprocessor to the rest of the system), the directory and cache controller for the L3 cache, the fabric bus controller, and a GX bus controller that enables I/O devices to connect to the central electronic complex (CEC). The L3 cache is a new component which was not available on the POWER3 architecture. The L3 caches are mounted on a separate module.

The p630, p650, and p655 take full advantage of the p690 architectural features. These machines use the IBM's POWER4 chip technology in a building-block approach for medium to large systems. This of course is consistent with the design principles of the p690, that is, full system design first[6b]. One key difference with the p630 is that the packaging uses a Single Chip Module (SCM) containing either one or two processor cores[11]. On the other hand, the p650 is the first p650 series system to utilize the POWER4+ chip. Similar to the p630 each chip is packaged on an SCM[11,12]. The pSeries 655 in comparison with the p630 and p650 is packaged with four-chip POWER4 MCM, the basic building block for the pSeries p690. The four-chip MCMs are either 8-way with a clock-speed of 1.10 GHz or 4-way with a clock-speed of 1.30 GHz[13]. In the case of the p655 1.10 GHz, in addition to using one node, we also utilized two and eight nodes. The two and eight nodes were connected with the standard IP network connection, a dual plane switch, and single plane switch.

All the timings in this work correspond to elapsed time. Timings were measured by using the clock and CPU times printed by each link using the *Gaussian* #P option[14]. This information was used as input for a utility program that tabulates timings and speedups[15]. The *Gaussian98* binaries utilized in this study were compiled on a POWER3 system with the xlf 7.1.0.2, xlf 7.1.1.3 FORTRAN compilers [17], and the Atlas libraries.

Parallel Gaussian

Gaussian [14] is a connected series of programs that can be used for performing a variety of electronic structure calculations; molecular mechanics, semi-empirical, *ab initio*, and density functional theory. *Gaussian* consists of a collection of programs commonly known as links; each link communicates through disk files and are grouped into overlays [16]. Links are independent executables located in the g98 directory and labeled as *lxxx.exe*; where *xxx* is the unique number of each link. In general, overlay zero is responsible for starting the program, including reading the input file. After the input file is read, the route card (keywords and options that specify all the *Gaussian* parameters) is translated into a sequence of links. Overlay 99 (*l9999.exe*) terminates the run, in most cases *l9999.exe* finishes with an archive entry (brief summary of the calculation).

As previously pointed out, the *Gaussian* architecture on distributed or shared-addressable memory systems is basically the same [3, 9]. Each link is responsible for continuing the sequence of links by invoking an `exec()` system call to run the next link. The links that run sequentially are mainly links that are responsible for setting up the calculation and assigning symmetry. Although in previous publications we have summarized all the links that run in parallel [3], it is important to note that the use of multiple processors benefit from calculations that make use of the PRISM routines [22].

In a self-consistent field (SCF) scheme, the two-electron integrals are part of the Fock matrix[19]:

$$F_{\mu\nu} = H_{\mu\nu} + \sum_{\lambda} \sum_{\sigma} P_{\lambda\sigma} [(\mu\nu|\lambda\sigma) - 1/2(\mu\lambda|\nu\sigma)] \quad (1)$$

where H represents the core Hamiltonian. μ , ν , λ , σ are atomic orbital indices. The quantities $(\mu\nu|\lambda\sigma)$ are two-electron repulsion integrals. In *Gaussian*, these quantities are computed once and stored or recomputed as many times as needed, depending on the disk space and memory available and the algorithm chosen.

In density functional theory (DFT), Eq. (1) can be rewritten by replacing the last term by the well-known exchange-correlation term F^{XC} [20].

$$F_{\mu\nu} = H_{\mu\nu} + \sum_{\lambda} \sum_{\sigma} P_{\lambda\sigma} (\mu\nu|\lambda\sigma) + F_{\mu\nu}^{\text{XC}} \quad (2)$$

In *Gaussian*, the two-electron integrals are parallelized by distributing batches of integrals among all the available processors (normally selected using the `%nproc` keyword in the

input file). This procedure is illustrated in Figure 1. The main loop over N_{PROCS} (number of available processors) determines the number of integrals that will be distributed among the processors. Each task looks at $1/N_{\text{PROCS}}$ of the shell quartets, discarding those that do not need to be done due to symmetry, cutoffs, or the fact that they have already been done. All the integrations are carried out in these loops, within these series of loops and still in the main N_{PROCS} loop. The last loop sums up the contributions to the Fock matrix, derivative matrices, or density matrices, depending on the types of integrals computed. The next loop outside the first N_{PROCS} loop is another N_{PROCS} loop that adds all the contributions to the Fock matrix together in a serial block of code. This scheme can be exploited to compute first and second two-electron integral derivatives, first and second one-electron integral derivatives, and electrostatic potential integrals.

```

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

```

Figure 1. Parallel loops to compute two-electron integrals in *Gaussian*.

The parallelization of the Fock matrix in *Gaussian98* on IBM shared-memory architectures has been accomplished by using standard UNIX fork and wait commands [9c]. `fork()` creates a new process. Upon successful completion, the fork subroutine returns a value of 0 to the child process and returns the process ID of the child process to the parent process. Otherwise, a value of -1 is returned to the parent process and no child process is created. In *Gaussian98* `fork()` is called within the N_{PROCS} loop and each new child executes a batch of integrals. All the children that have completed their task wait prior to entering the sequential do loop to add their contributions.

In addition, on all the pSeries systems *Gaussian* calculations can be carried out using the distributed-memory version. In this version the PRISM algorithm has been implemented using the Linda parallel programming model based on distributed data structures[20]. Processors and data objects communicate by means of the tuple space (TS) or Linda memory. This type of shared-addressable memory-like can be achieved on a truly distributed memory machine (clusters of p630, p650 or p655) or on a shared-memory machine such as the p670 or p690. In this study we illustrate the Linda parallel capabilities using the second-order Moller-Plesset (MP2) functionality in *Gaussian*. It is well known that the MP2 energy based on the Restricted Hartree-Fock (RHF) wave function can be expressed as:

$$E_{MP2} = E_{HF} + E^{(2)} = E_{HF} + \frac{1}{4} \sum a_{ij}^{ab} (ij \parallel ab) \quad (3)$$

$$a_{ij}^{ab} = \frac{\langle ij || ab \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b} \quad (4)$$

The notations ij and ab correspond to the occupied (O) and virtual (V) molecular orbitals, which are eigenfunctions of the Fock operator with eigenvalues ϵ_i , ϵ_j and ϵ_a , ϵ_b .

In Figure 2 we reproduce the algorithm that we presented in a previous publication[3]. It illustrates the use of Linda in the parallelization of the MP2 algorithm. Here we briefly summarize how this takes place.

Store scalar variables in tuple space (COMMON BLOCKS)

```

loop over  $N_{\text{procs}}$  ( $P=(N_{\text{procs}} - 1), 0, -1$ )
  if(P.eq.0)then
    do work on MASTER processor
  else
    eval (LindE2)
  endloop
loop over  $P_{\text{es}} - 1$ 
  store and/or retrieve results from tuple space(arrays)
  sum up energy contributions
  endloop
store or read scalar variables from tuple space
store or read array variables from tuple space
  compute MP2 energy

```

Figure 2. Parallel loops to compute MP2 energies in *Gaussian*.

This algorithm shows in line 1 that the common blocks need to be defined in the tuple space. Within the *Gaussian* program this step is carried out via `d2lin1` which functions solely to store/retrieve common blocks in/from the tuple space. Similarly this is done with `d2lin1` and `d2lin3`. On line 2 the main loop checks if the master is ready to do work; if not, it spawns workers to start transforming batches of integrals. The spawning of workers is carried out by performing an eval operation on the `linde2` routine. `linde2` is the driver routine that manages the computation of the integrals. Each worker gets a batch of i indices using one or more passes over the entire set of atomic orbital integrals. Each worker requires $O(OVN)$ of local memory to perform the integral transformation. As the $(ialjb)$ integrals are completed, their contribution to their MP2 energy is added. When all workers have completed the integrals, the partial MP2 energies are accumulated

Selected Benchmarks

Similarly, as in previous studies [3,9], we consider four major characteristics when studying parallel performance on the IBM SP or selecting benchmarks: job type, theoretical method, basis set size, and molecular size. The job type corresponds to a single point energy calculation, a gradient calculation, or calculation of second derivatives. Normally single point calculations are used to compute accurate energies at a level of theory that is too expensive to carry out a full geometry optimization. Due to the importance of molecular structure in chemistry, a large majority of calculations are geometry optimizations using HF, DFT, or MP2. Geometry optimizations are normally followed by a frequency calculation.

There are many options for carrying out calculations using *Gaussian*. In this study we have tried to select job types that reflect how users are currently running *Gaussian*. This is by no means a full representation of the options available in the program, but they do represent a large percentage of calculations carried out by typical researchers at computer centers. They also correspond to many of the benchmarks carried out for hardware procurements.

In this study we have chosen the following types of calculations: 1) single-point energy (SP) calculations at different levels of theory, 2) FORCE: this type of calculation corresponds to an SP calculation followed by the calculation of the first derivatives of the energy with respect to the position of the atoms in the molecule, and 3) frequency: this is a calculation of the second derivative of the energy and involves many SP and force calculations. The time required for a geometry optimization is a multiple of the time needed for a single FORCE calculation. *Rather than doing a full geometry optimization, a FORCE calculation is recommended for benchmarks that involve hardware performance.* It is equivalent to doing one cycle of the optimization and should provide a good approximation for the performance of an optimization calculation.

A large number of approximate theoretical methods have been reported in the literature [19]. These methods range in accuracy and computational cost. Since *Gaussian* provides most of these methods, it is important to understand how they perform as a function of system resources. In this study we refer to system resources as the number of processors, memory, and disk space needed for optimal performance. The theoretical methods chosen in this study have been extensively discussed in the literature [19] and it is beyond the scope of this work to describe these methods. The approximations used in this work correspond to Hartree-Fock [19], the three parameter hybrid density functional theory of Becke (B3-LYP) [21,22], and configuration interaction singles (CIS) energy and gradients [23].

The cases used in this study correspond to the same cases from a previous study [9b,9c]. Case I is an SP calculation of α -pinene at the HF level of theory using the 6-311G(df,p) basis set. Case II corresponds to a FORCE calculation on a fairly large system, valinomycin. Case III is an α -pinene frequency calculation at the B3-LYP/6-31G(d) level of theory. Cases II and III exercise several of the links that run in parallel. Case IV is a FORCE calculation of acetyl phenol excited states using CIS with the 6-31++G basis set [19]. Case V is a single-point MP2 energy on a hydrocarbon chain ($C_{20}H_{42}$). This last case is to test the distributed memory capabilities via Linda. This set of benchmarks represents small to large systems to test speedup and efficiency for most parallel links. All the geometries are available from Ref. [24].

Results and Discussion

In this study, as in our previous reports, we look at performance in terms of speedup [6b,6c]. Speedup (S) is defined as the ratio of the serial run time (elapsed time, t_s) over the time that it takes to do the same problem in parallel (elapsed time, t_p).

$$S = \frac{t_s}{t_p} \quad (5)$$

Efficiency (e) is the fraction of time that a processor is doing useful work. This measurement also indicates or provides an indirect indicator for the percentage of parallel code needed for linear or nearly linear scalability.

$$e = \frac{S}{N_{PROCS}} \quad (6)$$

To compare scalability we look at the measured speedup against ideal speedup, rather than using an extrapolated speedup as we have done in the past[9a]. We take this approach because we want to compare how POWER4 machines perform, rather than analyzing *Gaussian's* scalability. *It is important to note that clock speed of the POWER4 is significantly faster than all the machines tested on all our previous studies and this may affect scalability.*

Table 1 illustrates the performance between several pSeries POWER4 for a Hartree-Fock single-point energy calculation using a medium size basis set (Case I). Parallelization is achieved mainly in *l502.exe* (almost all of the CPU is spent in *l502.exe*). Table 1 looks at the scalability of *l502.exe* (or SCF procedure) and of the entire calculation (Total time or time to solution). The column under the label "Total" includes the overhead introduced by the links that run sequentially[9b,9c]. In the case of a single processor we have shown that the performance improvement in going from the POWER3, 375 MHz to the POWER4, 1.30 GHz (Turbo) is a factor of approximately 3.34 for *l502.exe* and the total time[9b,9c]. Similarly, in the case of the POWER4 pSeries family, for case I, we can order them from the fastest to the slowest as follows: p655 1.30 GHz > p690 Turbo > p655 1.10 GHz > p670 1.10 GHz > p650 1.45 GHz > p630 1.45 GHz > p630 1.00 GHz for *l502.exe* and the total time. If we normalize timings with respect to the p630 then the

speedups (total time) are: 16%, 21%, 29%, 39%, 22%, and 32% for p630 1.45 GHz, p650 1.45 GHz, p655 1.10 GHz, p655 1.30 GHz, p670 1.10 GHz, and p690 Turbo, respectively.

Table 1. Hartree-Fock Single-Point Energy Calculations on α -pinene(C₁₀H₁₆)^a

Number of Processors ^c	<i>l502.exe</i> ^b	S ^c	e ^c	Total ^{b,d}	S ^c	e ^c
1						
p630 1.00 GHz	1246	1.00	1.00	1251	1.00	1.00
p630 1.45 GHz	1049	1.00	1.00	1053	1.00	1.00
p650 1.45 GHz	984	1.00	1.00	986	1.00	1.00
p655 1.10 GHz	882	1.00	1.00	886	1.00	1.00
p655 1.30 GHz	763	1.00	1.00	766	1.00	1.00
p670 1.10 GHz	973	1.00	1.00	977	1.00	1.00
p690 Turbo	843	1.00	1.00	847	1.00	1.00
2						
p630 1.00 GHz	875	1.42	0.71	880	1.42	0.71
p630 1.45 GHz	519	2.01	1.00	523	2.01	1.00
p650 1.45 GHz	509	1.93	0.97	512	1.93	0.97
p655 1.10 GHz	477	1.85	0.93	482	1.84	0.92
p655 1.30 GHz	417	1.83	0.92	422	1.82	0.91
p670 1.10 GHz	504	1.93	0.97	509	1.92	0.96
p690 Turbo	439	1.92	0.96	444	1.91	0.96
4						
p630 1.00 GHz	682	1.83	0.92	689	1.82	0.91
p630 1.45 GHz	398	2.64	0.66	403	2.61	0.65
p650 1.45 GHz	308	3.19	0.80	313	3.15	0.79
p655 1.10 GHz	282	3.13	0.78	288	3.08	0.77
p655 1.30 GHz	251	3.04	0.76	256	2.99	0.75
p670 1.10 GHz	325	2.99	0.75	331	2.95	0.74
p690 Turbo	282	2.99	0.75	288	2.94	0.74
8						
p650 1.45 GHz	226	4.35	0.54	232	4.25	0.53
p655 1.10 GHz	208	4.24	0.53	217	4.08	0.51
p670 1.10 GHz	193	5.04	0.63	203	4.81	0.60
p690 Turbo	164	5.14	0.64	173	4.90	0.61
16						
p670 1.10 GHz	114	8.54	0.53	129	7.57	0.47
p690 Turbo	97	8.69	0.54	111	7.63	0.48
32						
p670 1.10 GHz	73	13.33	0.42	99	9.87	0.31
p690 Turbo	63	13.38	0.42	85	9.96	0.31

^a Total of 346 basis functions; 6-311G(df,p) basis set; C₁ symmetry.

^b All timings are in seconds and correspond to elapsed time.

^c Speedup and efficiency.

^d Total time to complete the run.

The scalability observed in this case is what one might expect (see Figures 3 and 4 for scalability of *l502.exe* and the entire run). Although this scalability is acceptable, the POWER3-based system shows slightly better scalability[9b,9c]. This is not surprising and partly we have attributed this difference due to the much faster clock-speed on the POWER4. In addition, we have not accounted for the difference in the size of the L2 cache. On the POWER3 systems that we have used in the past, the L2 cache is 8 MB while the L2 cache on the POWER4 is 1.41 MB. Preliminary studies using ethylene and caffeine molecules show that by optimizing for the L2 cache on the POWER4, we have seen single-processor performance improvements anywhere from 5% to more than 30%[25].

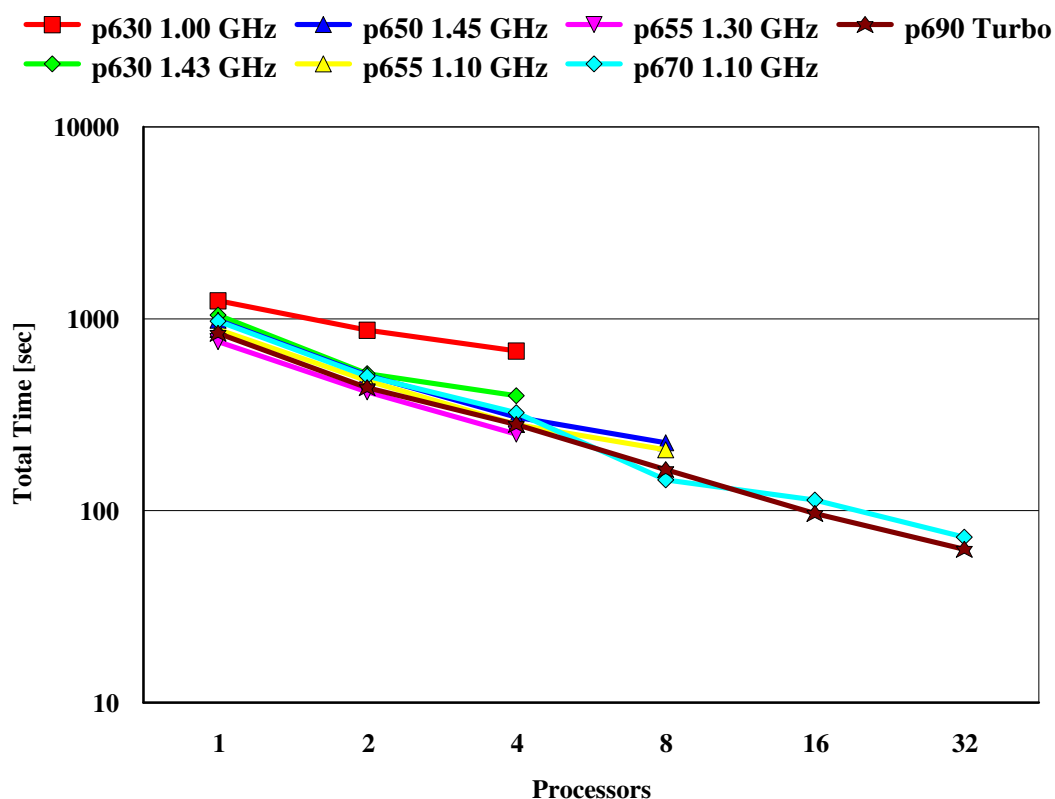


Figure 3. Scalability for *l502.exe* as a function of processors for the Hartree-Fock single-point energy calculation on α -pinene. .

Up to four-processors, the efficiencies for the pSeries machines are very similar. Also, it is interesting to note that as the number of processors increases and the parallel regions are completed more quickly, the serial overhead begins to dominate the calculation. This is illustrated in Table 1, Figure 3, and Figure 4. The efficiency between *l502.exe* and the ones that correspond to the entire run are almost indistinguishable up to 4 processors. This may indicate that the optimal number of processors for this particular case is indeed between 4 and 8 processors on POWER4 as previously reported[9b,9c]. From Table 1, we see that the efficiency decreases rapidly for a large number of processors. Larger basis sets are required to maintain a very high efficiency with a large number of processors. In other words, if the molecule remains the same (number and type of atoms), to maintain a high efficiency we need to increase the basis set to increase the number of integrals that are computed by each processor.

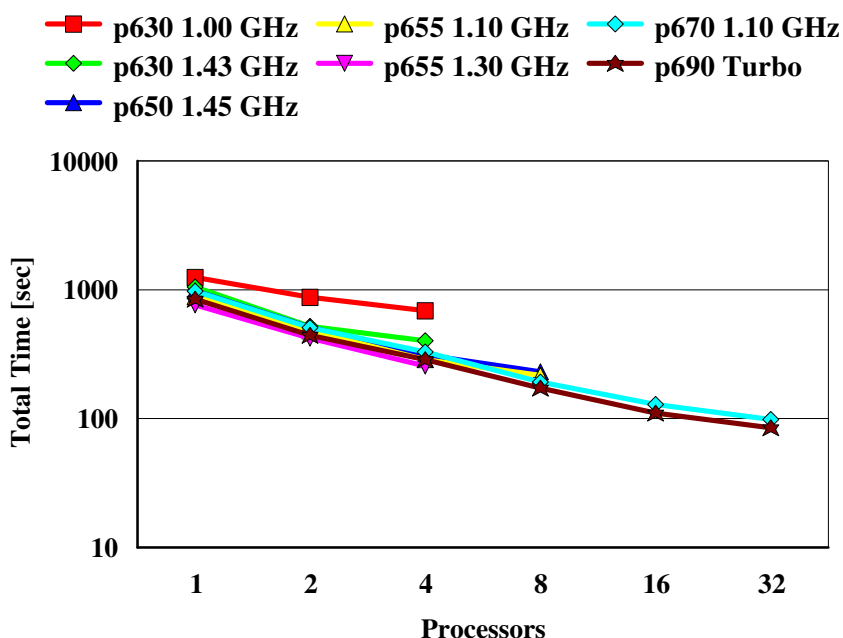


Figure 4. Total time as a function of processors for the Hartree-Fock single-point energy calculation on α -pinene.

Table 2 summarizes a FORCE calculation. In addition to *l502.exe*, this case also illustrates the scalability for *l703.exe* and of course, the scalability for the complete run when using DFT methods. Although these systems (and basis sets) are not the same, an approximate comparison may be obtained between *l502.exe* in Table 1 and Table 2.

Table 2. B3-LYP FORCE Calculation on Valinomycin ($C_{54}H_{90}N_6O_{18}$)^a

Number of Processors ^e	<i>l502.exe</i> ^b	S ^c	<i>l703.exe</i> ^b	S ^c	Total ^{b,d}	S ^c
1						

p630 1.00 GHz	17,985	1.00	4,444	1.00	22,508	1.00
p650 1.45 GHz	13,864	1.00	3,394	1.00	17,321	1.00
p655 1.10 GHz	13,242	1.00	3,077	1.00	16,391	1.00
p655 1.30 GHz	11,372	1.00	2,668	1.00	14,102	1.00
p670 1.10 GHz	14,231	1.00	3,502	1.00	17,806	1.00
p690 Turbo	12,437	1.00	3,039	1.00	15,541	1.00
2						
p630 1.00 GHz	12,830	1.40	3,361	1.32	16,268	1.38
p630 1.45 GHz	7,575		1,943		9,578	
p650 1.45 GHz	7,273	1.91	1,727	1.97	9,060	1.91
p655 1.10 GHz	7,254	1.83	1,641	1.88	8,965	1.83
p655 1.30 GHz	6,292	1.81	1,437	1.86	7,789	1.81
p670 1.10 GHz	7,603	1.87	1,805	1.94	9,478	1.88
p690 Turbo	6,509	1.91	1,651	1.84	8,222	1.89
4						
p630 1.00 GHz	7,729	2.33	2,442	1.82	10,254	2.20
p630 1.45 GHz	6,230		1,719		8,015	
p650 1.45 GHz	4,676	2.96	1,135	2.99	5,874	2.95
p655 1.10 GHz	4,621	2.87	1,082	2.84	5,777	2.84
p655 1.30 GHz	4,100	2.77	977	2.73	5,141	2.74
p670 1.10 GHz	5,122	2.78	1,261	2.78	6,458	2.76
p690 Turbo	4,421	2.81	1,095	2.78	5,583	2.78
8						
p650 1.45 GHz	3,543	3.91	943	3.60	4,555	3.80
p655 1.10 GHz	3,389	3.91	830	3.71	4,299	3.81
p670 1.10 GHz	3,137	4.54	739	4.74	3,957	4.50
p690 Turbo	2,692	4.62	626	4.85	3,389	4.59
16						
p670 1.10 GHz	2,092	6.80	471	7.44	2,661	6.69
p690 Turbo	1,788	6.96	398	7.64	2,273	6.84
32						
p670 1.10 GHz	1,606	8.86	349	10.03	2,084	8.54
p690 Turbo	1,370	9.08	294	10.34	1,777	8.75

^a Total of 882 basis functions; 3-21G basis set; C1 symmetry.

^b All timings are in seconds and correspond to elapsed time.

^c Speedup.

^d Total time to complete the run.

Similar to Case I, we normalize our results with respect to the p630 1.00 GHz and we look at the single processor performance. The speedups are approximately: 23%, 26%, 37%, 21%, and 31% for the p650 1.45 GHz, p655 1.10 GHz, p655 1.30 GHz, p670 1.10 GHz, and p690 Turbo, respectively. Figure 5 shows scalability for the entire run.

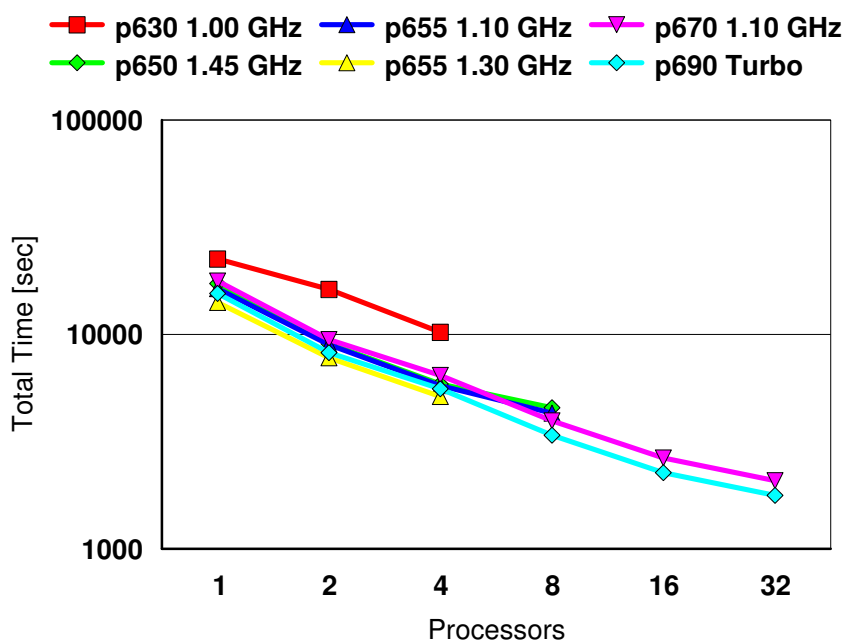


Figure 5. Total time as a function of processors for the B3-LYP FORCE calculation on Valinomycin.

For *l502.exe* the parallel results are similar to α -pinene, even though this is a larger system with more basis functions. This case shows slightly lower scalability than Case I which may be attributed to less integrals per batch; this lower scalability is either due to the small basis set or cutoff used for the integrals for large systems (loop sizes to compute integrals are also shorter).

l703.exe computes the first and second-derivatives of the two-electron integrals. We showed in our previous study that in Case II we saw that the POWER3 shows superlinear scalability with two processors and very good scalability up to 16 processors. The efficiency on the POWER3 with 16 processors is 65%. On the other hand, the efficiency on the Turbo 1.30 GHz or p670 1.10 GHz is about 50%. Again, this can be attributed to faster processor and/or large L2 cache.

Table 3 summarizes timings for a frequency calculation. In Case III, there are several parallel links that are exercised in this example: *l502.exe*, *l1110.exe*, *l1002.exe*, and *l703.exe*. *l502.exe* and *l703.exe* have been discussed previously. *l1002.exe* solves the couple-perturbed Hartree-Fock (CPHF) equations to produce the derivatives of the molecular orbital coefficients. And *l1110.exe* computes the two-electron contribution to the Fock matrix derivatives with respect to nuclear coordinates [16]. Since only the direct scheme of the atomic orbitals (AO) production integrals is parallelized, it is not surprising that *l1002.exe* does not show the same type of scalability as *l703.exe* and *l1110.exe* [3].

Table 3. B3-LYP Frequency Calculation on α -pinene (C₁₀H₁₆)^a

Number of Processors	<i>l502.exe^b</i>	<i>l1110.exe^b</i>	<i>l1002.exe^b</i>	<i>l703.exe^b</i>	Total ^{b,d}	S ^c
1						
p630 1.00 GHz	481	1635	1919	2017	6059	1.00
p630 1.45 GHz	391	1225	1624	1539	4784	1.00
p650 1.45 GHz	376	1203	1476	1478	4538	1.00
p655 1.10 GHz	388	1446	1668	1721	5229	1.00
p655 1.10 GHz ^e	392	1460	1672	1746	5276	1.00
p655 1.30 GHz	333	1229	1418	1465	4450	1.00
p670 1.10 GHz	412	1466	1720	1789	5393	1.00
p690 Turbo	355	1244	1469	1531	4603	1.00
2						
p630 1.00 GHz	297	913	1133	1157	3508	1.73
p630 1.45 GHz	192	619	855	784	2456	1.95
p650 1.45 GHz	190	609	764	757	2325	1.95
p655 1.10 GHz	202	736	885	890	2720	1.92
p655 1.10 GHz ^e	277	735	1143	880	3044	1.73
p655 1.30 GHz	175	626	753	760	2319	1.92
p670 1.10 GHz	212	746	920	924	2809	1.92
p690 Turbo	182	631	777	807	2403	1.92
4						
p630 1.00 GHz	205	528	665	760	2169	2.79
p630 1.45 GHz	123	372	525	505	1533	3.12
p650 1.45 GHz	109	348	466	440	1370	3.31
p655 1.10 GHz	111	414	518	506	1559	3.35
p655 1.10 GHz ^e	164	388	707	469	1736	3.04
p655 1.30 GHz	97	352	433	435	1325	3.35
p670 1.10 GHz	125	419	550	533	1636	3.30
p690 Turbo	108	369	477	454	1416	3.25
8						
p650 1.45 GHz	70	213	307	292	893	5.08
p655 1.10 GHz	73	245	355	321	1009	5.18
p655 1.10 GHz ^e	90	218	423	261	1003	5.26
p655 1.30 GHz	72	232	362	312	992	5.44
p670 1.10 GHz	63	202	313	267	857	5.37
p690 Turbo	56	191	285	260	804	5.71
16						
p670 1.10 GHz	42	127	264	169	628	8.59
p690 Turbo	37	109	227	145	541	8.51
32						
p670 1.10 GHz	28	72	222	100	471	11.45
p690 Turbo	25	62	192	86	406	11.34

^a Total of 346 basis functions; 6-31G(d) basis set; C1 symmetry.

^b All timings are in seconds and correspond to elapsed time.

^c Speedup.

^d Total time to complete the run.

^e Linda version; this calculation was carried out on two p655 1.10 GHz nodes.

In the case of the POWER3, while *l502.exe* and *l1110.exe* show superlinear scalability with 2 processors, the speedup for *l1002.exe* is 1.92[9b,9c]. However, on the Turbo 1.30 GHz and p670 1.10 GHz we do not see superlinear scalability the trend is the same with 2 processors as on the POWER3. Figure 5 illustrates different scalabilities for the various links in this calculation. Parallelization of additional terms in the solution of Coupled-Perturbed Hartree-Fock (CPHF) equations will help the scalability of *l1002.exe* [25].

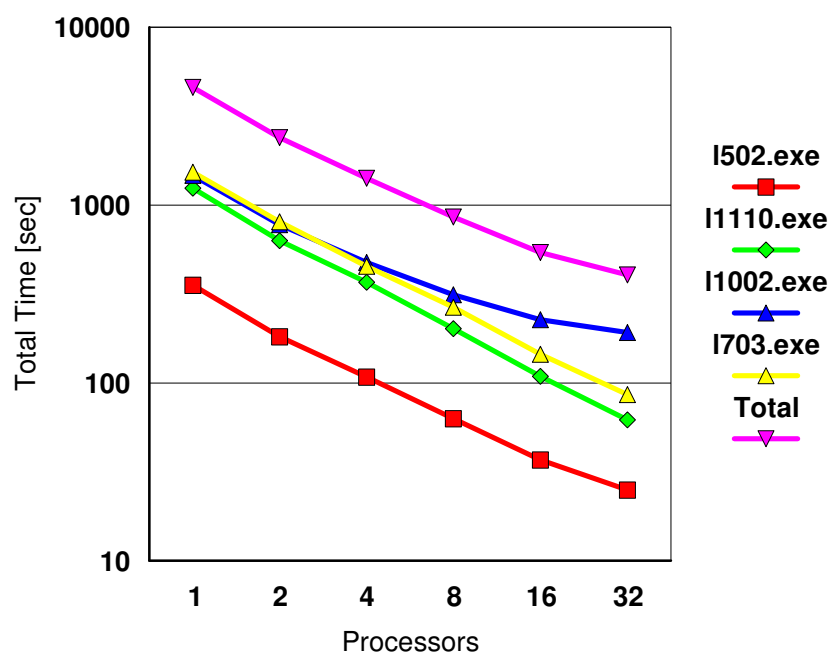


Figure 6. Scalability as a function of processors for the different links in the B3-LYP frequency calculation for α -pinene. All the timings correspond to p690 Turbo

On the other hand *l1110.exe* shows excellent scalability, similar to *l703.exe*. *l1110.exe* shows an almost linear scalability up to 16 processors. S is 1.54, 11.41, and 12.42 (speedups for individual links not shown in Table 3) for the p670 1.10 GHz, and p690 Turbo, respectively.

Table 3 also provides a direct comparison between the *Gaussian* version based on fork and wait (shared-memory) and the Linda distributed-memory version. A more detailed analysis will be presented in Table 5. A comparison between the shared-memory and

distributed-memory on a single processor for the α -pinene frequency calculation shows no difference in performance. This of course is not surprising. However, as we increase the number of processors we begin to see that the shared-memory shows slightly better scalability. This is particularly true for *11002.exe*.

Finally, Figure 7 summarizes the performance of the time to solution, that is, the time that it takes to complete the entire calculation.

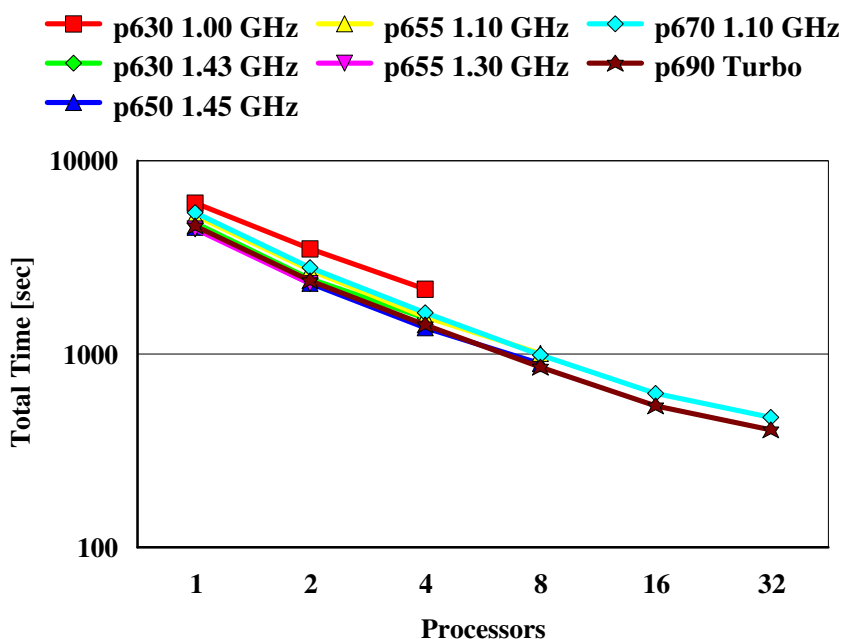


Figure 7. Total time as a function of processors for the B3-LYP frequency calculation on α -pinene.

Table 4 summarizes elapsed timings and total speedups for acetyl-phenol (Case IV). This benchmark consists of a CI-singles energy and FORCE calculation. This example illustrates the scalability of *1914.exe*. *1914.exe* computes excited states using CI-singles excitations [23]. This type of calculation runs in parallel since the repulsion two-electron integrals contributing to the CI-singles can be computed using the PRISM algorithm. The PRISM algorithm runs in parallel as shown in the previous sections.

Table 4. CI-Singles Energy and Force Calculation on Acetyl Phenol ($C_8H_8O_2$).

Number of Processors	1502.exe ^b	1914.exe ^b	11002.exe ^b	1703.exe ^b	Total ^{b,d}	S ^c
1						
p630 1.00 GHz	335	670	294	74	1378	1.00
p630 1.45 GHz	284	536	252	66	1142	1.00
p650 1.45 GHz	269	571	239	61	1143	1.00
p655 1.10 GHz	240	453	212	51	961	1.00
p655 1.30 GHz	207	394	186	46	835	1.00

p670 1.10 GHz	267	497	237	60	1065	1.00
p690 Turbo	231	432	205	52	924	1.00
2						
p630 1.00 GHz	193	418	202	50	869	1.59
p630 1.45 GHz	129	266	124	34	557	2.05
p650 1.45 GHz	126	289	122	32	573	1.99
p655 1.10 GHz	119	246	115	28	514	1.87
p655 1.30 GHz	105	215	101	26	451	1.85
p670 1.10 GHz	126	257	124	31	544	1.96
p690 Turbo	110	225	107	27	476	1.94
4						
p630 1.00 GHz	197	340	174	48	767	1.80
p630 1.45 GHz	108	193	94	27	428	2.67
p650 1.45 GHz	86	173	75	21	360	3.17
p655 1.10 GHz	81	151	69	18	326	2.95
p655 1.30 GHz	72	134	62	16	290	2.76
p670 1.10 GHz	90	167	78	22	364	2.93
p690 Turbo	79	147	68	17	319	2.90
8						
p650 1.45 GHz	64	118	53	15	259	4.41
p655 1.10 GHz	60	108	50	13	245	3.92
p670 1.10 GHz	56	103	46	12	230	4.63
p690 Turbo	49	92	40	11	203	4.55
16						
p670 1.10 GHz	38	63	26	8	157	6.78
p690 Turbo	30	58	21	6	135	6.84
32						
p670 1.10 GHz	38	43	26	5	151	7.05
p690 Turbo	30	39	21	4	130	7.11

^a Total of 154 basis functions; 6-31++G basis set; C1 symmetry.

^b All timings are in seconds and correspond to elapsed time.

^c Speedup.

^d Total time to complete the run.

This link shows (see Table 4) good scalability consistent with *l502.exe*. Clearly, this case is becoming too small for the POWER4. All the links with 16 processors take less than 60 seconds. *l703.exe* takes only 5 seconds. Furthermore, as illustrated in Figure 8, the scalability in going from 16 to 32 processors is almost flat. Figure 8 illustrates the total time as a function of processors for the CI-singles energy and gradients.

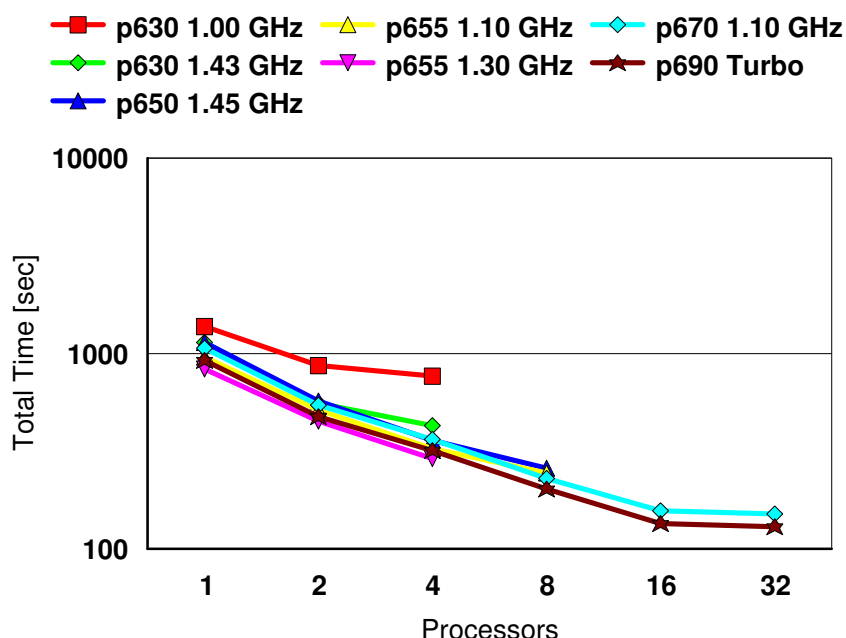


Figure 8. Total time as a function of processors for the CI-singles energy and FORCE calculation on acetyl Phenol.

Table 5 summarizes all the calculations carried out at the MP2 level. This set of calculations illustrates scalability for the Linda version as a function of the connectivity between nodes. This calculation was carried out on a linear hydrocarbon ($C_{20}H_{42}$) containing 384 basis functions. The orbitals used for correlation correspond to molecular orbital 21 through 284 for a total of 61 occupied. The algorithm uses OVN of memory for 11 occupied orbitals at one time. We have a total of six batches of integrals, two batches for four processors and finally one batch for more than four processors.

Table 5. MP2 Calculation on a Linear Hydrocarbon ($C_{20}H_{42}$)^a

Number of Processors	<i>1502.exe</i> ^b	S ^c	<i>1906.exe</i> ^b	S ^c	Total ^{b,d}	S ^c
1						
p655 1.10 GHz ^e	175	1.00	679	1.00	861	1.00
p655 1.10 GHz ^f	175	1.00	681	1.00	862	1.00
p655 1.10 GHz ^g	173	1.00	677	1.00	858	1.00
p655 1.10 GHz ^h	174	1.00	680	1.00	861	1.00
p655 1.10 GHz ⁱ	174	1.00	678	1.00	859	1.00
2						
p655 1.10 GHz ^e	154	1.14	358	1.90	520	1.66
p655 1.10 GHz ^f	152	1.15	345	1.97	506	1.70
p655 1.10 GHz ^g	152	1.14	347	1.95	507	1.69
p655 1.10 GHz ^h	153	1.14	358	1.90	521	1.65
p655 1.10 GHz ⁱ	151	1.15	345	1.97	505	1.70

4						
p655 1.10 GHz ^e	65	2.69	229	2.97	303	2.84
p655 1.10 GHz ^f	88	1.99	197	3.46	293	2.94
p655 1.10 GHz ^g	88	1.97	197	3.44	293	2.93
p655 1.10 GHz ^h	90	1.93	205	3.32	303	2.84
p655 1.10 GHz ⁱ	97	1.79	197	3.44	303	2.83
8						
p655 1.10 GHz ^e	54	3.24	178	3.81	241	3.57
p655 1.10 GHz ^f	40	4.38	101	6.74	150	5.75
p655 1.10 GHz ^g	40	4.33	101	6.70	152	5.64
p655 1.10 GHz ^h	45	3.87	105	6.48	159	5.42
p655 1.10 GHz ⁱ	81	2.15	106	6.40	199	4.32

^a Total of 384 basis functions; 6-31G(d) basis set; C_{2h} symmetry; MAXDISK=110MW.

^b All timings are in seconds and correspond to elapsed time.

^c Speedup.

^d Total time to complete the run.

^eLinda version; this calculation was carried out on one node p655 1.1 GHz nodes.

^fLinda version; this calculation was carried out on eight nodes p655 1.1 GHz nodes; each node interconnected via a dual plane switch.

^gLinda version; this calculation was carried out on eight nodes p655 1.1 GHz nodes; each node interconnected via a single plane switch.

^hLinda version; this calculation was carried out on eight nodes p655 1.1 GHz nodes; each node interconnected via a single plane switch.

ⁱLinda version; this calculation was carried out on eight nodes p655 1.1 GHz nodes; each node interconnected via ethernet.

It is clear from Table 5 and Figure 9 that for a system of 8 processors the speedup is more or less independent of how each node is interconnected. This data illustrates an almost linear speedup up to 8 processors for *1906.exe*. These results are consistent with our previous studies[3].

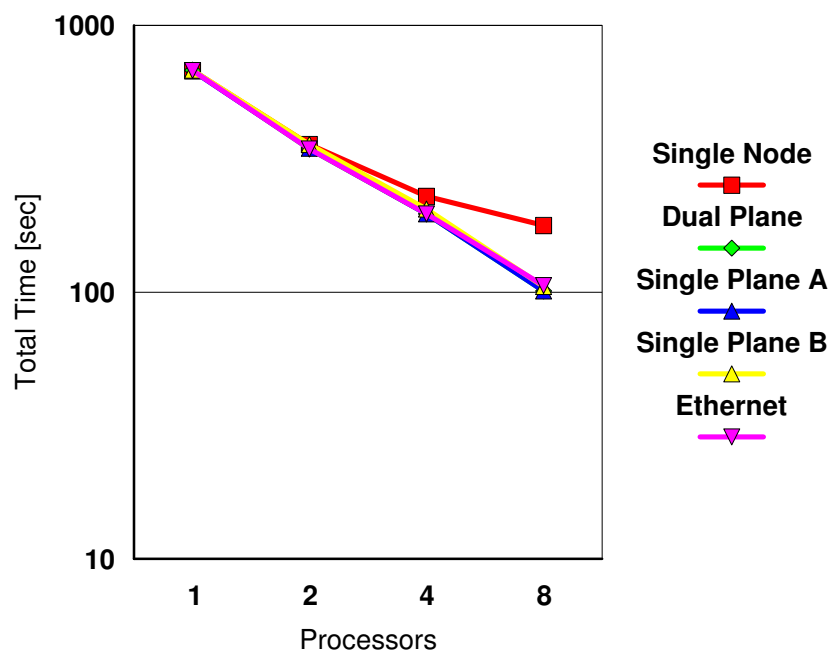


Figure 9. 1906.exe times as a function of processors for the $C_{20}H_{42}$ MP2 single-point calculation.

Summary

In this study we have tried to provide information on the performance of several options in the *Gaussian* program for the new family of IBM POWER pSeries. These options are commonly used by researchers at many supercomputer centers but by no means is this an exhaustive set of benchmarks. In this third exploratory work we have included an MP2 single-point energy calculation to test distributed-memory machines via Linda. The levels of theory tested here correspond to Hartree-Fock (HF), density functional theory (DFT), configuration interaction-single excitations and second order Moller-Plesset perturbation theory. As first approximations to many higher-order methods, this set of methods is routinely used and certainly can provide valuable information when comparing the newest POWER architecture.

In general, HF and DFT calculations with and without first and/or second derivatives have shown to run efficiently on the machines tested in this study as well as the MP2 benchmark tested here. However, when designing benchmarks for scalability it is important to consider the system and basis sets. The first case that we studied here, α -pinene, has been used as a system that scales well with large number of processors [9]. A comparison of the scalability of *1502.exe* for Cases I-III gives an approximate indication of scalability, it is approximate because each case uses a different level of theory with slightly different options and different convergence thresholds. That is in order to achieve better scalability, basis set size is more important than either the level of theory or size of the molecule. In terms of the seven computer systems tested in this study, the speedups presented for this family of pSeries systems can be order for this set of systems as follows: p655 1.30 GHz > p690 Turbo > p650 1.45 GHz > p670 1.10 GHz > p655 1.10 GHz > p630 1.00 GHz.

Acknowledgments

We would like to thank S. Selzo, and S. Qualters for all their help with the “Regatta” systems. CPS would like to thank Bruce Hurley for encouragement and making this type of study possible.

References

- [1] Gaussian News, **4**, 1(1993), Gaussian, Inc., Pittsburgh, PA.
- [2] R. J. Harrison and R. Shepard, *Annu. Rev. Phys. Chem.* **45**, 623(1994).
- [3] C. P. Sosa, J. Ochterski, J. Carpenter, and M. J. Frisch, *J. Comp. Chem.* **19**, 1053(1998); and references therein.
- [4] M. J. Flynn, *Proc. of the IEEE* **54**, 1901(1966).
- [5] M. J. Flynn, *IEEE Trans. on Computers* **C21**, 948(1972).
- [6] a) S. Behling, R. Bell, P. Farrell, A. Holthoff, F. O’Connell, and W. Weir *The POWER4 Processor Introduction and Tuning Guide*, IBM Corporation, International Technical Support Organization, Austin TX, 2001; and references therein; b) J. M. Tendler, J. S. Dodson, J. S. Fields, Jr., H. Le, and B. Sinharoy, *IBM J. Res. & Dev.* **46**, 5(2002).
- [7] J. Ochterski, C. P. Sosa, J. Carpenter, Performance of Parallel Gaussian94 on Cray Research Vector Supercomputers, in: B. Winget and K. Winget, Editors, *Cray User Group Proceedings*, Cray User Group, Shepherdstown, WV, p. 108, 1996.
- [8] D. P. Turner, G. W. Trucks, M. J. Frisch, Ab Initio Quantum Chemistry on a Workstation Cluster, in: T. G. Matson, Editor, *Parallel Computing in Computational Chemistry*, ACS Series 592, American Chemical Society, Washington, DC, p. 62.

- [9] a) C. P. Sosa, G. Scalmani, R. Gomperts, and M. J. Frisch, *Parallel Computing* **26**, 843(2000); b) C. P. Sosa and S. Andersson *Gaussian Benchmarks put the pSeries 690 Server Through Its Paces*, IBM eServer Developer Domain, February 2002, http://www-1.ibm.com/servers/esdd/articles/gauss_bench/index.html; c) C. P. Sosa and P. Fast, *Ab Initio Quantum Chemistry on the IBM pSeries 690: A Comparison Between Turbo 1.3 GHz and 1.1 GHz*, IBM RedPaper (REDP0444), April 2002, <http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp0444.html?Open>
- [10] H. M. Matis, J. D. McCalpin, M-C, Chiang, F. P. O'Connell, P. Buckland IBM pSeries 690 Configuring for Performance, IBM Corporation, Austin, TX, 2001.
- [11] G. Anselmi and S. Lutz, pSeries 650 Model 6M2 Technical Overview and Introduction, IBM Corporation, ITSO, November 2002, Austin, Texas.
- [12] B. Olszewski, IBM eServer p650 Performance Tuning, IBM Corporation, Marketing Communications Server group, December 2002, Somers, NY.
- [13] H. M. Mathis, J. D. McCalpin, J. Thomas, IBM eServer pSeries 655 Basic Building Block for 21st Century High Performance Clustering, IBM Corporation, Marketing Communication server group, November 2002, Somers, NY.
- [14] AE. Frisch and M. J. Frisch, *Gaussian98 User's Reference*, 2nd Edition, Gaussian Inc., Pittsburgh, PA; <http://www.gaussian.com>
- [15] The utility program can be obtained from one of the authors (C.P. Sosa) at: cpsosa@msi.umn.edu
- [16] M. J. Frisch, A. B. Nielsen, and AE. Frisch, *Gaussian98 Programmer's Reference*, Gaussian Inc., Pittsburgh, PA; <http://www.gaussian.com>
- [17] XL Fortran for AIX®, *User's Guide Version 7 Release 1*, IBM Canada, North York, Ontario, Canada, SC09-2866-00, 2000.
- [18] P. M. Gill, M. Head-Gordon, and J. A. Pople, *J. Phys. Chem.*, **94**, 5564(1990).
- [19] [W. J. Hehre, L. Radom, P. V. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory*, John Wiley & Sons, New York, NY, 1985.
- [20] B. G. Johnson, P. M. W. Gill, and J. A. Pople, *J. Chem. Phys.*, **98**, 5612(1993).
- [21] A. D. Becke, *J. Chem. Phys.*, **98**, 5648(1993).
- [22] C. P. Sosa and N. Carriero, CUG Spring 1997 Proceedings, B. Winget, Ed., Fine Point Editorial Services, San Jose, CA, 1997.
- [23] J. B. Foresman, M. Head-Gordon, J. A. Pople, and M. J. Frisch, *J. Phys. Chem.* **96**, 135(1992).
- [24] To obtain all the input files used in this study send an e-mail message to: cpsosa@msi.umn.edu
- [25] C. P. Sosa and M. J. Frisch, in preparation.
- [26] M. Head-Gordon, J. A. Pople, and M. J. Frisch, *Chem. Phys. Lett.*, **153**, 503(1990).
- [27] M. J. Frisch, M. Head-Gordon, and J. A. Pople, *Chem. Phys. Lett.*, **166**, 275(1990).
- [28] M. J. Frisch, M. Head-Gordon, and J. A. Pople, *Chem. Phys. Lett.*, **166**, 281(1990).

Special Notices

This document was produced in the United States. IBM may not offer the products, programs, services or features discussed herein in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the products, programs, services, and features available in your area. Any

reference to an IBM product, program, service or feature is not intended to state or imply that only IBM's product, program, service or feature may be used. Any functionally equivalent product, program, service or feature that does not infringe on any of IBM's intellectual property rights may be used instead of the IBM product, program, service or feature. IBM makes no representation or warranty regarding third-party products or services.

Information in this document concerning non-IBM products was obtained from the suppliers of these products, published announcement material or other publicly available sources. Sources for non-IBM list prices and performance numbers are taken from publicly available information including D.H. Brown, vendor announcements, vendor WWW Home Pages, SPEC Home Page, GPC (Graphics Processing Council) Home Page and TPC (Transaction Processing Performance Council) Home Page. IBM has not tested these products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquiries, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of a specific Statement of General Direction.

The information contained in this document has not been submitted to any formal IBM test and is distributed "AS IS". While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. The use of this information or the implementation of any techniques described herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. Customers attempting to adapt these techniques to their own environments do so at their own risk.

IBM is not responsible for printing errors in this publication that result in pricing or information inaccuracies.

The information contained in this document represents the current views of IBM on the issues discussed as of the date of publication. IBM cannot guarantee the accuracy of any information presented after the date of publication.

All prices shown are IBM's suggested list prices; dealer prices may vary.

IBM products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Information provided in this document and information contained on IBM's past and present Year 2000 Internet Web site pages regarding products and services offered by IBM and its subsidiaries are "Year 2000 Readiness Disclosures" under the Year 2000 Information and Readiness Disclosure Act of 1998, a U.S. statute enacted on October 19, 1998. IBM's Year 2000 Internet Web site pages have been and will continue to be our primary mechanism for communicating year 2000 information. Please see the "legal" icon on IBM's Year 2000 Web site (<http://www.ibm.com/year2000>) for further information regarding this statute and its applicability to IBM.

Any performance data contained in this document was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM, e(logo), pSeries, RS/6000

IBM Trademarks information can be found at: <http://www.ibm.com/legal/copytrade.shtml>.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SPEC, SPECjbb, SPECint, SPECfp, SPECweb, and SPECsfs are trademarks of the Standard Performance Evaluation Corporation and information can be found at: <http://www.spec.org>.

TPC, TPC-R, TPC-H, TPC-C, and TPC-W are trademarks of the Transaction Processing Performance Council. Information can be found at: <http://www.tpc.org>.

Other trademarks are the property of their respective owners.

Notes on Benchmarks and Values

The benchmarks and values shown here were derived using particular, well configured, development-level computer systems. Unless otherwise indicated for a system, the values were derived using 32-bit applications and external cache, if external cache is supported on the system. All benchmark values are provided "AS IS" and no warranties or guarantees are expressed or implied by IBM. Actual system performance may vary and is dependent upon many factors including system hardware configuration and software design and configuration. Buyers should consult other sources of information to evaluate the performance of systems they are considering buying and should consider conducting application oriented testing. For additional information about the benchmarks, values and systems tested, contact your local IBM office or IBM authorized reseller or access the following on the Web:

TPC	http://www.tpc.org
GPC	http://www.spec.org/gpc
SPEC	http://www.spec.org
Pro/E	http://www.proe.com
Linpack	http://www.netlib.no/netlib/benchmark/performance.ps
Notesbench Mail	http://www.notesbench.org
VolanoMark	http://www.volano.com
Fluent	http://www.fluent.com
Gaussian	http://www.gaussian.com

Unless otherwise indicated for a system, the performance benchmarks were conducted using AIX V4.2.1 or 4.3, IBM C Set++ for AIX/6000 V4.1.0.1, and AIX XL FORTRAN V5.1.0.0 with optimization where the compilers were used in the benchmark tests. The preprocessors used in the benchmark tests include KAP 3.2 for FORTRAN and KAP/C 1.4.2 from Kuck & Associates and VAST-2 v4.01X8 from Pacific-Sierra Research. The preprocessors were purchased separately from these vendors.

The following SPEC and Linpack benchmarks reflect the performance of the microprocessor, memory architecture, and compiler of the tested system:

- SPECint95 - SPEC component-level benchmark that measures integer performance. Result is the geometric mean of eight tests that comprise the CINT95 benchmark suite. All of these are written in the C language. SPECint_base95 is the result of the same tests as CINT95 with a maximum of four compiler flags that must be used in all eight tests.
- SPECint_rate95 - Geometric average of the eight SPEC rates from the SPEC integer tests (CINT95). SPECint_base_rate95 is the result of the same tests as CINT95 with a maximum of four compiler flags that must be used in all eight tests.
- SPECfp95 - SPEC component-level benchmark that measures floating-point performance. Result is the geometric mean of ten tests, all written in FORTRAN, that are included in the CFP95 benchmark suite. SPECfp_base95 is the result of the same tests as CFP95 with a maximum of four compiler flags that must be used in all ten tests.
- SPECfp_rate95 - Geometric average of the ten SPEC rates from SPEC floating-point tests (CFP95). SPECfp_base_rate95 is the result of the same tests as CFP95 with a maximum of four compiler flags that must be used in all ten tests.
- SPECint2000 - New SPEC component-level benchmark that measures integer performance. Result is the geometric mean of twelve tests that comprise the CINT2000 benchmark suite. All of these are written in C language except for one which is in C++. SPECint_base2000 is the result of the same tests as CINT2000 with a maximum of four compiler options that must be used in all twelve tests.

- SPECint_rate2000 - Geometric average of the twelve SPEC rates from the SPEC integer tests (CINT2000).
SPECint_base_rate2000 is the result of the same tests as CINT2000 with a maximum of four compiler options that must be used in all twelve tests.
- SPECfp2000 - New SPEC component-level benchmark that measures floating-point performance. Result is the geometric mean of fourteen tests, all written in FORTRAN and C languages, that are included in the CFP2000 benchmark suite.
SPECfp_base2000 is the result of the same tests as CFP2000 with a maximum of four compiler options that must be used in all fourteen tests.
- SPECfp_rate2000 - Geometric average of the fourteen SPEC rates from SPEC floating-point tests (CFP2000).
SPEC_base_rate2000 is the result of the same tests as CFP2000 with a maximum of four compiler options that must be used in all fourteen tests.
- SPECweb96 - Maximum number of Hypertext Transfer Protocol (HTTP) operations per second achieved on the SPECweb96 benchmark without significant degradation of response time. The Web server software is ZEUS v.1.1 from Zeus Technology Ltd.
- SPECweb99 - Number of conforming, simultaneous connections the Web server can support using a predefined workload. The SPECweb99 test harness emulates clients sending the HTTP requests in the workload over slow Internet connections to the Web server. The Web server software is Zeus from Zeus Technology Ltd.
- LINPACK DP (Double Precision) - n=100 is the array size. The results are measured in megaflops (MFLOPS).
- LINPACK SP (Single Precision) - n=100 is the array size. The results are measured in MFLOPS.
- LINPACK TPP (Toward Peak Performance) - n=1,000 is the array size. The results are measured in MFLOPS.
- LINPACK HPC (Highly Parallel Computing) - solve largest system of linear equations possible. The results are measured in GFLOPS.

VolanoMark is a 100% Pure Java™ server benchmark characterized by long-lasting network connections and high thread counts. In this context, long-lasting means the connections last several minutes or longer, rather than just a few seconds. The VolanoMark benchmark creates client connections in groups of 20 and measures how long it takes for the clients to take turns broadcasting their messages to the group. At the end of the test, it reports a score as the average number of messages transferred by the server per second.

VolanoMark 2.1.2 local performance test measures throughput in messages per second. The final score is the average of the best two out of three results.

The following SPEC benchmark reflects the performance of the microprocessor, memory subsystem, disk subsystem, network subsystem:

- SPECsfs97_R1 - the SPECsfs97_R1 (or SPEC SFS 3.0) benchmark consists of two separate workloads, one for NFS V2 and one for NFS V3, which report two distinct metrics, SPECsfs97_R1.v2 and SPECsfs97_R1.v3, respectively. The metrics consist of a throughput component and an overall response time measure. The throughput (measured in operations per second) is the primary component used when comparing SFS performance between systems. The overall response time (average response time per operation) is a measure of how quickly the server responds to NFS operation requests over the range of tested throughput loads.

The following Transaction Processing Performance Council (TPC) benchmarks reflect the performance of the microprocessor, memory subsystem, disk subsystem, and some portions of the network:

- tpmC - TPC Benchmark C throughput measured as the average number of transactions processed per minute during a valid TPC-C configuration run of at least twenty minutes.
- \$/tpmC - TPC Benchmark C price/performance ratio reflects the estimated five year total cost of ownership for system hardware, software, and maintenance and is determined by dividing such estimated total cost by the tpmC for the system.
- QppH is the power metric of TPC-H and is based on a geometric mean of the 17 TPC-H queries, the insert test, and the delete test. It measures the ability of the system to give a single user the best possible response time by harnessing all available resources. QppH is scaled based on database size from 30GB to 1TB.
- QthH is the throughput metric of TPC-H and is a classical throughput measurement characterizing the ability of the system to support a multiuser workload in a balanced way. A number of query users is chosen, each of which must execute the full set of 17 queries in a different order. In the background, there is an update stream running a series of insert/delete operations. QthH is scaled based on the database size from 30GB to 1TB.

- $\$/\text{QphH}$ is the price/performance metric for the TPC-H benchmark where QphD is the geometric mean of QppH and QthH. The price is the five-year cost of ownership for the tested configuration and includes maintenance and software support.

The following graphics benchmarks reflect the performance of the microprocessor, memory subsystem, and graphics adapter:

- SPECxpc results - Xmark93 is the weighted geometric mean of 447 tests executed in the x11perf suite and is an indicator of 2D graphics performance in an X environment. Larger values indicate better performance.
- SPECplb results (graPHIGS) - PLBwire93 and PLBsurf93 are geometric means of literal and optimized Picture Level Benchmark (PLB) tests for 3D wireframe and 3D surface tests, respectively. The benchmark and tests were developed by the Graphics Performance Characterization (GPC) Committee. The results shown used the graPHIGS API. Larger values indicate better performance.
- SPECopc results - CDRS-03, CDRS-04, DX-03, DX-04, DX-05, DRV-04, DRV-05, DRV-06, Light-01, Light-02, Light-02, AWadvs-01, AWadvs-02, AWadvs-03, and ProCDRS-02 are weighted geometric means of individual viewset metrics. The viewsets were developed by ISVs (independent software vendors) with the assistance of OPC (OpenGL Performance Characterization) member companies. Larger values indicate better performance.

The following graphics benchmarks reflect the performance of the microprocessor, memory subsystem, graphics adapter, and disk subsystem:

Bench95 and Bench97 Pro/E results - Bench95 and Bench97 Pro/E benchmarks have been developed by Texas Instruments to measure UNIX and Windows NT[®] workstations in a comparable real-world environment. Results shown are in minutes. Lower numbers indicate better performance.

The NotesBench Mail workload simulates users reading and sending mail. A simulated user will execute a prescribed set of functions 4 times per hour and will generate mail traffic about every 90 minutes. Performance metrics are:

- NotesMark - transactions/minute (TPM).
- NotesBench users - number of client (user) sessions being simulated by the NotesBench workload.
- $\$/\text{NotesMark}$ - ratio of total system cost divided by the NotesMark (TPM) achieved on the Mail workload.
- $\$/\text{User}$ - ratio of total system cost divided by the number of client sessions successfully simulated for the Mail NotesBench workload measured.

Total system cost is the price of the server under test to the customer, including hardware, operating system, and Domino Server licenses.