CFP2000 Result

Copyright ©1999-2004, Standard Performance Evaluation Corporation

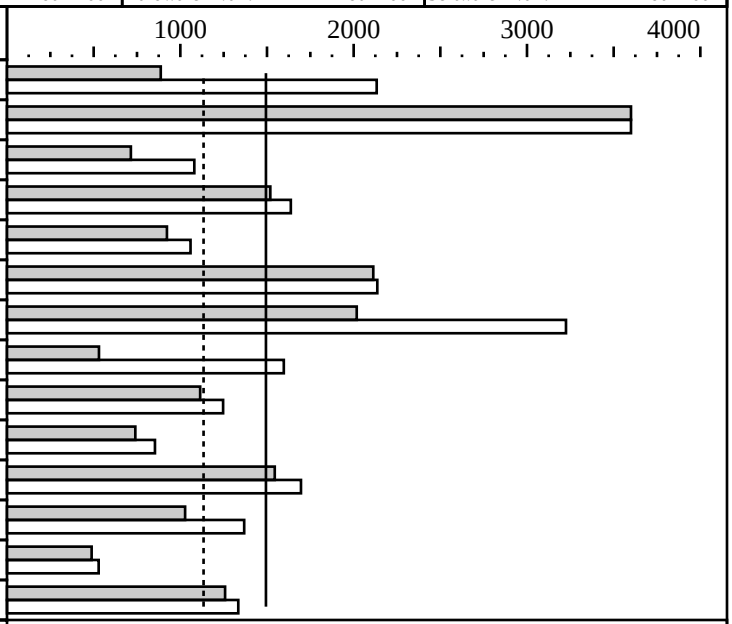
Hewlett-Packard Company
AlphaServer ES80 7/1150

SPECfp2000 = 1494

SPECfp_base2000 = 1134

SPEC license #: 2 | Tested by: HP | Test date: Jun-2004 | Hardware Avail: Jul-2004 | Software Avail: Jul-2004

Benchmark	Reference Time	Base Runtime	Base Ratio	Runtime	Ratio
168.wupwise	1600	180	887	75.0	2133
171.swim	3100	86.1	3600	86.1	3600
172.mgrid	1800	252	715	167	1081
173.applu	2100	138	1519	128	1637
177.mesa	1400	152	923	132	1059
178.galgel	2900	137	2113	136	2136
179.art	2600	129	2018	80.6	3225
183.equake	1300	245	531	81.4	1597
187.facerec	1900	170	1114	152	1247
188.ammp	2200	297	740	258	854
189.lucas	2000	129	1545	118	1696
191.fma3d	2100	204	1028	153	1368
200.sixtrack	1100	225	488	208	529
301.apsi	2600	207	1258	195	1334



Hardware

CPU:	Alpha 21364
CPU MHz:	1150
FPU:	Integrated
CPU(s) enabled:	1 core, 1 chip, 1 core/chip
CPU(s) orderable:	2 to 8
Parallel:	No
Primary Cache:	64KB(I)+64KB(D) on chip
Secondary Cache:	1.75MB on chip per CPU
L3 Cache:	None
Other Cache:	None
Memory:	4GB per CPU; 512MB RIMMs
Disk Subsystem:	AdvFS
Other Hardware:	None

Software

Operating System:	Tru64 UNIX V5.1B + IPK
Compiler:	Compaq C V6.5-011-48C5K Program Analysis Tools V2.0 Spike V5.2 (510 USG) HP Fortran V5.5A-3548-48D88 HP Fortran 77 V5.5A-3548-48D88 KAP Fortran V4.3 000607 KAP Fortran 77 V4.1 980926 KAP C V4.1 000607
File System:	MFS, 8GB
System State:	Multi-user

Notes/Tuning Information

```
Baseline   C: cc  -arch ev7 -fast -O4 ONESTEP
     Fortran: f90 -arch ev7 -fast -O5 ONESTEP


Peak:
  All use -g3 -arch ev7 -non_shared ONESTEP
  except these (which use only the tunings shown below):
     173.applu 188.ammp 191.fma3d
  Individual benchmark tuning:
  168.wupwise: kf77 -call_shared -inline all -tune ev67
               -unroll 12 -automatic -align commons -arch ev67
               -fkapargs=' -aggressive=c -fuse
               -fuselevel=1 -so=2 -r=1 -o=1 -interleave
               -ur=6 -ur2=060 ' +PFB
      171.swim: same as base
     172.mgrid: kf90 -call_shared -arch generic -O5 -inline
                manual -nopipeline -transform_loops -unroll 9 -automatic
```

Notes/Tuning Information (Continued)

```
                  -fkapargs='-aggressive=a -fuse -interleave
                  -ur=2 -ur3=5 -cachesize=128,16000 ' +PFB
     173.applu: kf90  -O5 -transform_loops
                -fkapargs=' -o=0 -nointerleave -ur=14
                -ur2=260 -ur3=18' +PFB
      177.mesa: kcc -fast -O4 +CFB +IFB
    178.galgel: f90 -O5 -fast -unroll 5 -automatic
       179.art: kcc  -assume whole_program -ldensemalloc
                -call_shared -assume restricted_pointers
                -unroll 16 -inline none -ckapargs='
                -fuse -fuselevel=1 -ur=3' +PFB
    183.equake: cc -call_shared -arch generic -fast -O4
                -ldensemalloc -assume restricted_pointers
                -inline speed -unroll 13 -xtaso_short +PFB
   187.facerec: f90 -O4 -nopipeline -inline all
                -non_shared -speculate all -unroll 7
                -automatic -assume accuracy_sensitive
                -math_library fast +IFB
      188.ammp: cc -arch host -O4 -ifo -assume nomath_errno
                -assume trusted_short_alignment -fp_reorder
                -readonly_strings -ldensemalloc -xtaso_short
                -assume restricted_pointers -unroll 9
                -inline speed +CFB +IFB +PFB
     189.lucas: kf90 -O5 -fkapargs='-ur=1' +PFB
     191.fma3d: kf90 -O4 -transform_loops -fkapargs='-cachesize=128,16000 ' +PFB
  200.sixtrack: f90 -fast -O5 -assume accuracy_sensitive
                -notransform_loops +PFB
      301.apsi: kf90 -O5 -inline none -call_shared -speculate all
                -align commons -fkapargs=' -aggressive=ab
                -tune=ev5 -fuse -ur=1 -ur2=60 -ur3=20
                -cachesize=128,16000'

Most benchmarks are built using one or more types of
profile-driven feedback.  The types used are designated
by abbreviations in the notes:

+CFB: Code generation is optimized by the compiler, using
      feedback from a training run.  These commands are
      done before the first compile (in phase "fdo_pre0"):

            mkdir /tmp/pp
            rm -f /tmp/pp/${baseexe}*

      and these flags are added to the first and second compiles:

            PASS1_CFLAGS = -prof_gen_noopt -prof_dir /tmp/pp
            PASS2_CFLAGS = -prof_use_feedback -prof_dir /tmp/pp

     (Peak builds use /tmp/pp above; base builds use /tmp/pb.)

+IFB: Icache usage is improved by the post-link-time optimizer
      Spike, using feedback from a training run.  These commands
      are used (in phase "fdo_postN"):

            mv ${baseexe} oldexe
            spike oldexe -feedback oldexe -o ${baseexe}
```

Copyright ©1999-2004, Standard Performance Evaluation Corporation

Notes/Tuning Information (Continued)

```
+PFB: Prefetches are improved by the post-link-time optimizer
      Spike, using feedback from a training run.  These
      commands are used (in phase "fdo_post_makeN"):

           rm -f *Counts*
           mv ${baseexe} oldexe
           pixie -stats dstride oldexe 1>pixie.out 2>pixie.err
           mv oldexe.pixie ${baseexe}

      A training run is carried out (in phase "fdo_runN"), and
      then this command (in phase "fdo_postN"):

           spike oldexe -fb oldexe -stride_prefetch -o ${baseexe}

When Spike is used for both Icache and Prefetch improvements,
only one spike command is actually issued, with the Icache
options followed by the Prefetch options.

vm:
        vm_bigpg_enabled = 1
        vm_bigpg_thresh = 6
        vm_swap_eager = 0
        ubc_maxpercent = 50

proc:
        max_per_proc_address_space = 34359738368
        max_per_proc_data_size = 34359738368
        max_per_proc_stack_size = 34359738368
        max_proc_per_user = 2048
        max_threads_per_user = 4096
        maxusers = 2048
        per_proc_address_space = 34359738368
        per_proc_data_size = 34359738368
        per_proc_stack_size = 34359738368


Portability: galgel: -fixed

Information on UNIX V5.1B Patches can be found at
http://ftp1.service.digital.com/public/unix/v5.1b/

Processes were bound to CPUs using "runon".

This result was measured on model ES80.
Model ES47 and model ES80 are electronically equivalent.
```