

6.6.1.2 IPC Bandwidth

Imbench addresses the performance issues of interprocess communication bandwidth with the pipe bandwidth and TCP bandwidth micro-benchmarks. UNIX pipes are an interprocess communication that provides a one-way flow of data. A pipe is created by the pipe system call. Two file descriptors are returned *-filedes[0]* which is open for reading, and *filedes[1]* which is open for writing. TCP sockets are similar to pipes except they are bi-directional and can cross machine boundaries because they are a networking facility. The benchmarks included in the *IPC bandwidth* component are: *bw_pipe* and *bw_tcp*.

Pipe bandwidth is measured by the benchmark *bw_pipe* (Table 8). *bw_pipe* creates a UNIX pipe between two processes, a writer and a reader, which transfers 50M bytes through the pipe in 64K bytes chunks.

TCP bandwidth is measured similarly to the pipe bandwidth measurements, except that the default amount of data sent is 3M bytes (Table 9). *bw_tcp*, the micro-benchmark, is a client/server program that moves data over a TCP/IP socket. The sockets are configured to use the largest receive/send buffers that the OS will allow.

System	pipe (bw_pipe)
HP K210	93
Linux/i686	89
IBM Power 2	84
Linux/Alpha	73
Unixware/i686	68
Sun Ultra 1	61
DEC Alpha@300	46
Solaris/i686	38
DEC Alpha @ 150	35
SGI Indigo 2	34
Linux/i586	34
IBM Power PC	30
FreeBSD/i586	23
SGI Challenge	17
Sun SC1000	9
DAISY systems	
FreeBSD/i586 (p5-90)	10.72
HEAT systems	
DEC Alpha	41.06
HP 9000/735	30.37
IBM RS6000	16.91
SUN SS10	17.04
SGI IRIX	15.67

Table 8. bw_pipe results (MB/s).

System	Network	TCP (bw_tcp)	
		local host	remote host
SGI PowerChallenge	hppi	na	79.3
Sun Ultra 1	100baseT	na	9.5
HP 9000/735	fddi	na	8.8
FreeBSD/i586	100baseT	na	7.9
SGI Indigo 2	10baseT	na	0.9
HP 9000/735	10baseT	na	0.9
Linux/i586@90Mhz	10baseT	na	0.7
DAISY systems			
FreeBSD/i586 (p5-90)	10base2	0.21	0.76
FreeBSD/i586 (p5-90)	100baseT	5.96	6.26
HEAT systems			
DEC Alpha	fddi	13.1	9.76
HP 9000/735	fddi	29.06	9.02
IBM RS6000	fddi	2.91	4.54
SUN SS10	fddi	4.68	0.76
SGI IRIX	fddi	12.98	1.35

Table 9. bw_tcp results (MB/s). na = not available

6.6.1.3 Cached I/O Bandwidth

The reusing of data in the file system page cache can be a performance issue. Imbench addresses the cached I/O bandwidth problem with the bw_file_rd and the bw_mmap_rd micro-benchmarks. The benchmarks results in Table 10 are not disk read measurements, they are memory read measurements.

The *read* (bw_file_rd) benchmark is implemented by rereading a file (8M bytes in the results) in 64K buffers. Basically the bw_file_rd benchmark measures the speed at which cached file pages can be reused.

System	file read (bw_file_rd)	file mmap (bw_mmap_rd)
IBM Power 2	187	106
HP K210	88	52
Sun Ultra 1	85	101
DEC Alpha@300	67	78
Unixware/i686	62	200
Solaris/i686	52	94
DEC Alpha@150	40	50
Linux/i686	40	36
IBM PowerPC	40	51
SGI Challenge	36	56
SGI Indigo 2	32	44
FreeBSD/i586	30	53
Linux/Alpha	24	18
Linux/i586	23	9
Sun SC1000	20	28
DAISY systems		
FreeBSD/i586 (p5-90)	18.28	36.75
HEAT systems		
DEC Alpha	42.47	55.01
HP 9000/735	33.71	35.76
IBM RS6000	32.45	36.23
SUN SS10	36.62	21.18
SGI IRIX	19.6	26.06

Table 10. bw_file_rd and bw_mmap_rd results (MB/s).

The *mmap* (bw_mmap_rd) benchmark provides a way to access the kernel's file cache without copying the data. bw_mmap_rd is implemented by mapping the entire file into the process's address space. The file is then summed to force the data into the cache.

6.6.2 Latency Measurements

6.6.2.1 Memory Latency

In the memory read latency benchmark (lat_mem_rd) the entire memory hierarchy is measured (as shown in Figure 16 for DAISy and HEAT), including on-board data cache latency and size, external data cache latency and size, and main memory latency. The benchmark varies two parameters, array size and array stride. For each size, a list of pointers is created for all the different strides. Then the list is walked thus:

```
mov r0,(r0) # C code: p = *p;
```

The time to do about 1,000,000 loads is measured and reported. The appendices shows the results of all strides and array sizes, varying 8 - 32768 for strides, and 1K up to 8M bytes in array size. The curves contain a series of horizontal plateaus, where each plateau represents a level in the memory hierarchy. The results in Table 11 show the memory latency for main memory with a stride of 8192Mbytes for DAISy and HEAT.

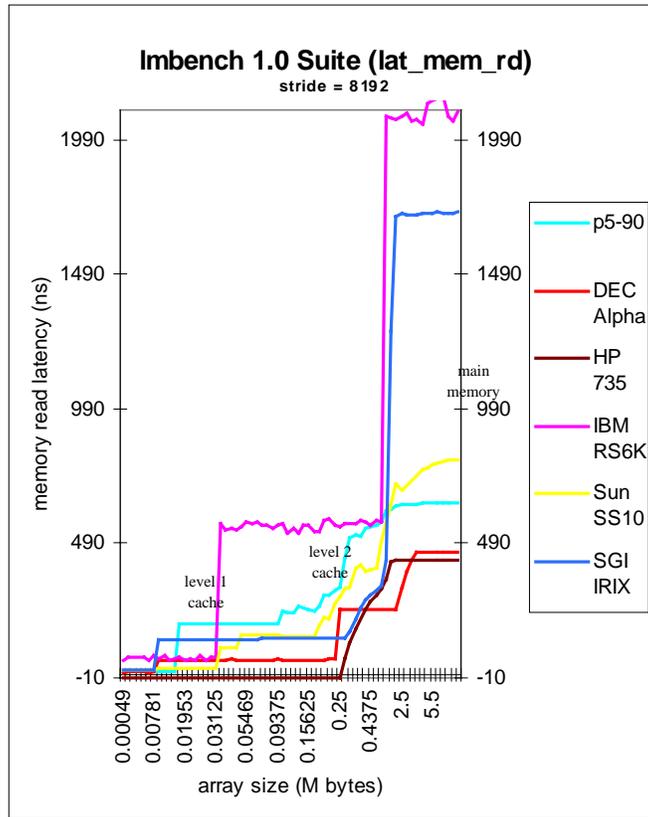


Figure 16. DAISy and HEAT lat_mem_rd results.

System	memory latency (lat_mem_rd)
Linux/i586	150
Unixware/i686	175
Linux/i686	179
FreeBSD/i586	182
IBM Power 2	260
Sun Ultra 1	270
Solaris/i686	281
DEC Alpha@150	291
HP K210	349
Linux/Alpha	357
IBM PowerPC	394
DEC Alpha@300	400
SGI Indigo 2	1170
SGI Challenge	1189
Sun SC1000	1236
DAISY systems	
FreeBSD/i586 (p5-90)	645
HEAT systems	
DEC Alpha	458
HP 9000/735	430
IBM RS6000	2146
SUN SS10	801
SGI IRIX	1723

Table 11. lat_mem_rd results with .stride of 8192Mbytes for DAISy and HEAT (nanoseconds).