## CINT2000 Result

**Hewlett-Packard Company**  
**hp AlphaServer DS25 68/1000**  

**SPECint_rate2000 = 15.5**  
**SPECint_rate_base2000 = 14.3**

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Base Copies</th>
<th>Base Runtime</th>
<th>Base Ratio</th>
<th>Copies</th>
<th>Runtime</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>164.gzip</td>
<td>2</td>
<td>303</td>
<td>10.7</td>
<td>2</td>
<td>299</td>
<td>10.9</td>
</tr>
<tr>
<td>175.vpr</td>
<td>2</td>
<td>266</td>
<td>12.2</td>
<td>2</td>
<td>263</td>
<td>12.3</td>
</tr>
<tr>
<td>176.gcc</td>
<td>2</td>
<td>158</td>
<td>16.2</td>
<td>2</td>
<td>142</td>
<td>18.0</td>
</tr>
<tr>
<td>181.mcf</td>
<td>2</td>
<td>327</td>
<td>12.8</td>
<td>2</td>
<td>250</td>
<td>16.7</td>
</tr>
<tr>
<td>186.crafty</td>
<td>2</td>
<td>123</td>
<td>18.9</td>
<td>2</td>
<td>123</td>
<td>18.9</td>
</tr>
<tr>
<td>197.parser</td>
<td>2</td>
<td>433</td>
<td>9.65</td>
<td>2</td>
<td>346</td>
<td>12.1</td>
</tr>
<tr>
<td>252.eon</td>
<td>2</td>
<td>164</td>
<td>18.4</td>
<td>2</td>
<td>159</td>
<td>19.0</td>
</tr>
<tr>
<td>253.perlbmk</td>
<td>2</td>
<td>313</td>
<td>13.4</td>
<td>2</td>
<td>295</td>
<td>14.2</td>
</tr>
<tr>
<td>254.gap</td>
<td>2</td>
<td>245</td>
<td>10.4</td>
<td>2</td>
<td>207</td>
<td>12.3</td>
</tr>
<tr>
<td>255.vortex</td>
<td>2</td>
<td>224</td>
<td>19.7</td>
<td>2</td>
<td>210</td>
<td>21.0</td>
</tr>
<tr>
<td>256.bzip2</td>
<td>2</td>
<td>226</td>
<td>15.4</td>
<td>2</td>
<td>212</td>
<td>16.4</td>
</tr>
<tr>
<td>300.twolf</td>
<td>2</td>
<td>379</td>
<td>18.4</td>
<td>2</td>
<td>374</td>
<td>18.6</td>
</tr>
</tbody>
</table>

### Notes/Tuning Information

**Baseline C**: cc -arch ev6 -fast +CFB ONESTEP  
**C++**: cxx -arch ev6 -O2 ONESTEP

**Peak**:  
All but 252.eon: cc -g3 -arch ev6 ONESTEP  
164.gzip: -fast -O4 -non_shared +CFB  
175.vpr: -fast -O4 -assume restricted_pointers +CFB  
176.gcc: -fast -O4 -xtaso_short -all -ldensemalloc -none +CFB +IFB  
181.mcf: -fast -xtaso_short +CFB +IFB +PFB  
186.crafty: same as base  
197.parser: -fast -O4 -xtaso_short -non_shared +CFB  
252.eon: cxx -arch ev6 -O2 -all -ldensemalloc -none  
253.perlbmk: -fast -non_shared +CFB +IFB  
254.gap: -fast -O4 -non_shared +CFB +IFB +PFB  
255.vortex: -fast -non_shared +CFB +IFB  
256.bzip2: -fast -O4 -non_shared +CFB  
300.twolf: -fast -O4 -assume restricted_pointers -all -ldensemalloc -none +CFB +IFB

**Hardware**
- CPU: Alpha 21264C  
- CPU MHz: 1000  
- FPU: Integrated  
- CPU(s) enabled: 2 cores, 2 chips, 1 core/chip  
- CPU(s) orderable: 1 to 2  
- Parallel: No  
- Primary Cache: 64KB(I)+64KB(D) on chip  
- Secondary Cache: 8MB off chip per CPU  
- L3 Cache: None  
- Other Cache: None  
- Memory: 8GB  
- Disk Subsystem: 18.2GB SCSI  
- Other Hardware: None

**Software**
- Operating System: Tru64 UNIX V5.1A  
- Compiler: Compaq C V6.4-215-46B7O  
- Program Analysis Tools V2.0  
- Spike V5.2 DTK (1.471.2.2 46B5P)  
- Compaq C++ V6.3-010-46B2F  
- File System: AdvFS  
- System State: Multi-user
Hewlett-Packard Company
hp AlphaServer DS25 68/1000

SPECint_rate2000 = 15.5
SPECint_rate_base2000 = 14.3

Notes/Tuning Information (Continued)

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 -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}
```

+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.


Spike, and the Program Analysis Tools, are part of the Developers' Tool Kit Supplement, http://www.tru64unix.compaq.com/dtk/ . The features used in this SPEC submission will be available at the website as a production release in October, 2001. The C compiler for this SPEC submission has been available at the same location, as a production release, since August, 2001.