

IFS in the DEISA Benchmark Suite

Peter Towers
ECMWF

Outline



- **Overview of the DEISA benchmark suite**
- **Porting of IFS into the suite**
- **The JuBE benchmark framework**
- **Preliminary results on the XT4 and Power6**
- **Future plans**

DEISA



- DEISA is a consortium of eleven leading European National Supercomputer Centres
- Strongly focused on High Performance distributed supercomputing
- Continental wide network and global file system
- DEISA Common Production Environment
- DEISA Extreme Computing Initiative
- Advancing Science in Europe

www.deisa.eu



RI-222919



The Benchmark Suite

- **Developed as part of the eDEISA project over a two year period**
- **Aim to provide codes and datasets suitable for benchmarking machines with hundreds of Teraflops of peak performance**
- **European wide team**
- **14 codes from a broad range of disciplines**
- **Representative of the scientific applications running on the DEISA HPC facilities**
- **Plus low level kernels**
- **Uses a consistent framework to build and run all codes**
 - JuBE from Juelich

The Initial Platforms

● Jump

- IBM eServer P690 at FZJ
- 1312 cores
- Power4+ 1.7GHz

● Louhi

- Cray XT4 at CSC
- 2024 cores
- AMD Opteron dual core 2.6GHz

● HLRB II

- SGI Altix 4700 at LRZ
- 9728 cores
- Intel Itanium Montecino dual core 1.6GHz

The Initial Codes

- **Astrophysics** GADGET, RAMSES
- **CFD and Combustion** Fenfloss
- **Earth Sciences and Climate Research** NEMO, ECHAM5
- **Life Sciences and Informatics** IQCS, NAMD
- **Materials Science** CPMD, QuantumESPRESSO, DL_POLY
- **Plasma Physics** GENE, PEPC
- **Quantum Chromodynamics** BQCD, SU3_AHIGGS

The Results



- **Runs were performed on up to 512 cores on each platform**
- **Results published in May 2008 on the DEISA website at**
 - **www.deisa.eu/science/benchmarking**

DEISA 2



- **A new two year benchmark project started June 2008 under DEISA 2**
- **The benchmark suite will be maintained and will evolve**
 - **New platforms will be benchmarked**
 - **Larger data sets will be used**
 - **DL-POLY no longer included**
 - **Access to the initial version no longer available**
 - **IFS will be included (using funds from eDEISA)**
 - **Forecast Model Only**

New Platforms

- **Platforms change**

- Jump and Louhi have been upgraded

- **vip**

- IBM eServer 575 at RZG
- 6560 cores on 205 32 way nodes
- Power6 dual core 4.7GHz

- **HECToR**

- Cray XT4 at EPCC
- 11238 cores on 5664 2 way nodes
- AMD Opteron dual core 2.8GHz

- **And soon?**

- IBM Blue Gene/P and Cray XT5

New Results



- **Deliverable sent to the EU Oct 31st**
 - Reports on the first 5 months work of the DEISA 2 benchmark team
- **Results from HECToR available on the DEISA website**
 - Runs performed on up to 2048 cores

IFS

- **Decision to include IFS taken in June 2008**
- **Work started late July**
- **Began with RAPS 10 and cycle 31r2**
- **Checked out the integration with JuBE**
- **Used JuBE to edit the necessary config files and Makefiles used by the existing Perl scripts**

- **Program call Bench written in Perl**
- **Uses XML files to control**
 - **Compilation**
 - **Job preparation**
 - **Job execution**
 - **Results verification**
 - **Analysis of run times**
- **Creates platform specific makefiles , job scripts and input parameter files on the fly**

Compilation with JuBE

- **Bench creates a unique directory for every run**
- **Option to create a new executable or reuse one**
- **Sources are extracted from a gzipped tarball**
- **Template files are used to create platform specific files**
 - For IFS we need 4 files
 - `config.in`, `project_config.in`, `Makefile.in` and `mkabs_fc.in`
- **XML syntax in a file `compile.xml` is used to modify placeholders in the templates and create**
 - `config`, `project_config`, `Makefile.arch`, `mkabs_fc`
- **The build scripts are then launched**

Execution with JuBE

- **Follows automatically after a clean build**
- **Data files and template namelist file are linked or copied using prepare.xml**
- **Template namelist file and job scripts are modified using execute.xml**
 - **Setting number of nodes, tasks, threads, nproma and other namelist values**
- **Batch job is submitted**
- **Multiple jobs can be submitted in one go**
 - **Each gets a unique identifier**

Recompilation with JuBE

- **All files are retained after a compilation or execution**
- **Can investigate failures, make corrections and compile manually**
- **The resulting executable can be copied and reused**
- **JuBE can then be used to submit further jobs**

IFS continued

- **The tests at 31r2 were positive**
 - Used both the Power5+ and our Linux Cluster
- **Switched to cycle 33r2 which was then in Esuite**
- **Ran control experiments at T159L91, T799L91, T1279L91**
 - Wave model was turned off
- **Generated reference output**
 - Results must not change with task or thread count and must be within 1% of the reference values
- **Extracted IFS sources and input data files**
- **Built and tested standalone from PrepIFS**
- **The next challenge was to reduce the volume of source code**

Reducing the IFS sources

- IFS at cycle 33r2 amounts to over 2.7 million source lines
- Compilation takes about 3 hours and creates 26 libraries
- Dr Hook was used to find which routines were called in the 3 experiments
- A perl script then dummied out the routines not used
 - These are the routines used only by 4DVAR
- Many libraries not needed at all apart from a few dummies or interface blocks
- Resulting sources amount to about 750 thousand lines
- Compilation time reduced to 40 minutes creating 8 libraries

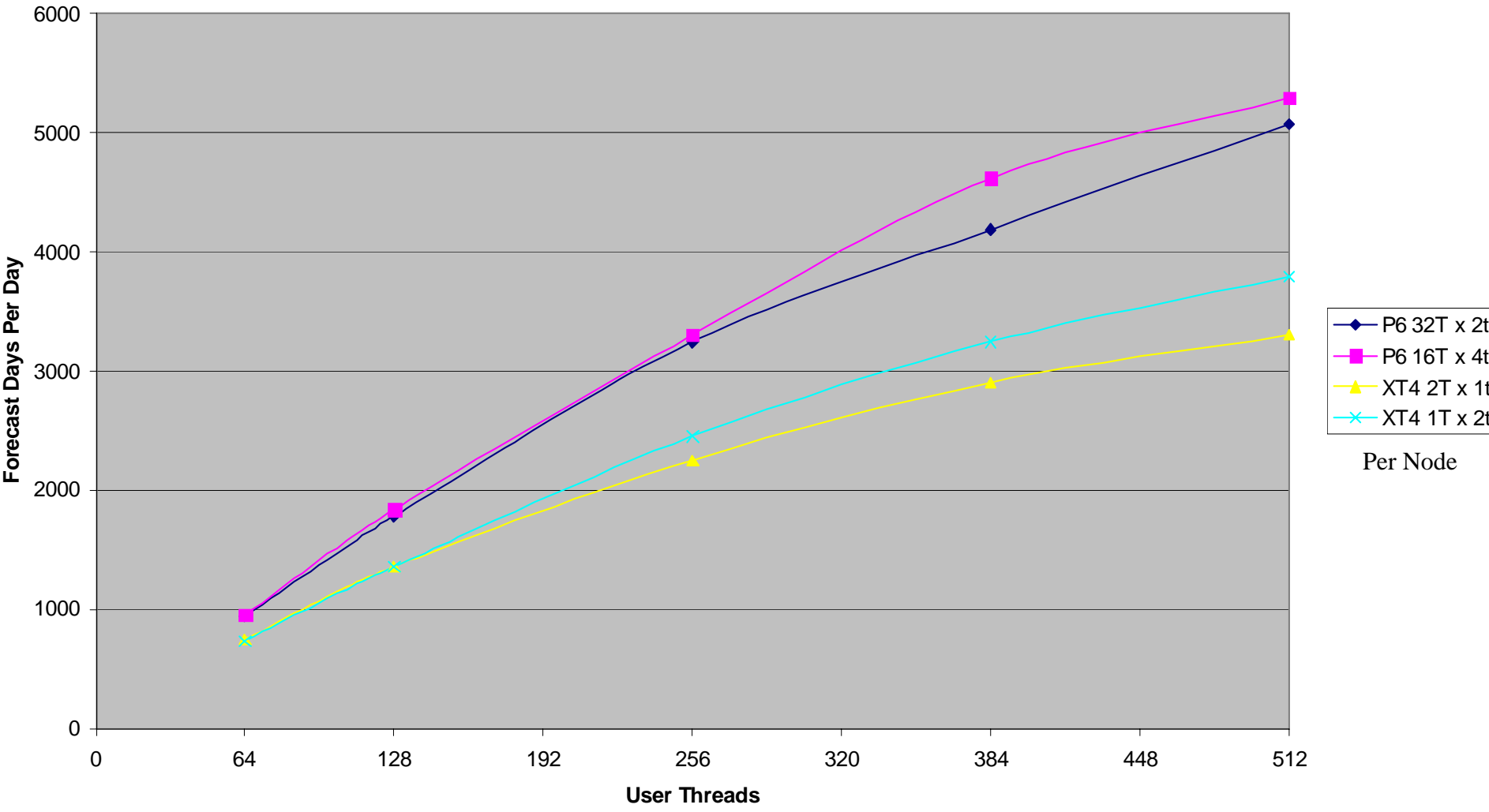
Preliminary Results

- **Started work on the XT4 at EPCC and the Power 6 at RZG in mid September**
 - Moved to Power 6 at ECMWF late October
- **Now have initial results from both platforms**
 - User Threads defined to be T MPI tasks \times t OpenMP threads
 - T159 up to 512 user threads
 - T799 up to 2048 user threads
 - T1279 up to 4096 user threads
- **Runs performed for 8 forecast hours**
 - Stay friends with the people giving us machine time
- **Wall time for each hour recorded**
 - Ignore the first and last
 - Average the rest
 - Compute forecast days per day

Preliminary Results



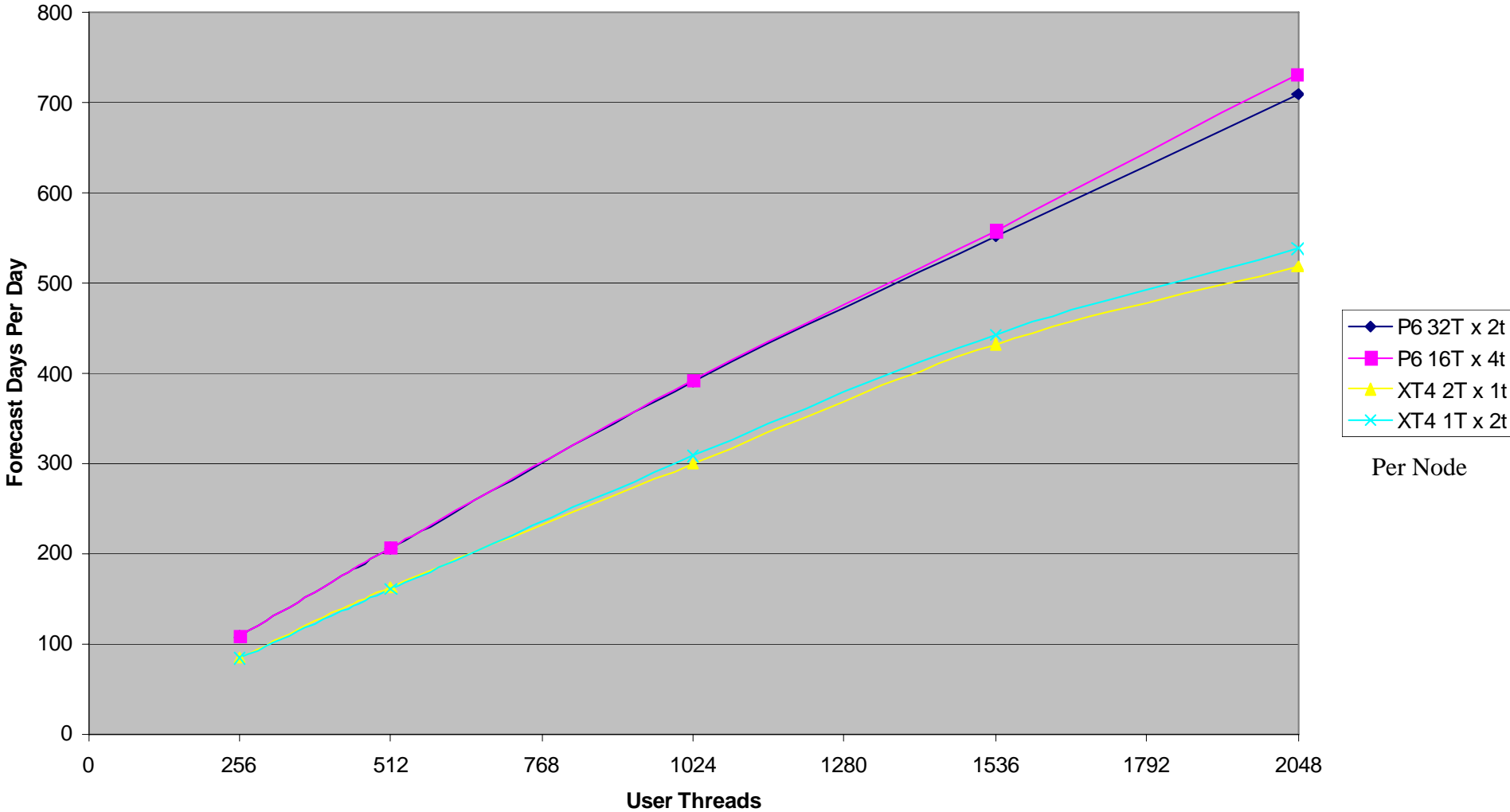
T159



Preliminary Results



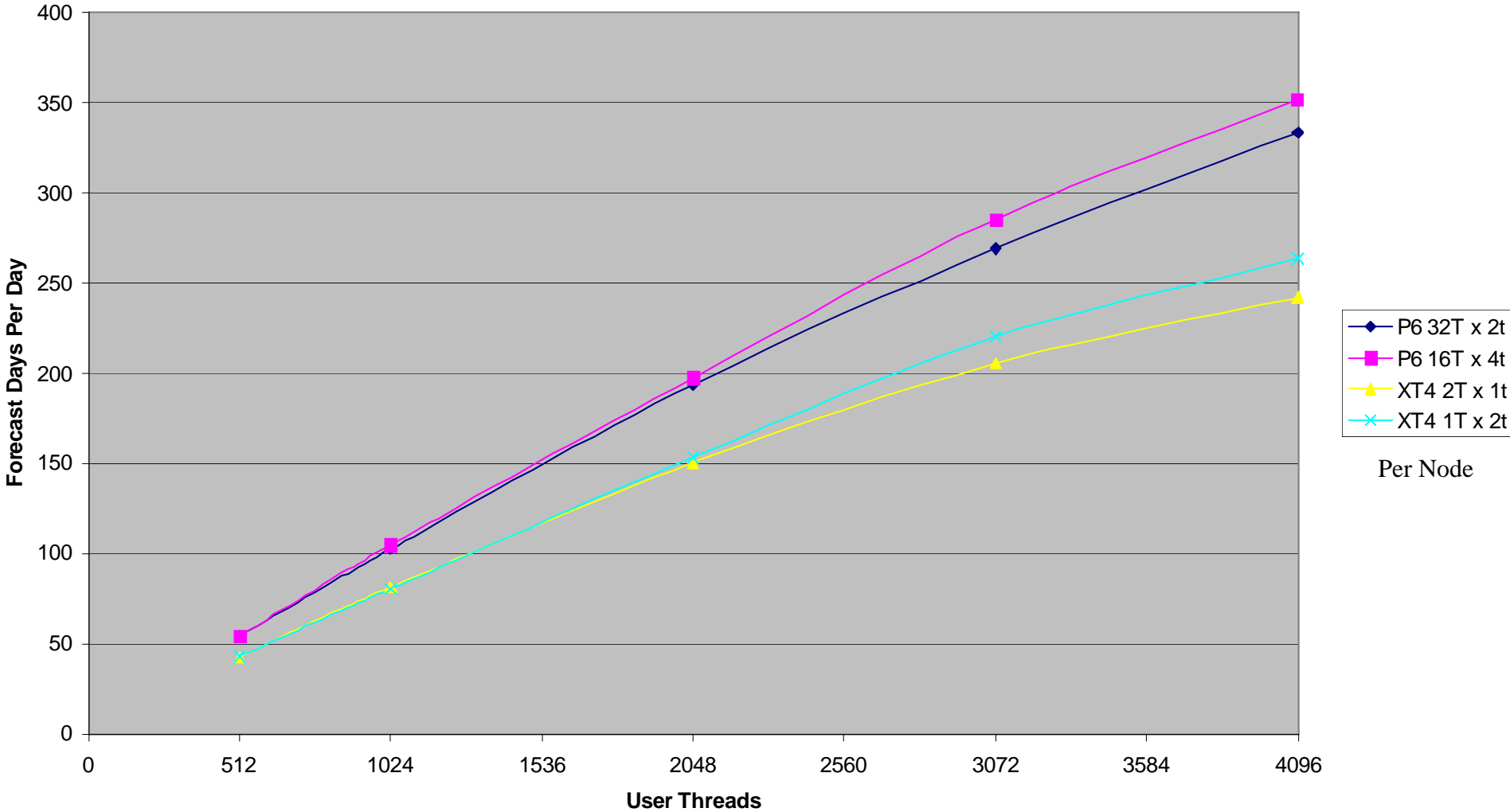
T799



Preliminary Results



T1279



Results – Some Conclusions

- **Mixed Mode is beneficial on both platforms**
 - 4 threads best on the Power 6
- **For 256 Tasks x 2 threads at T1279 we have**
 - Power 6 runs at 496 Gflops – 10.2% of peak
 - XT4 runs at 386 Gflops – 13.8% of peak
- **For 2048 Tasks x 2 threads at T1279 we have**
 - Power 6 runs at 3052 Gflops – 7.9% of peak
 - XT4 runs at 2378 Gflops – 10.4% of peak
- **Near identical parallel efficiency from 256 to 2048 tasks**
 - Both platforms are 77% efficient (6.1x out of 8)

Next Steps

- **Detailed comparison of results for 2048 tasks x 2 threads**
 - May identify tuning opportunities
 - Examine message passing layer closely
 - It is optimised for Power5 and Federation
- **Move to cycle 35R2?**
- **Move to a higher resolution**
 - Have T159, T1279 and T2047 as the data cases
 - Will increase the longevity of the benchmark
- **Move to new platforms**
 - Blue Gene/P , XT5, SGI Altix 4700?
- **Then ...**

Next Steps

- **Write documentation**
- **Make benchmark available under licence**
 - Benchmark framework will be available from the Deisa website
 - Source tarball available from ECMWF
- **Publish results**
- **All before the end of May**

And Finally Thankyou

- **To Deborah Salmond for putting my name forward for the project**
- **To Michele Weiland (EPCC) and the rest of the DEISA benchmarking team for their assistance**
- **To Jason Beech-Brandt from Cray for helping with the bells and whistles that tweak MPICH and the portals layer**
- **To George Mozdzynski for getting me back up to speed with IFS**
- **To Ricardo Correa for helping me understand DEISA**
- **And**

Thankyou for Listening



Questions?