



***Parallel Matlab:  
RTExpress on 64-bit SGI Altix  
with SCSL and MPT***

**Cosmo Castellano**  
[castellano@sensors.com](mailto:castellano@sensors.com)

**Integrated Sensors, Inc**

**502 Court Street**

**Suite 210**

**Utica, NY 13502**

**(315) 798-1377**

**[www.sensors.com](http://www.sensors.com)**

# Matlab FFT Benchmark Test on Shared Memory SGI

---

- A Matlab script performs the 2D complex FFT

```
matrix = ones(fftsize, fftsize) + j * ones(fftsize, fftsize)
```

```
loop
```

```
    store time t1
```

```
    a = fft(init_matrix)
```

```
    store time t2
```

```
    a = a'
```

```
    store time t3
```

```
    a = fft(a)
```

```
    store time t4
```

```
end loop
```

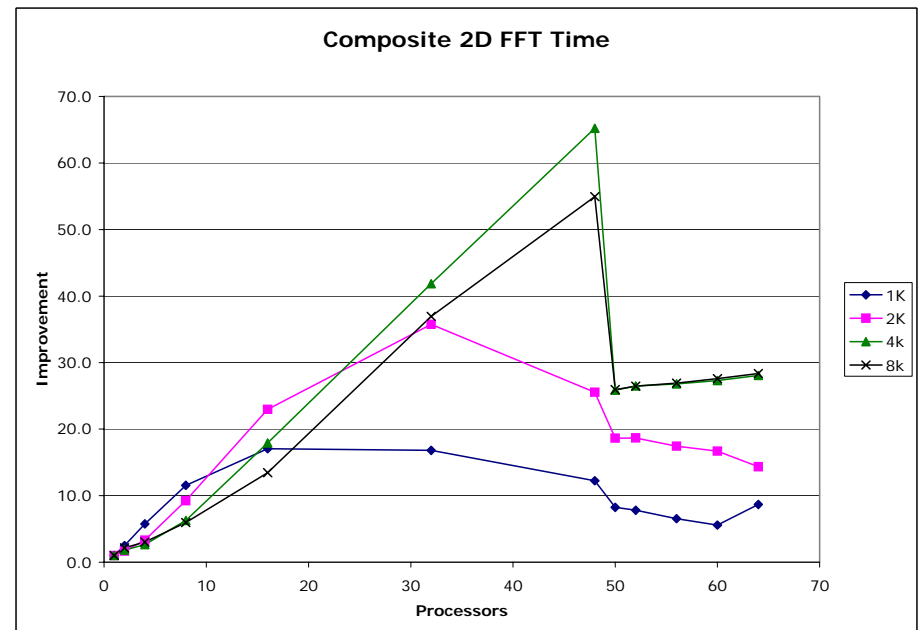
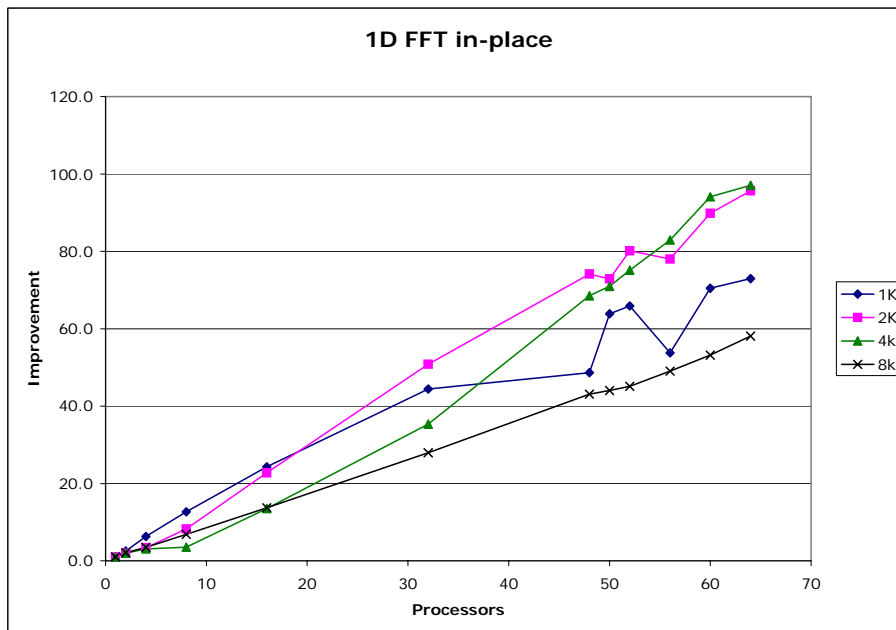
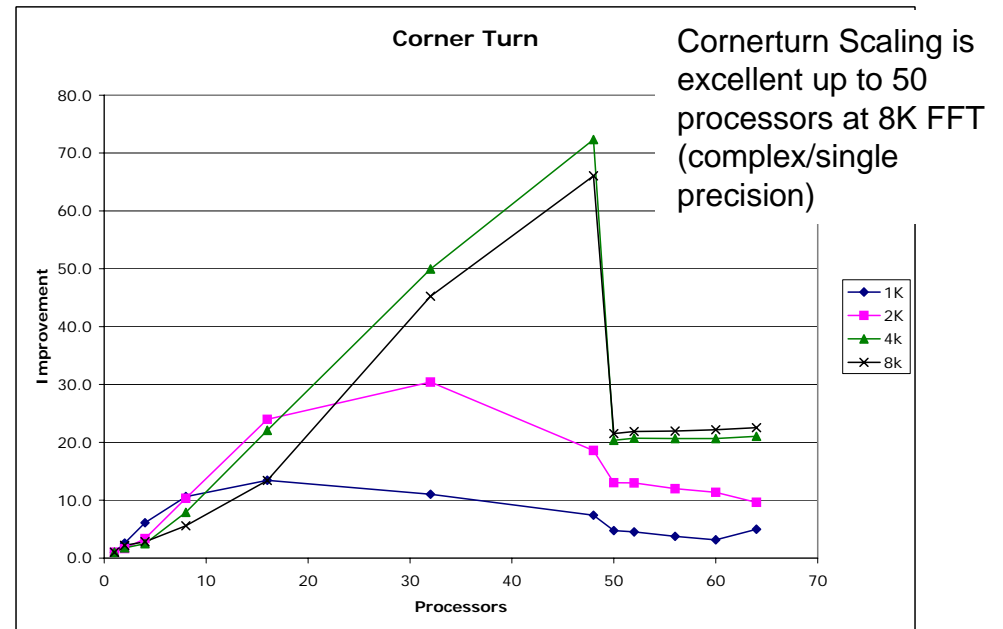
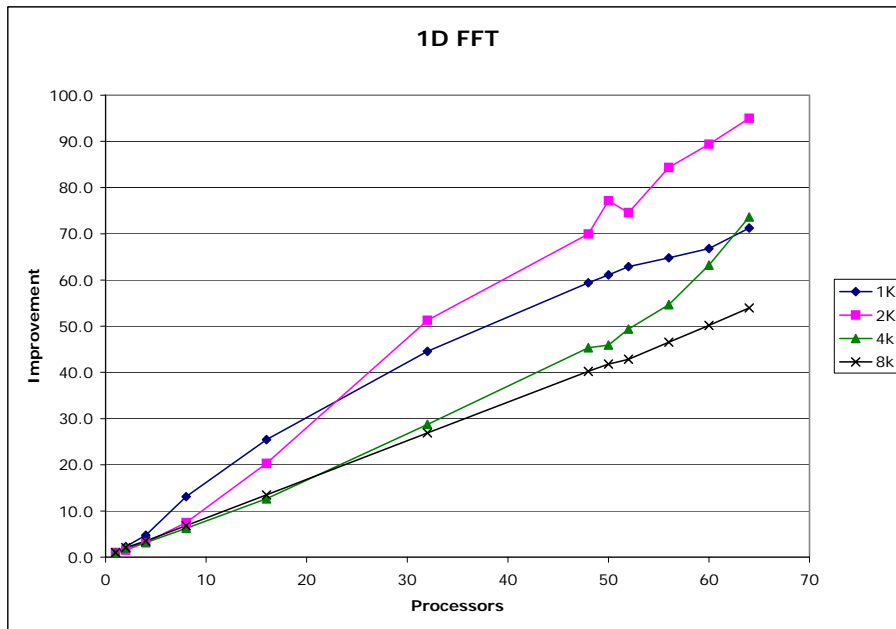
fft1 elapsed time =  $t2 - t1$

ctm elapsed time =  $t3 - t2$

fft2 elapsed time =  $t4 - t3$

- RTExpress is used to compile and run the MATLAB script on the 64-bit parallel computer on varying numbers of processors in data-parallel
- Elapsed times are computed, averaging time over several iterations
- First iteration is not counted

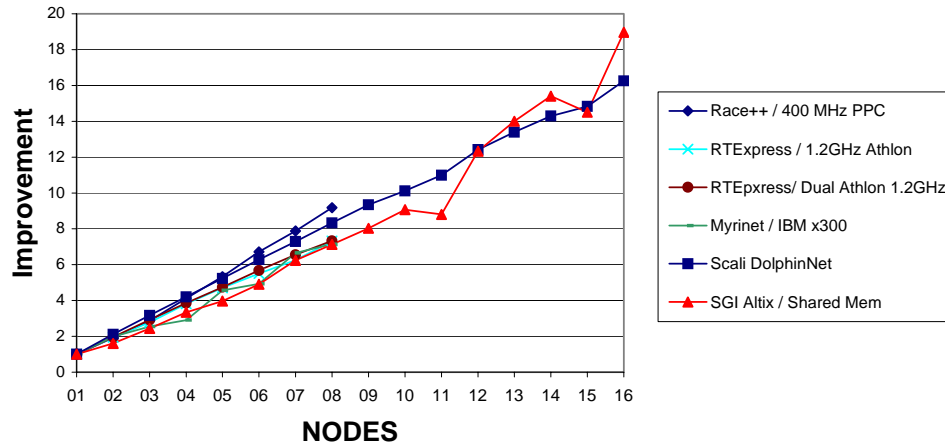
# Scaling: 64-Processor Altix



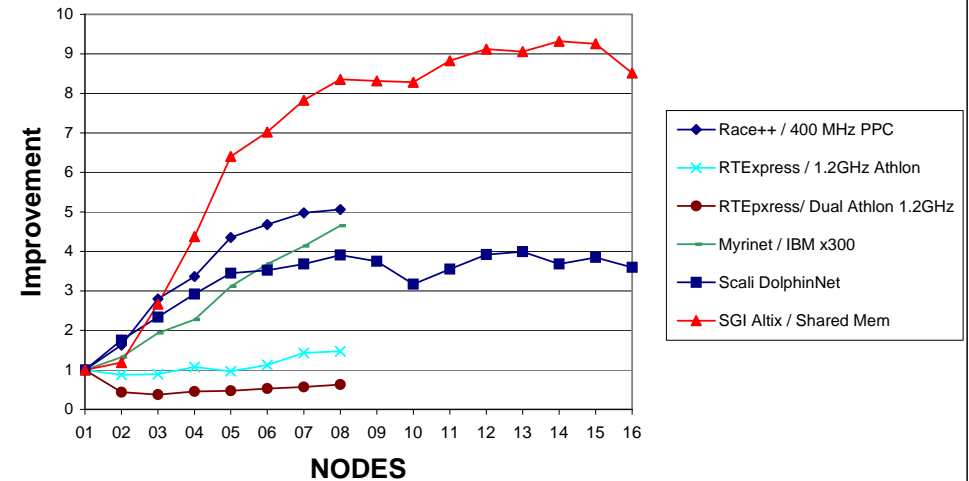
64p 1.7GHz/9MB cache Altix 3700 (note: production systems are 1.6GHz)

# SGI Shared Memory Improves Cornerturn and 2D FFT

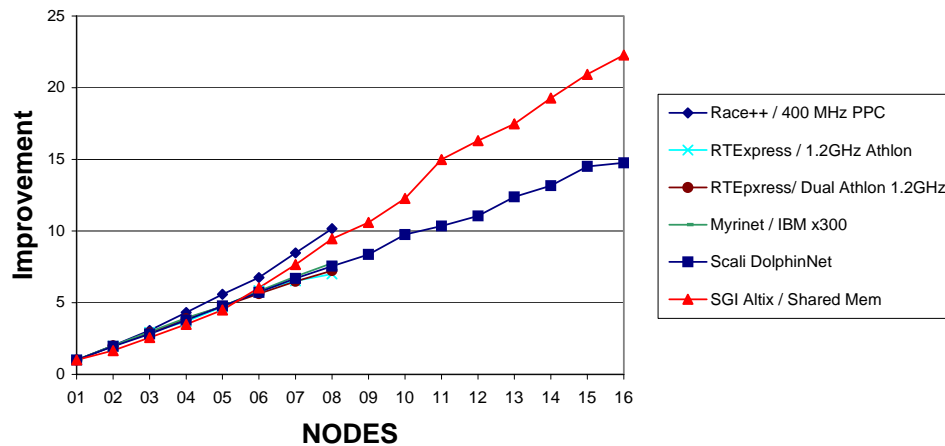
**FFT Timing  
(1024 x 1024 out of place)**



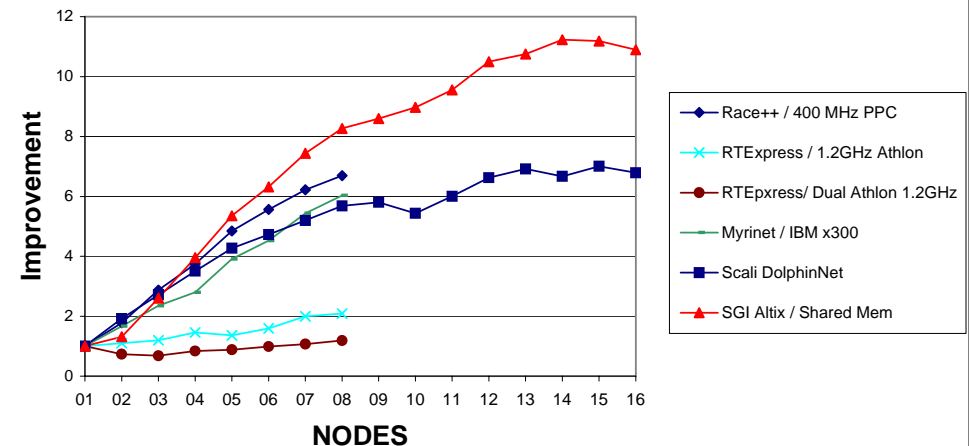
**1024 x 1024 Cornerturn Timing**



**FFT Timing  
(1024 x 1024 in-place)**



**2D FFT Timing  
(1024 x 1024)**



Altix350 1.4 Ghz/3MB L3 cache

# Timing Information NOTES

---

- Please note that all timing information gathered is not intended to provide a recommendation for any particular hardware, but to illustrate parallel operation with various combinations of processors and interconnect systems
- Equipment used in the following tests may no longer be the hardware vendor's current offering
- “Improvement,” or speed-up, as compared to first-processor performance used to examine scaling rather than absolute timing
- Maximum RTExpress performance may be gained by fully using vector operations in MATLAB rather than using sequential loops

**Parallel performance is extremely dependent on a specific application and implementation**