

Scaling ECMWF's IFS forecast model to large numbers of processors

George.Mozdzynski@ecmwf.int

Outline

- **Background**
 - ECMWF
 - IFS
 - RAPS/benchmarking
- **Scaling Issues**
 - Computer Architecture
 - Application

IFS : T799 L91 10 day forecast on p575+

Run on hpcf (ECMWF second cluster) at IBM
Poughkeepsie on 2112 CPUs =132 nodes

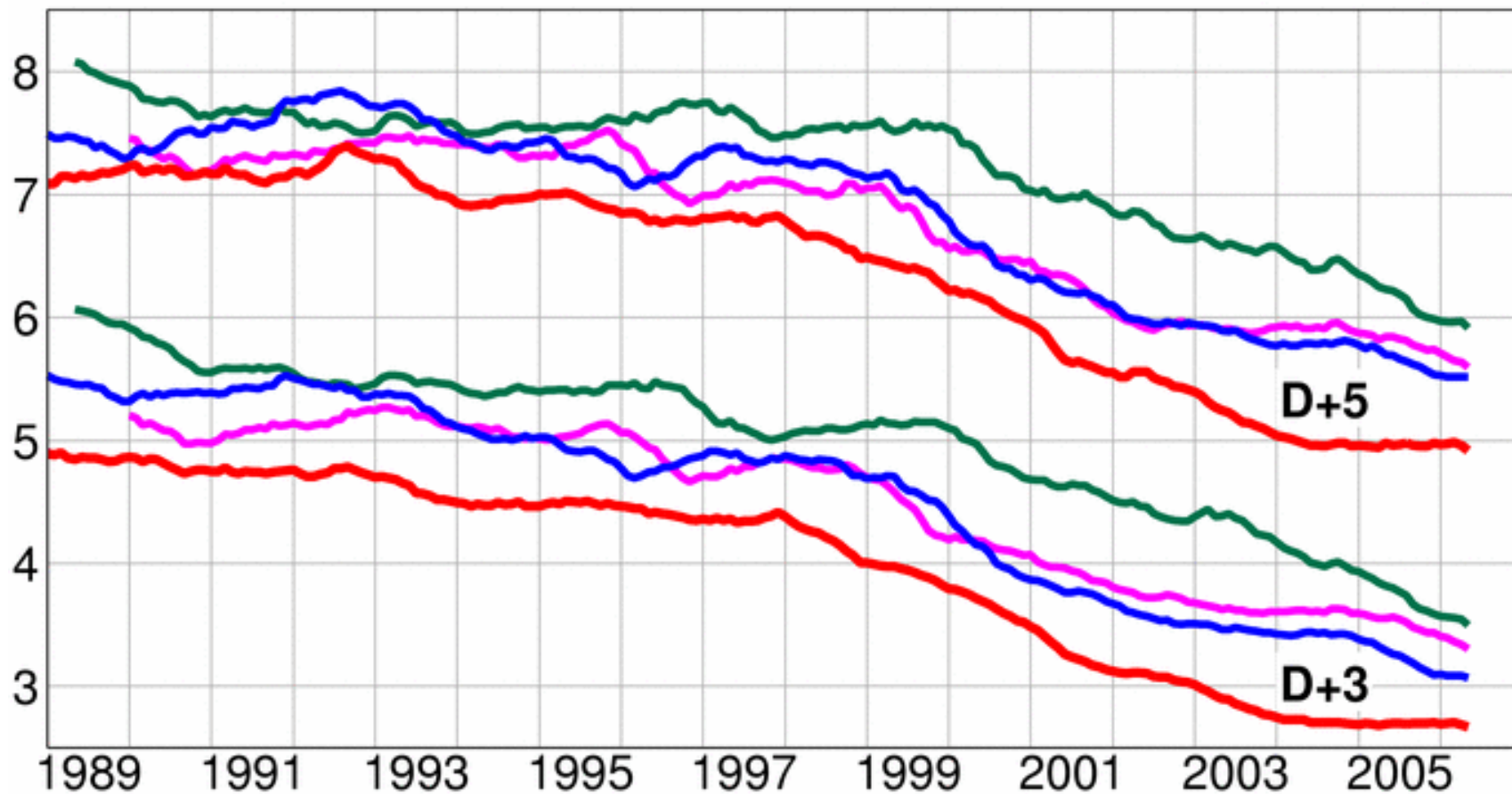
528 MPI tasks each with 8 OpenMP threads
(SMT) - 'parallelised over 4224 user threads'

Run time for 10-day forecast was 804 seconds
running at **2.08 Tflops** (13% of peak)

ECMWF is the leader in the accuracy of forecasts

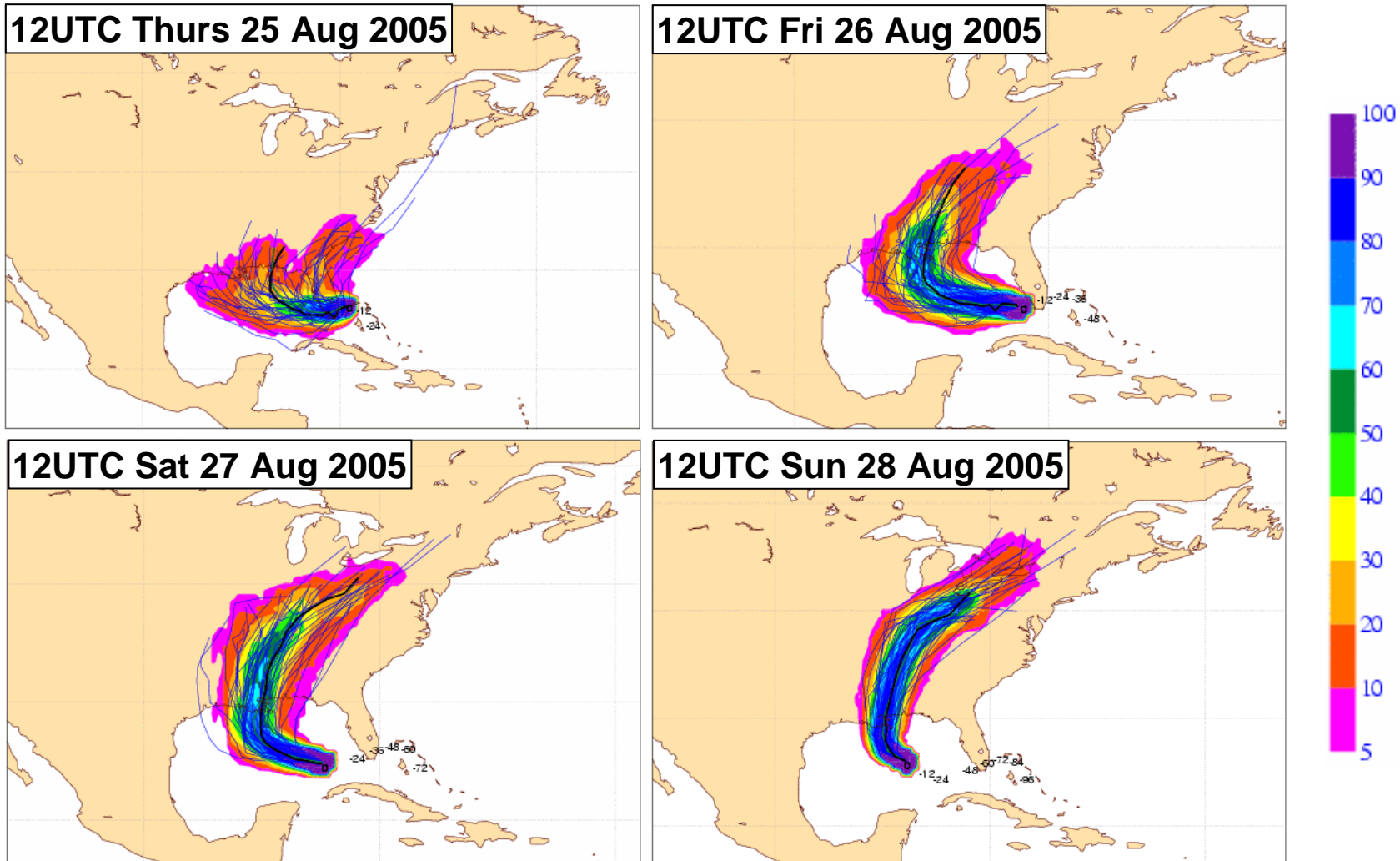
R.m.s. error (hPa) of surface-pressure forecasts for three and five days ahead

— ECMWF — UK — USA — JAPAN



Ensemble forecasts of track of Hurricane Katrina

Probability that KATRINA will pass within 120km radius during the next 120 hours
tracks: black=OPER, green=CTRL, blue=EPS numbers: observed positions at t+.h



ECMWF:

- A European organisation, located in Reading, UK
- Principal objectives:
 - development of methods for forecasting weather beyond two days ahead
 - collection and storage of appropriate meteorological data
 - daily production & distribution of forecasts to the Member States
 - provision of archival/retrieval facilities to the Member States
 - provision of computer resources to the Member States
- Emerging role in global environmental monitoring
- Annual budget of about 40 million Euro
- Staff of about 220

Member States:

Belgium	Denmark	Germany
Spain	France	Greece
Ireland	Italy	Luxembourg
The Netherlands	Norway	Austria
Portugal	Switzerland	Finland
Sweden	Turkey	United Kingdom

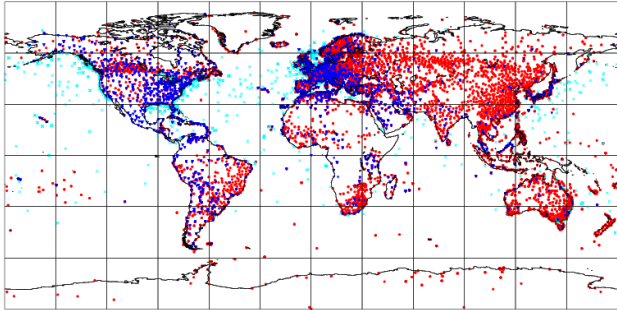
Co-operation agreements with:

Croatia	Czech Republic	Estonia	
Iceland	Hungary	Romania	
Slovenia	Serbia & Montenegro		
ACMAD	CLRTAP	CTBTO	ESA
EUMETSAT	JRC	WMO	

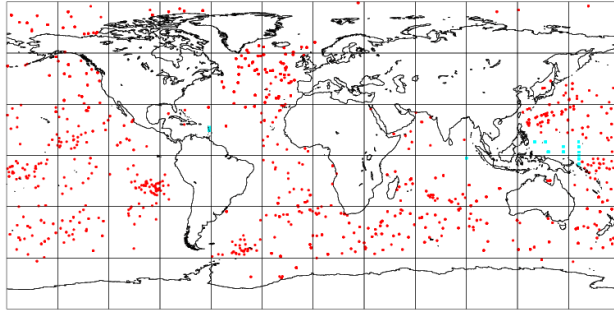
Operational system: 2006

- Four-dimensional variational data assimilation based on 25km/80km (T799/T255) horizontal resolution and 91-level vertical resolution (4D-Var)
- 25km 91-level model for single deterministic forecast
- 50km/80km/125km (T399/T255/T159) 62-level atmospheric model, coupled with ocean model, for 50-member ensemble forecasts to 10/15/32 days ahead (EPS)
- 125km 62-level atmospheric model, coupled with oceanic model, for seasonal prediction

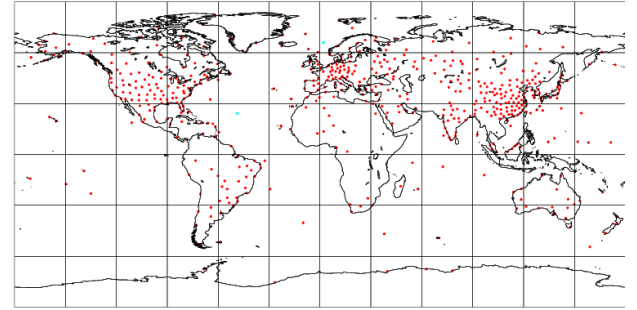
Synops and ships



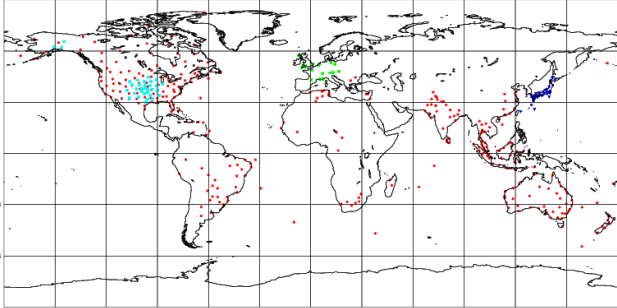
Buoys



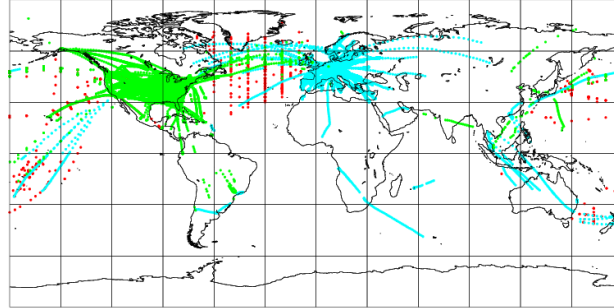
Radiosondes



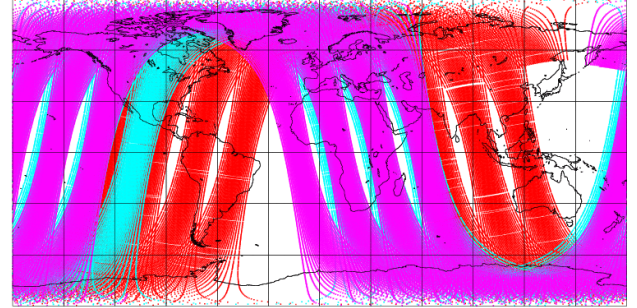
Pilots and profilers



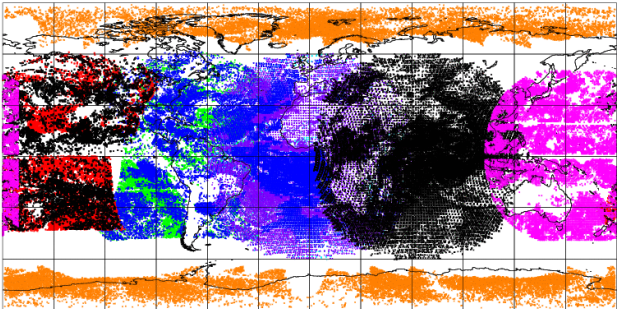
Aircraft



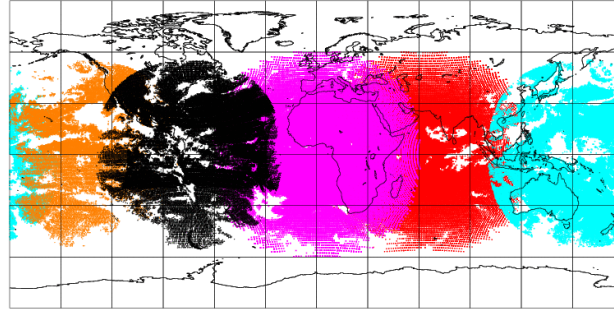
IR and MW sounders



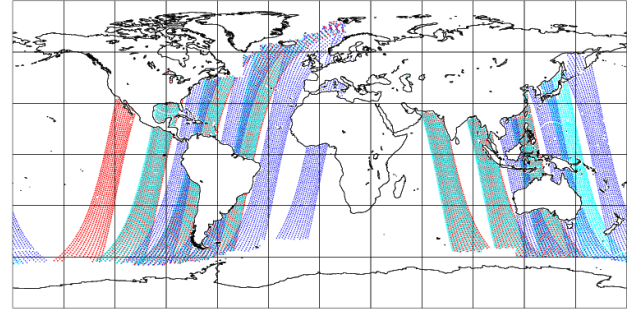
Satellite winds



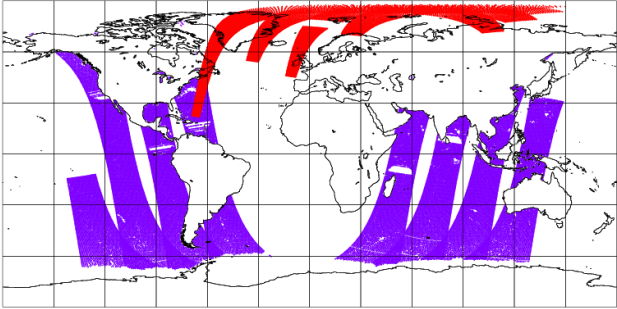
Water-vapour radiances



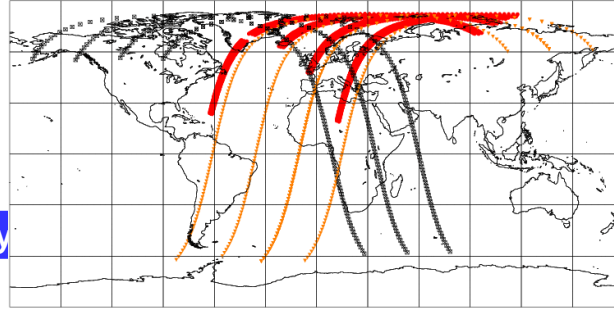
SSM/I



Scatterometer



Ozone



**observations: a load
balance problem**

Current HPC system - IBM cluster 1600

- **Service contract with several phases**
 - Phase 1 (2002 – 2004) p690 Server / Power4 @ 1.3 GHz
 - *Phase 2 (skipped due to earlier delivery of Phase 3)*
 - Phase 3 (2004 – 2006) p690 Server / Power4 @ 1.9 GHz
 - Phase 4 (2006 – Q1/2009) p5-575 Server / Power5 @ 1.9 GHz
- **HPC system - Phase 4 (2006 – 2009)**
 - Two identical IBM AIX Clusters, each with ~150 nodes
 - Each 32 GB node has 16 Power5 @ 1.9 GHz SMT processors
 - Ten nodes per cluster are dedicated to I/O and networking
 - 100 TB of FC disk storage in total
 - **Total sustained performance on ECMWF's codes is around 4 Tflops (based on a typical job mixture)**

2004-2006



2H2006

hpcc & hpcd

IBM p690+

Peak performance
7.6 Gflops per CPU
(Power4+ 1.9 GHz)

1 Gbyte memory/CPU

32 CPUs/node

hpce & hpcf

IBM p575+

Peak performance
7.6 Gflops per CPU
(Power5+ 1.9GHz)

2 Gbytes memory/CPU

16 CPUs per node
with SMT

Same Federation Switch

SMT = Simultaneous Multi-Threading

- **Power5+ Node has physical 16 CPUs = 8 dual-core chips**
- **2 'logical CPUs' are allocated to each 'physical CPU'**
- **These CPUs can be used with MPI tasks or OpenMP**
- **Programs benefit from SMT if mix of memory / FP ops**
e.g. IFS
- **Some programs don't benefit from SMT**
e.g. If they have a lot of memory traffic per FP op
or if they are 'FP bound' like SGEMM
or the program doesn't scale well - uses 2*CPUs
- **User can choose not to use SMT inside loadleveler**
@ resources = ConsumableCpus(2*Threads)

RAPS (ReaApplications on ParalleL Systems)

- Initiative founded in the early 90's, consisting of the Consortium ("Application Developers") and the Working Group of Hardware Vendors
- Goal was, to reduce the effort for benchmarking for the Consortium members and also for the hardware vendors
- The Consortium members agree on a common portable programming model (F90 + MPI in the beginning)
- Availability of benchmark codes ahead of a formal procurement
- Meetings now once a year, next at ECMWF 1st Nov,
 - Use of HPC in Meteorology Mon 30 Oct - Fri 3 Nov
 - RAPS workshop day is Wednesday 1st Nov
 - more info via hpcworkshop@ecmwf.int

IFS - T799L91 10-day forecast from RAPS9

	CPUs MPI x OMP	WALL (secs)	% of peak
Power4+ 1.9GHz p690+ hpcd	768 192 x 4	3848	7.6%
Power5+ 1.9GHz p575+ hpce	768 SMT 192 x 8	2457	11.8%

Speed-up: Power4+ → Power5+ = 1.56

IFS - T799L91 10-day forecast from RAPS9

	CPUs MPI x OMP	WALL (secs)	%Comms	Gflops	% of peak
Power4+ 1.9GHz p690+ hpcd	768 192 x 4	3848	12.6%	444	7.6%
Power5+ 1.9GHz p575+ hpce	768 SMT 192 x 8	2457	14.0%	696	11.8%

Total floating-point ops = $1,710,000 \times 10^9$

Integrated Forecasting System (IFS)

- IFS 1992 - today
 - Collaboration between Meteo France and ECMWF
 - Source ~ 1.8 million lines
 - Fortran 95, some C
 - Good performance on scalar and vector systems
- IFS model characteristics:
 - Spectral
 - Semi-implicit
 - Semi-Lagrangian

IFS - Parallelised using 'mixed' MPI and OpenMP

MPI communications

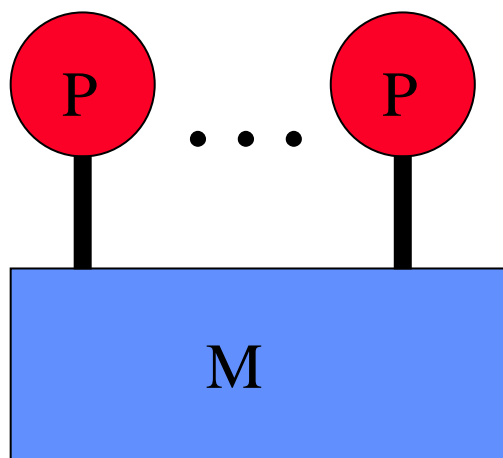
- Transpositions
 - Between Grid point, Fourier and Spectral spaces
- Wide halo exchange
 - Semi Lagrangian method
 - Radiation grid interpolation
- Long messages
- Typically MPI_ISEND/RECV/WAITALL or collective

OpenMP

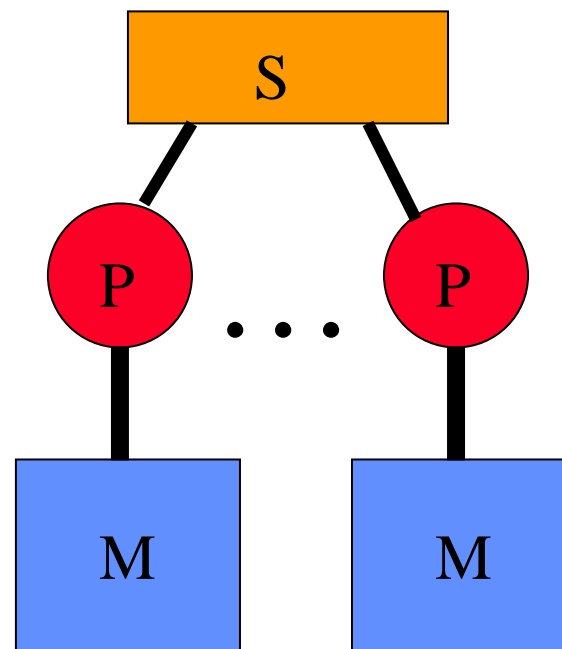
- Shared memory nodes
- Memory efficient
- Use 4/8 threads

Types of Parallel Computer

P=Processor
M=Memory
S=Switch



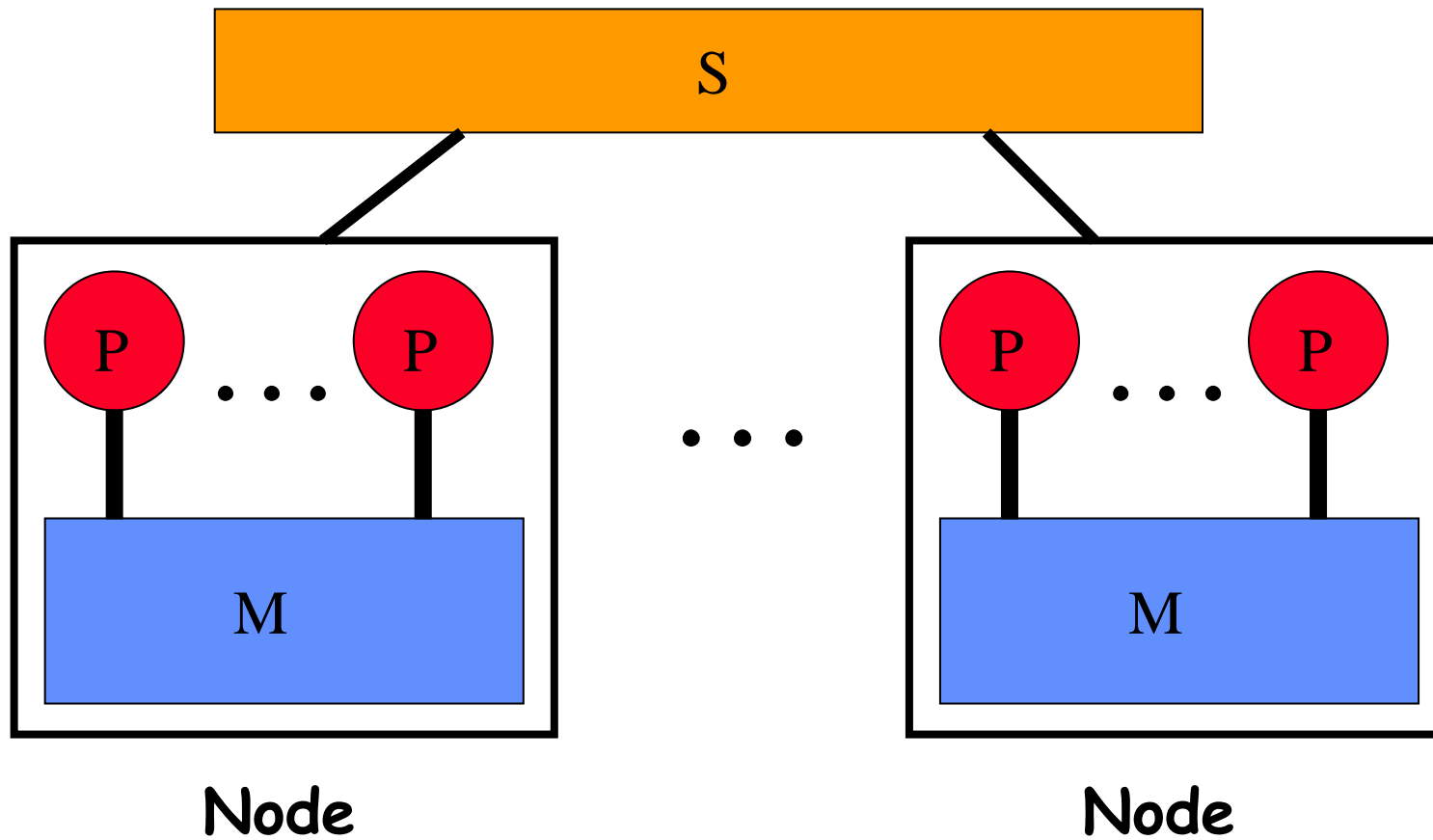
Shared Memory



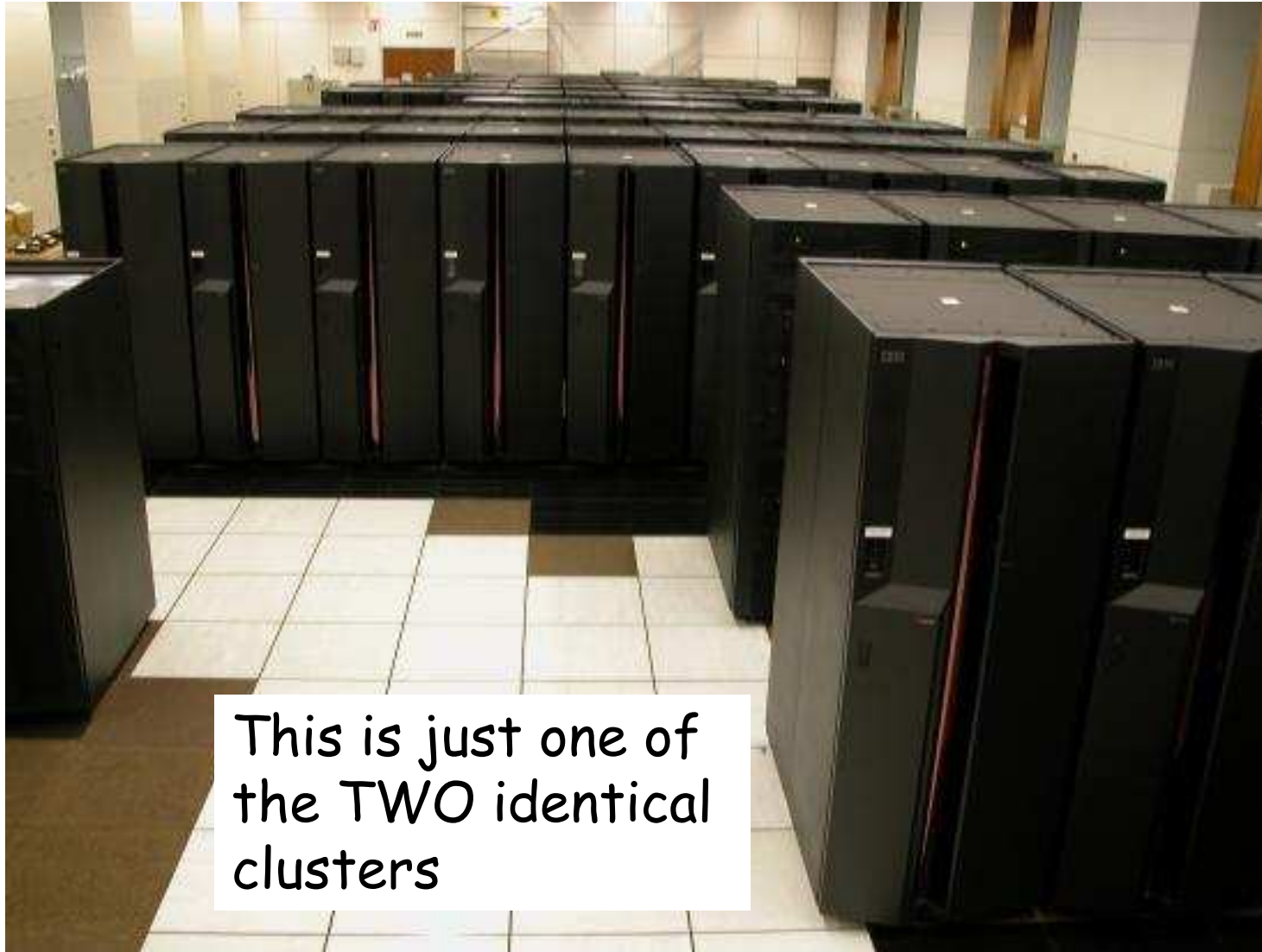
Distributed Memory

IBM Cluster 1600 (at ECMWF)

P=Processor
M=Memory
S=Switch



IBM Cluster 1600's at ECMWF

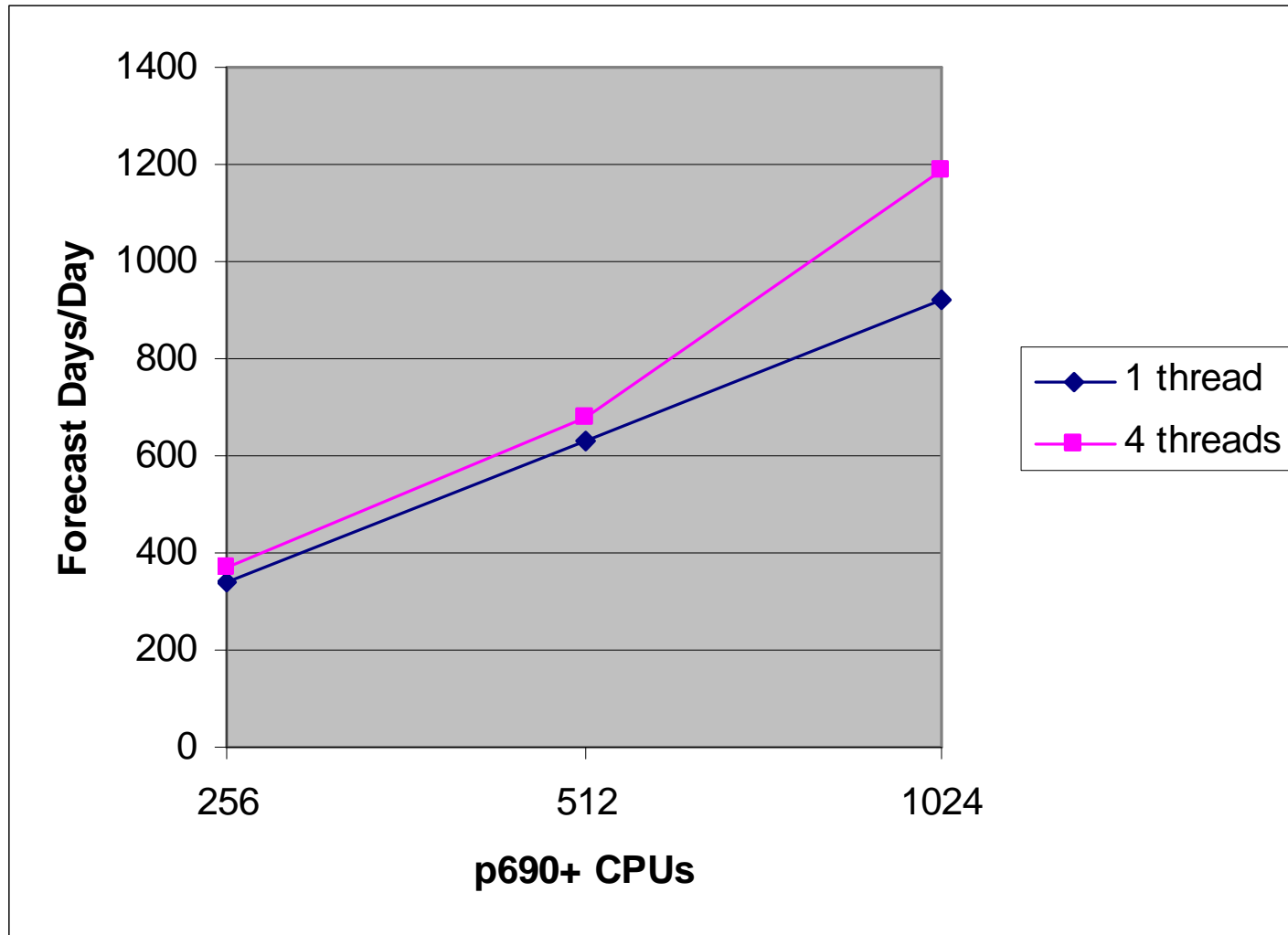


T799 L60 10 day forecast with MPITRACE library (excerpt for task 1 of 512 tasks)

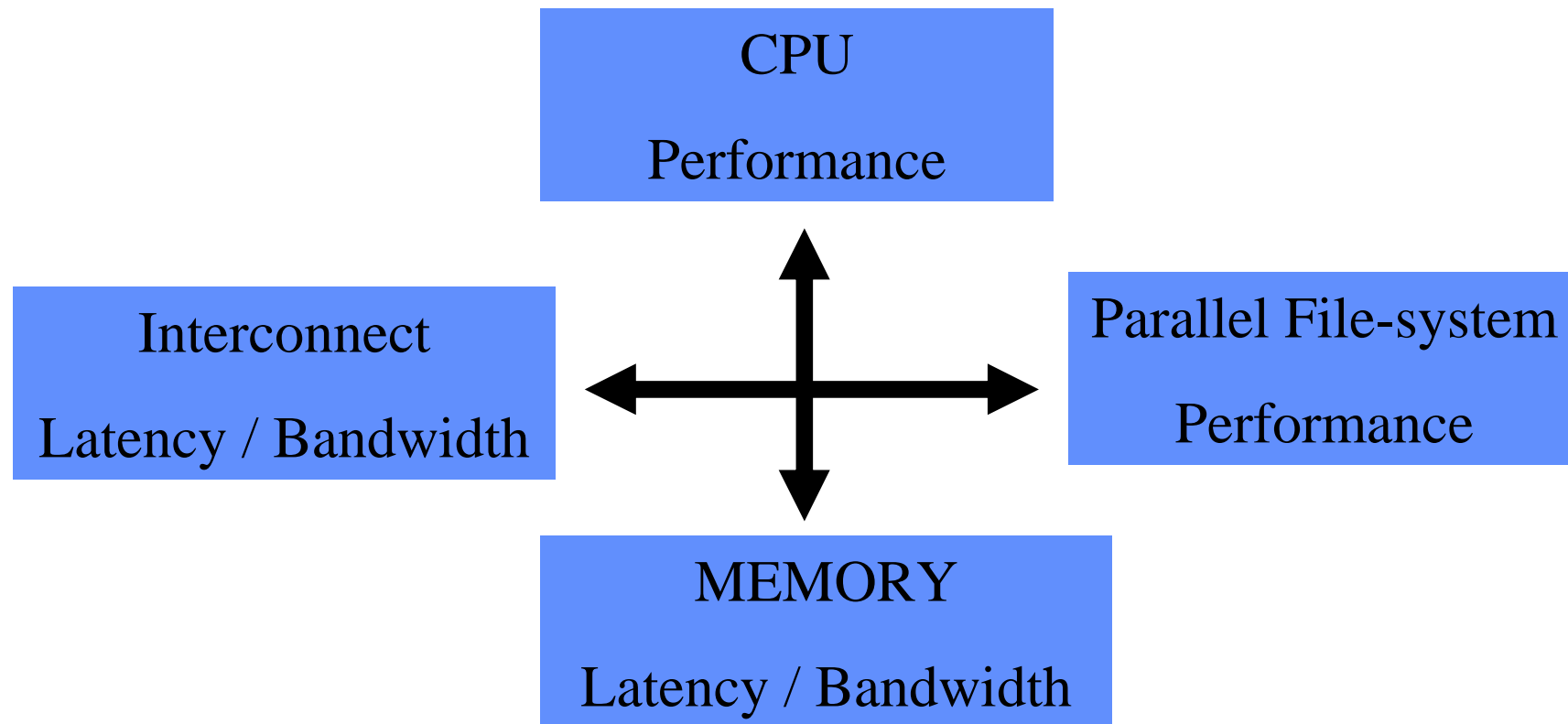
```
-----  
MPI Routine                #calls      avg. bytes    time(sec)  
-----  
MPI_Comm_size              2469         0.0           0.020  
MPI_Comm_rank              2            0.0           0.000  
MPI_Send                   16863        13179.5       0.614  
MPI_Bsend                  735878       1138.5        5.095  
MPI_Isend                  336706       107239.5     6.145  
MPI_Recv                   332213       264324.3     302.072  
MPI_Buffer_attach         2            0.0           0.000  
MPI_Buffer_detach         2            0.0           0.000  
MPI_Waitall                44657        0.0           34.346  
MPI_Bcast                  8857         22847.4       3.360  
MPI_Barrier                5375         0.0           87.483  
MPI_Gatherv                1780         5113.2        13.567  
MPI_Allgatherv             2400         28120.0       152.894  
MPI_Allreduce              42           1020.0        0.036  
MPI_Alltoallv              2468         114502.6     134.614  
-----  
total communication time = 740.246 seconds. ***  
total elapsed time      = 4330.087 seconds.
```

***Above run is with no OpenMP (for 128 tasks x 4 threads run total comms is 535 secs)

T511 L60 forecast model (MPI v OpenMP)



Key Architectural Features of a Supercomputer



“a balancing act to achieve sustained Teraflop performance”

Memory Latency / Bandwidth

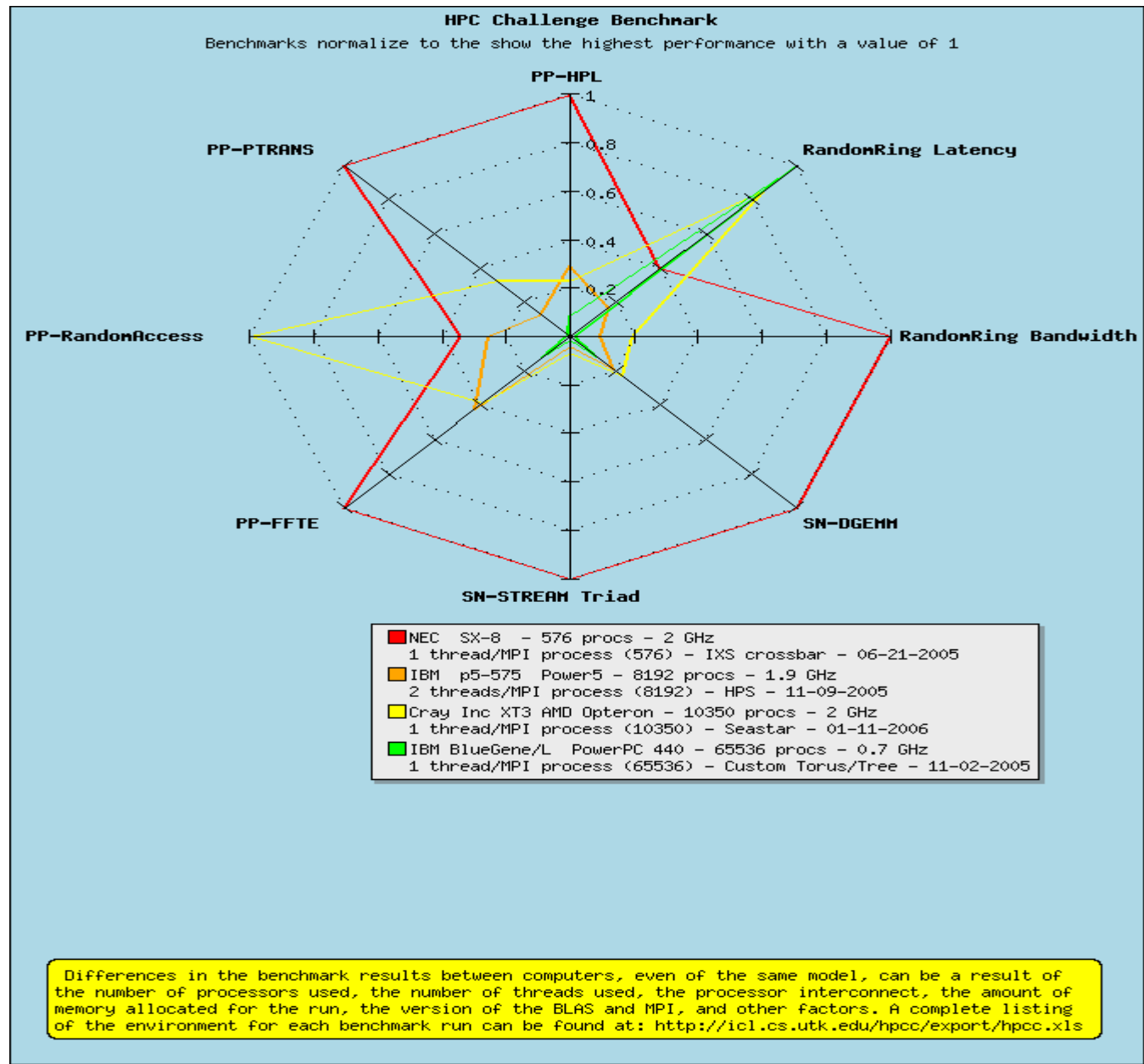
System	CPUs	G-HPL Gflop/s	EP-Stream GW/s	Balance (low is good)
NEC SX-8 2 GHz	576	8,000	2,944	2.72
CRAY XT3 Opteron 2.6 GHz	1100	4,780	660	7.25
IBM p5-575 Power5 1.9 GHz	10,240	57,870	6,899	8.39
IBM P655 Power4+ 1.5 GHz	1,024	3,110	217	14.33

Figures from HPC Challenge Benchmark, Balance is G-HPL Gflops/s divided by EP-Stream bandwidth

HPC Challenge Benchmark Kiviatic Chart

PP: Per Proc

SN: Single node



Grid point space blocking for Cache

```
DO J=1,ngptot,nproma  
    ibeg=j  
    iend=min(nproma, ngptot-J+1)  
    call gp_calcs(ibeg,iend,...)  
ENDDO
```

Many such code blocks in IFS

Adding OpenMP

```
!$OMP PARALLEL DO PRIVATE(J,IBEG,IEND)
```

```
DO J=1,ngptot,nproma
```

```
    ibeg=j
```

```
    iend=min(nproma, ngptot-J+1)
```

```
    call gp_calcs(ibeg,iend,...)
```

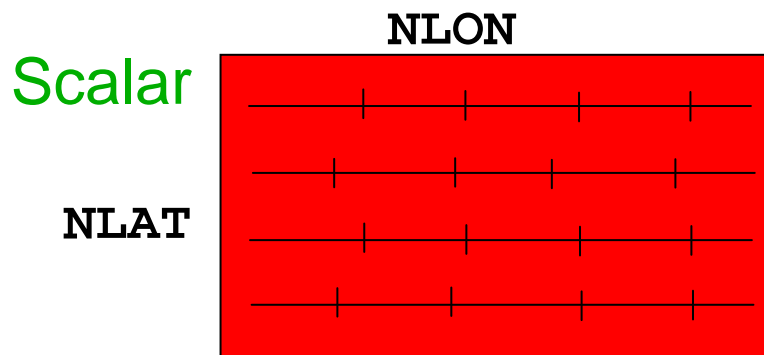
```
ENDDO
```

```
!$OMP END PARALLEL DO
```

Also some load balancing (OpenMP only)

```
!$OMP PARALLEL DO PRIVATE(J,IBEG,IEND) &  
!$OMP& SCHEDULE(DYNAMIC,1)  
DO J=1,ngptot,nproma  
    ibeg=j  
    iend=min(nproma, ngptot-J+1)  
    call gp_calcs(ibeg,iend,...)  
ENDDO  
!$OMP END PARALLEL DO
```

Grid-Point Calculations



```
DO J=1, NGPTOT, NPROMA  
  CALL GP_CALC  
ENDDO
```

Lots of work
Independent for each J

High Level Blocking Scheme :

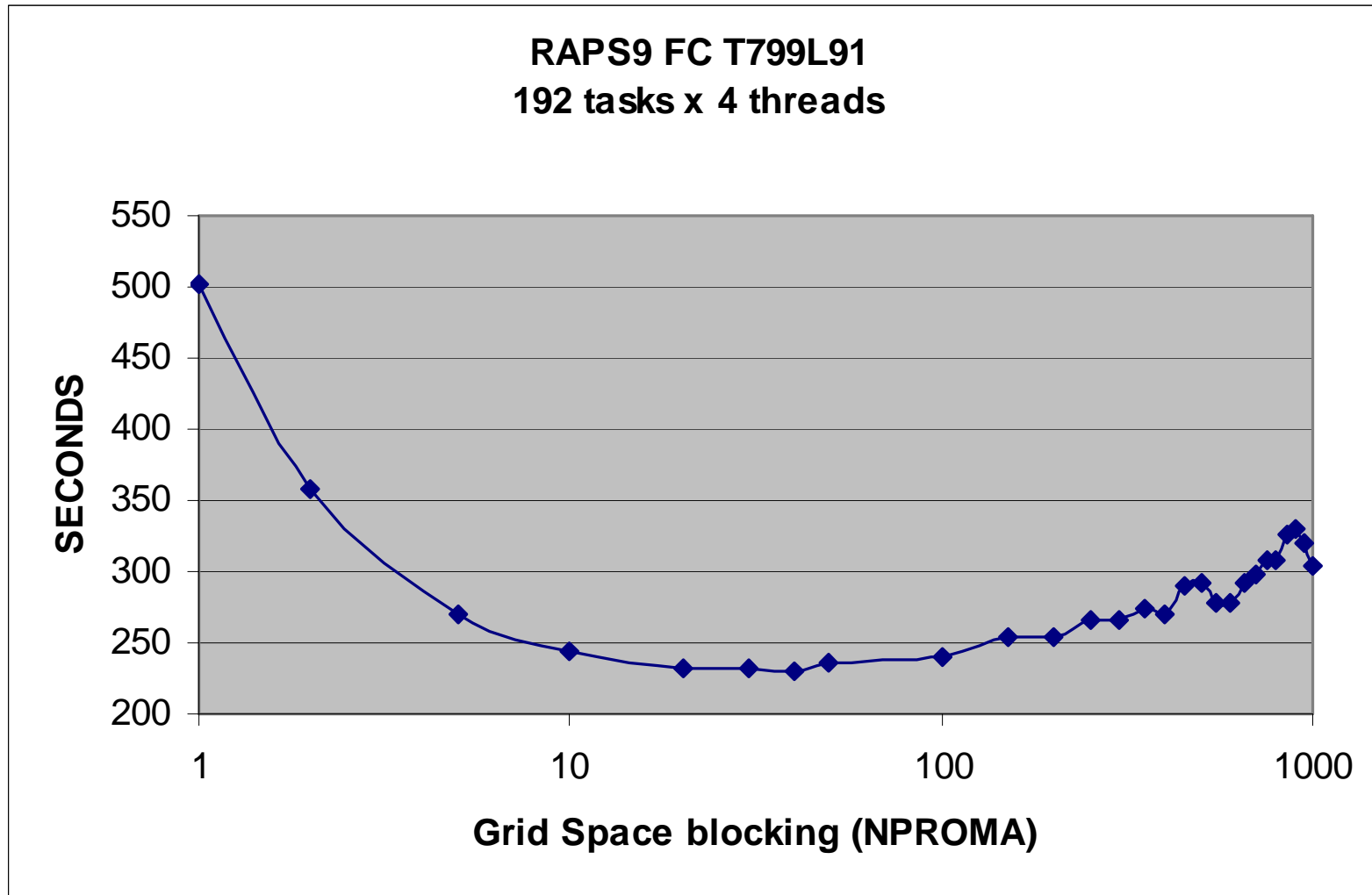
```
U(NGPTOT,NLEV)
```

```
NGPTOT = NLAT * NLON
```

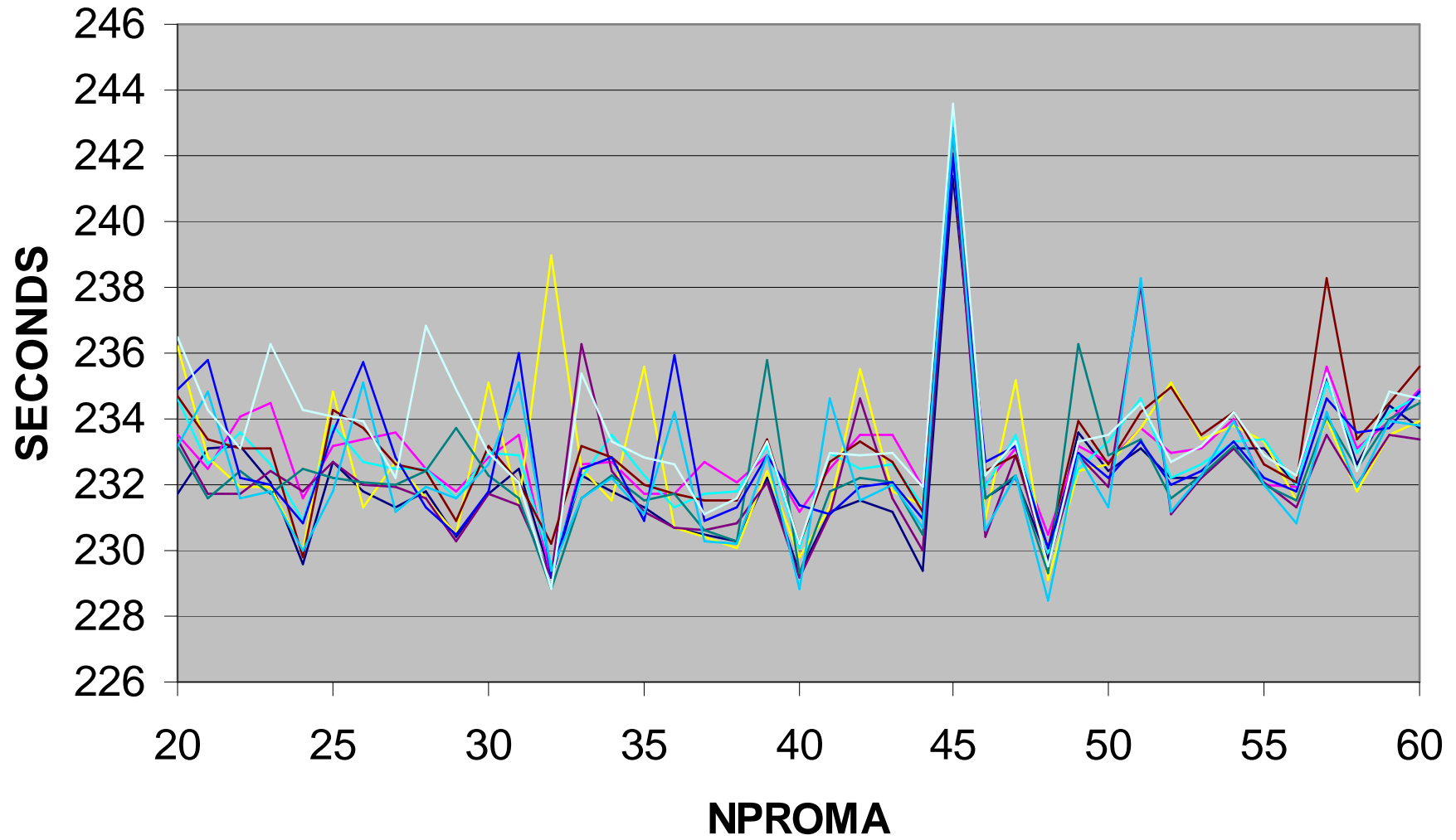
```
NLEV = vertical levels
```

```
SUB GP_CALC  
  
DO I=1,NPROMA  
ENDDO  
  
END
```

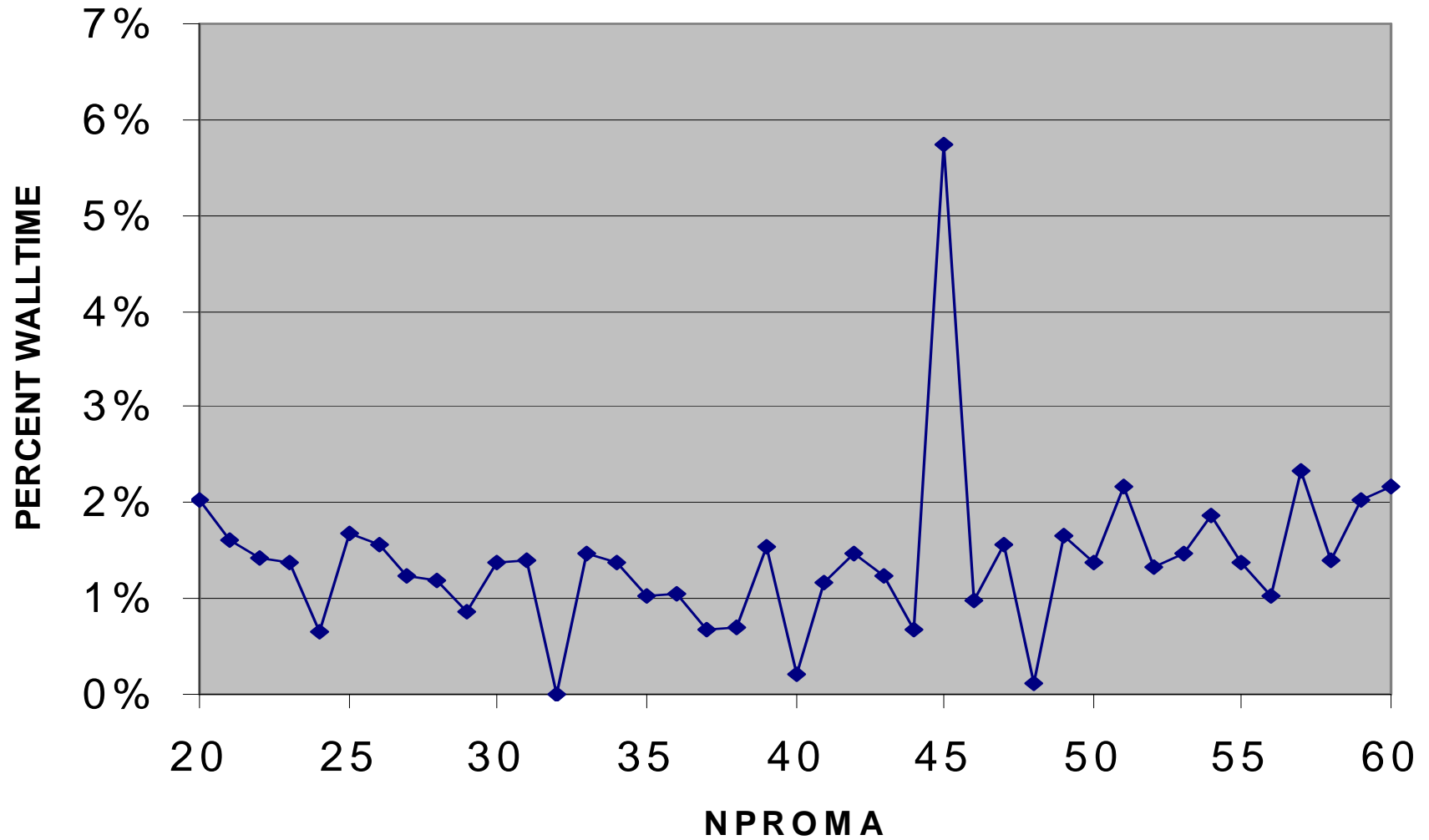
Grid point space blocking for Cache (p690+)



T799 FC 192x4 (10 runs)

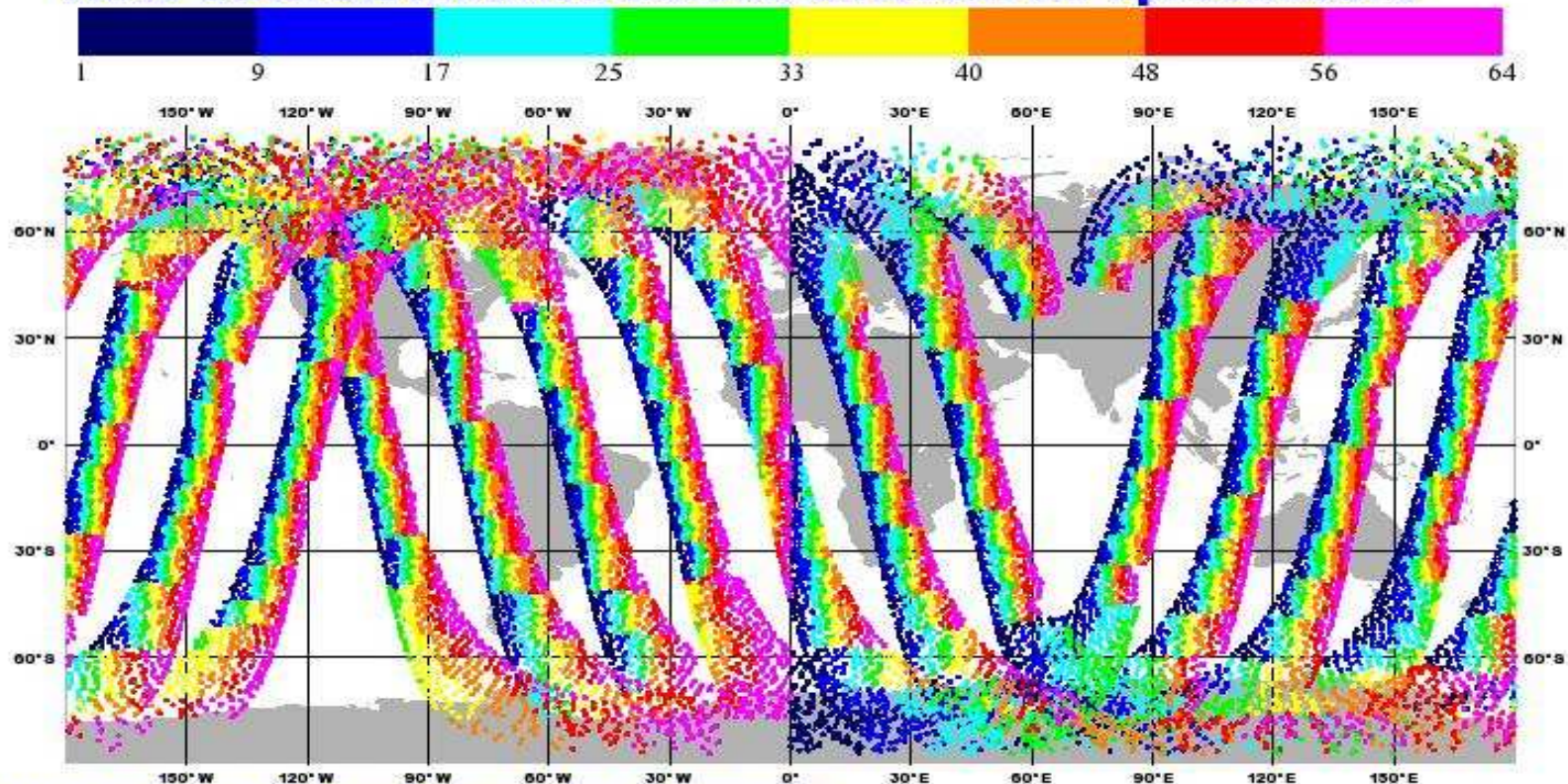


T799 FC 192x4 (average)



Observation load-imbalance

ODB database: CCMA Data query: target
No. of data points 20088
AIRS data distribution across different MPI-processors



MAGICS 6.9 gy/rl - mps Thu Sep 16 13:26:11 2004 /hda1/data/b/igbnp/mps/DC DA.2004091012/CC MA



Instrumentation - Dr. Hook (Sami Saarinen) profile for T511 forecast model

#	%Time (self)	Cumul (sec)	Self (sec)	Total (sec)	#calls	MIPS	MFlops	%Div	Routine
1	7.43	35.027	35.027	40.573	49	961	273	2.9	WVCOUPLE@1 [567,1]
2	3.67	52.349	17.322	17.367	5824	1113	546	3.6	*CLOUDSC@1 [5,4]
3	3.65	52.349	17.204	17.287	5791	1116	548	3.6	CLOUDSC@4 [5,4]
4	3.64	52.349	17.181	17.289	5769	1118	549	3.6	CLOUDSC@2 [5,4]
5	3.63	52.349	17.138	17.202	5770	1117	549	3.6	CLOUDSC@3 [5,4]
6	3.51	68.918	16.569	16.584	54	783	0	27.6	TRMTOL_COMMS@1 [525,1]
7	2.76	81.935	13.017	18.260	51	926	1	2.8	TRGTOL@1 [520,1]
8	2.51	93.763	11.829	11.831	54	742	0	24.8	TRLTOG_COMMS@1 [523,1]
9	2.41	105.145	11.382	30.536	11540	1106	88	3.4	*CUASCN@3 [30,4]
10	2.40	105.145	11.336	30.436	11538	1112	88	3.4	CUASCN@2 [30,4]
11	2.39	105.145	11.274	30.394	11582	1110	88	3.4	CUASCN@4 [30,4]
12	2.39	105.145	11.267	30.072	11648	1113	86	3.4	CUASCN@1 [30,4]
13	2.36	116.296	11.150	11.185	3492	2135	2172	0.0	*MXMAOP@1 [166,4]
14	2.31	116.296	10.897	10.940	3502	2218	2259	0.0	MXMAOP@2 [166,4]
15	2.30	116.296	10.832	10.920	3474	2216	2258	0.0	MXMAOP@4 [166,4]
16	2.29	116.296	10.816	10.910	3484	2224	2266	0.0	MXMAOP@3 [166,4]
17	1.94	125.448	9.152	9.327	27785	1433	682	0.0	*LAITQM@3 [138,4]
18	1.94	125.448	9.130	9.263	27980	1434	679	0.0	LAITQM@1 [138,4]
19	1.92	125.448	9.073	9.256	27715	1432	682	0.0	LAITQM@4 [138,4]
20	1.92	125.448	9.045	9.220	27750	1440	686	0.0	LAITQM@2 [138,4]
21	1.85	134.173	8.725	8.785	5563	985	592	2.2	*SLTEND@4 [297,4]
22	1.85	134.173	8.724	8.777	5596	987	593	2.2	SLTEND@1 [297,4]
23	1.83	134.173	8.654	8.741	5541	986	593	2.2	SLTEND@2 [297,4]
24	1.83	134.173	8.621	8.658	5546	989	595	2.2	SLTEND@3 [297,4]
25	1.82	142.737	8.565	8.580	51	782	0	21.6	TRLTOM_COMMS@1 [524,1]
26	1.80	151.219	8.482	69.102	13	581	22	10.6	RADINTG@1 [207,1]

Dr. Hook memory profile for T799 forecast

Memory-profiling information for program='./MASTER', proc#1:

No. of instrumented routines called : 576

Memory usage : 1505 MBytes (max.seen), 294 MBytes (leaked),
 2062 MBytes (heap), 1834 MBytes (max.rss),
 128 MBytes (max.stack), 1145 (paging)

#	Memory-% (self)	Self-alloc (bytes)	+ Children (bytes)	Self-Leak (bytes)	Heap (bytes)	Max.Stack (bytes)	Paging (delta)	#Calls	#Allocs	#Frees	Routine
1	20.00	587569264	554450336	76982472	2162753248	40784	51	120	723	720	GP_MODEL@1
2	9.95	292161088	78809176	0	2162753248	4404832	0	24	268	268	RADINTG@1
3	8.02	235681728	79457488	246216	2162753248	134632240	50	120	1920	1917	CALL_SL@1
4	6.52	191547280	0	0	2162753248	10530944	4	123	244	244	>TRS-FTINV
5	6.25	183480072	370970264	0	2162753248	3900704	1	24	528	528	RADDRV@1
6	4.92	144638640	0	0	698940640	20593680	0	2	10	10	>TRS-SULEG
7	4.72	138637064	134851248	96178120	968162656	18432	0	1	34	31	SUSC2B@1
8	4.59	134851248	0	134851248	871955744	19392	0	1	2	0	GMV_SUBS
9	4.39	128847896	128505720	280	698875072	13056	0	1	50	49	SUTRANS@1
10	4.19	123052080	0	0	2162753248	9690880	0	123	242	242	>TRS-LTINV
11	2.79	82091920	0	0	2162753248	9692368	1	121	242	242	>TRS-FTDIR
12	2.71	79457488	0	0	2162753248	135193904	0	120	360	360	SLCOMM2A@1
13	2.68	78809176	0	0	1998651072	4408480	0	2	6	6	SLCOMM@1
14	2.39	70328856	0	0	2162753248	10530944	4	123	121	121	>TRS-FTINV
15	2.18	64001688	400	1664	423754272	13024	0	1	8	3	SUMPINI@1
16	1.63	47978864	13602032	0	1016200576	17115456	1	1	1	1	SUGRIDUG@1
17	1.56	45774496	144638640	280	698940640	27184	835	1	113	112	SUECRAD@1
18	1.51	44311240	0	0	2162753248	134635936	0	120	360	360	SLCOMM1@1
19	1.31	38481456	0	0	2162753248	21983104	0	120	240	240	TRSTOM@1
20	1.27	37441424	0	0	2162753248	21982880	0	120	240	240	TRMTOS@1
21	1.05	30763032	0	0	2162753248	9692368	0	121	121	121	>TRS-FTDIR
22	1.05	30763032	82091920	0	2162753248	672608	0	120	240	240	TRANSDIR_MDL@1
23	0.79	23068936	0	40	698875072	13264	0	1	4	3	SUALSPA@1
24	0.70	20555752	0	0	2162753248	4701600	0	46	138	138	SLCOMM2@1

Dr. Hook memory profile for T799 forecast

#	Memory-% (self)	Self-alloc (bytes)	+ Children (bytes)	Routine
1	20.00	587569264	554450336	GP_MODEL@1
2	9.95	292161088	78809176	RADINTG@1
3	8.02	235681728	79457488	CALL_SL@1
4	6.52	191547280	0	>TRS-FTINV
5	6.25	183480072	370970264	RADDRV@1
6	4.92	144638640	0	>TRS-SULEG
7	4.72	138637064	134851248	SUSC2B@1
8	4.59	134851248	0	GMV_SUBS
9	4.39	128847896	128505720	SUTRANS@1
10	4.19	123052080	0	>TRS-LTINV

Dr. Hook memory profile for T799 forecast

#Calls	#Allocs	#Frees	Routine
120	723	720	GP_MODEL@1
24	268	268	RADINTG@1
120	1920	1917	CALL_SL@1
123	244	244	>TRS-FTINV
24	528	528	RADDRV@1
2	10	10	>TRS-SULEG
1	34	31	SUSC2B@1
1	2	0	GMV_SUBS
1	50	49	SUTRANS@1
123	242	242	>TRS-LTINV

Dr. Hook Traceback

```
0: 15:57:40 STEP 936 H= 234:00 +CPU= 41.379
13:[myproc#14,tid#4,pid#55924]: Received signal#24 (SIGXCPU) ; Memory: 2019178K (heap)
13:[myproc#14,tid#1,pid#55924]: MASTER ,#1,st=1,wall=0.000s/0.000s
13:[myproc#14,tid#1,pid#55924]: CNT0 ,#1,st=1,wall=0.000s/0.000s
13:[myproc#14,tid#1,pid#55924]: CNT1 ,#1,st=1,wall=0.000s/0.000s
13:[myproc#14,tid#1,pid#55924]: CNT2 ,#1,st=1,wall=0.000s/0.000s
13:[myproc#14,tid#1,pid#55924]: CNT3 ,#1,st=1,wall=0.000s/0.000s
13:[myproc#14,tid#1,pid#55924]: CNT4 ,#1,st=1,wall=0.000s/0.000s
13:[myproc#14,tid#1,pid#55924]: STEPO ,#978,st=1,wall=10531.259s/0.000s
13:[myproc#14,tid#1,pid#55924]: SCAN2H ,#1018,st=1,wall=8913.967s/0.043s
13:[myproc#14,tid#1,pid#55924]: SCAN2MDM ,#1018,st=1,wall=8913.896s/32.036s
13:[myproc#14,tid#1,pid#55924]: GP_MODEL ,#938,st=1,wall=8845.641s/4.830s
13:[myproc#14,tid#1,pid#55924]: EC_PHYS ,#213893,st=1,wall=6144.597s/22.378s
13:[myproc#14,tid#1,pid#55924]: CALLPAR ,#213893,st=1,wall=5856.788s/88.130s
13:[myproc#14,tid#1,pid#55924]: SLTEND ,#213893,st=1,wall=662.390s/179.559s
13:[myproc#14,tid#1,pid#55924]: CUADJTQ ,#117188599,st=1,wall=1992.364s/477.382s

13: Signal received: SIGXCPU - CPU time limit exceeded
13:
13: Traceback:
13: Location 0x0000377c
13: Offset 0x0000009c in procedure _event_sleep
13: Offset 0x00000318 in procedure sigwait
13: Offset 0x000006c8 in procedure pm_async_thread
13: Offset 0x000000a4 in procedure _pthread_body
13: --- End of call chain ---
```



Bit Reproducibility

- **What?**

Getting IFS applications to produce bit identical results when,

- A) The application is rerun on the same number of tasks and threads
- B) The number of threads is changed
- C) The number of tasks is changed

- **Why?**

- Non-reproducibility often hides a bug (application or compiler)

- **How?**

- Example: $X + Y + Z \neq X + Z + Y$
- Computations must be performed in the same order as if they would be done using a single task with one thread
- By application design (all the spaces, grid, spectral, obs, control vector, ...)
- Compiler support (e.g. IBM `-qstrict`)

Bit Reproducibility

- IFS model

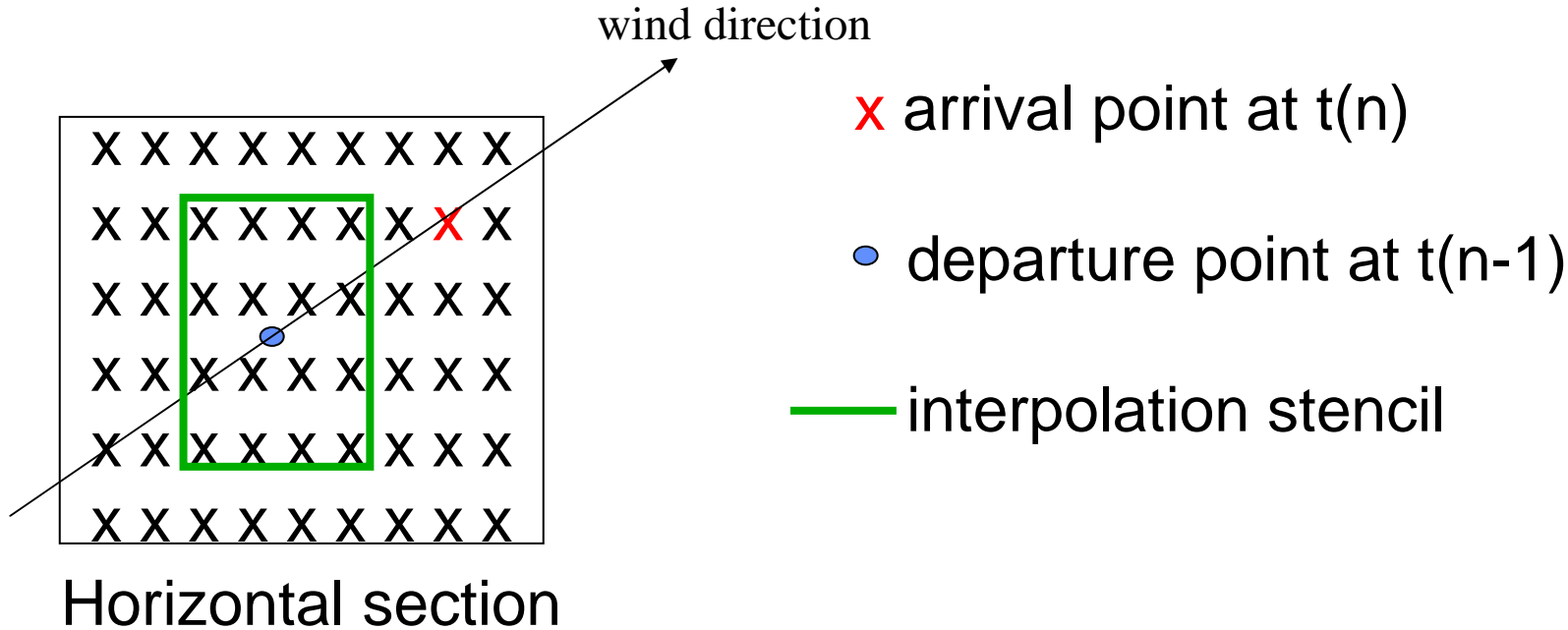
- A,B,C reproducibility (default)

- IFS 4D-VAR

- A,B reproducibility (default)
- C reproducibility (with namelist setting LREPRO4DVAR=T)
 - Costs 10% more than with LREPRO4DVAR=F
 - Support for vector/scalar architectures (LVECADIN=T/F)

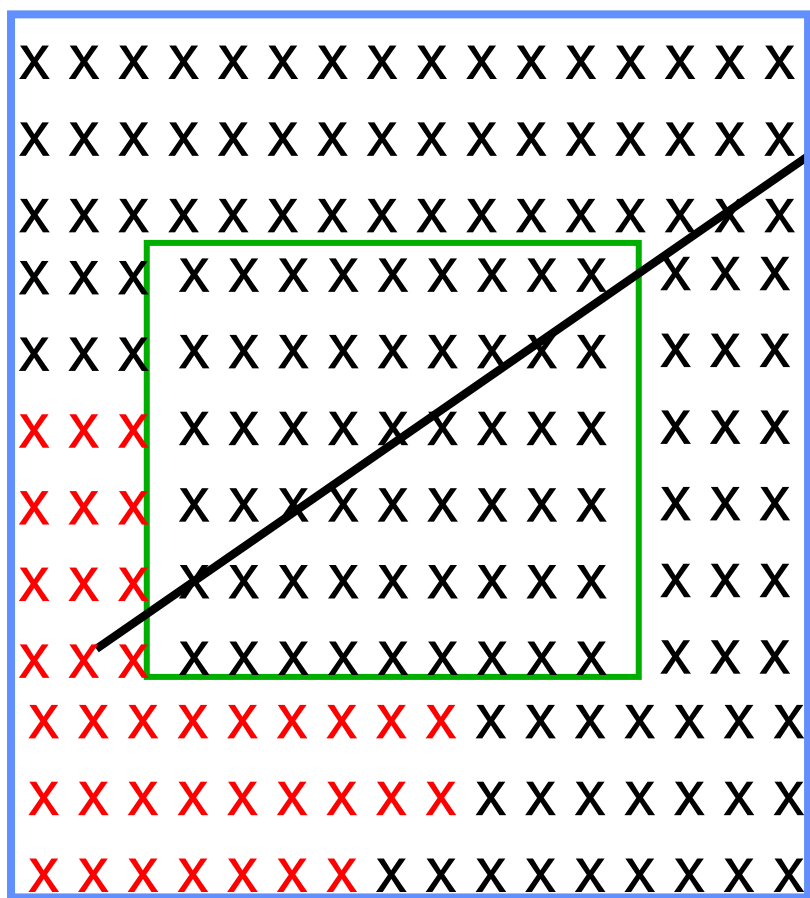
- Testing must be done for each new cycle!

IFS - Semi-Lagrangian Advection



Full interpolation in 3-D is 32 point

IFS - Semi-Lagrangian 'On Demand'

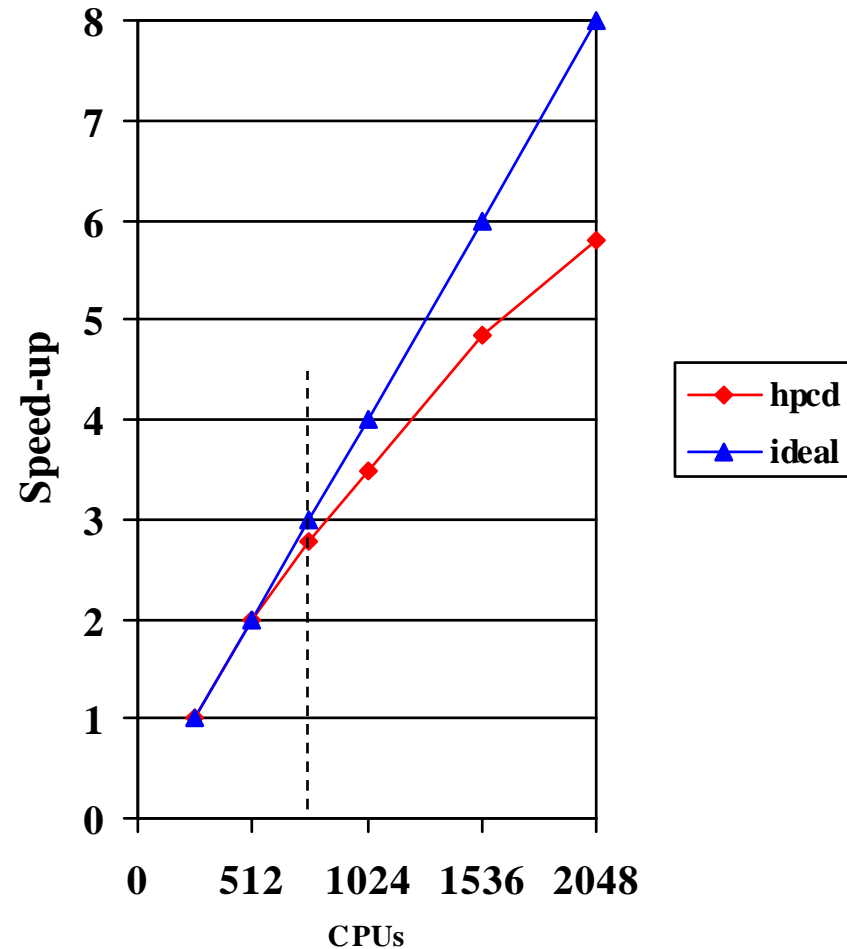


- x points needed in halo
- points located on task n
- halo

IFS partitions are irregular shaped due to 2D partitioning of gaussian grid (and not like in this simple diagram)

IFS RAPS-8: T799 L91 10 day forecast on p690+ (hpcd)

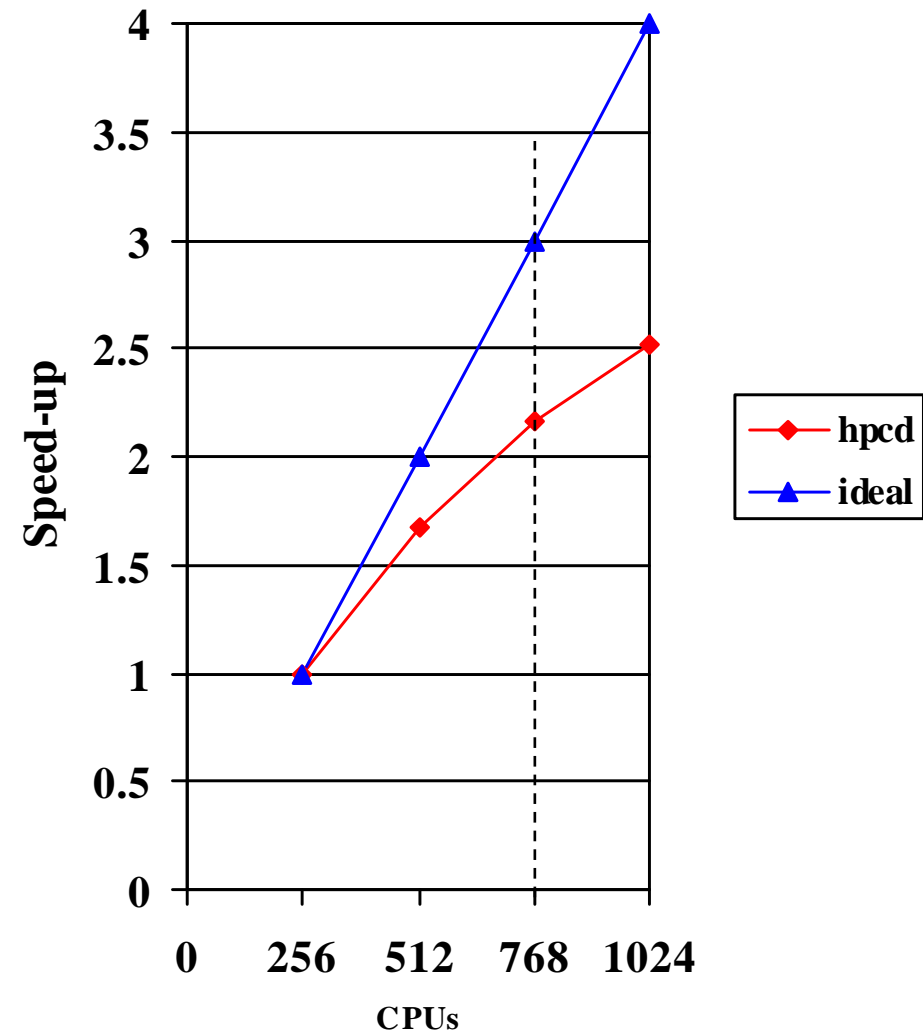
MPI × OpenMP	Wall (secs)	Gflops
64×4	8850	193
128 × 4	4410	369
192 × 4	3187	509
256 × 4	2534	644
384 × 4	1830	886
256 × 8	1523	1073



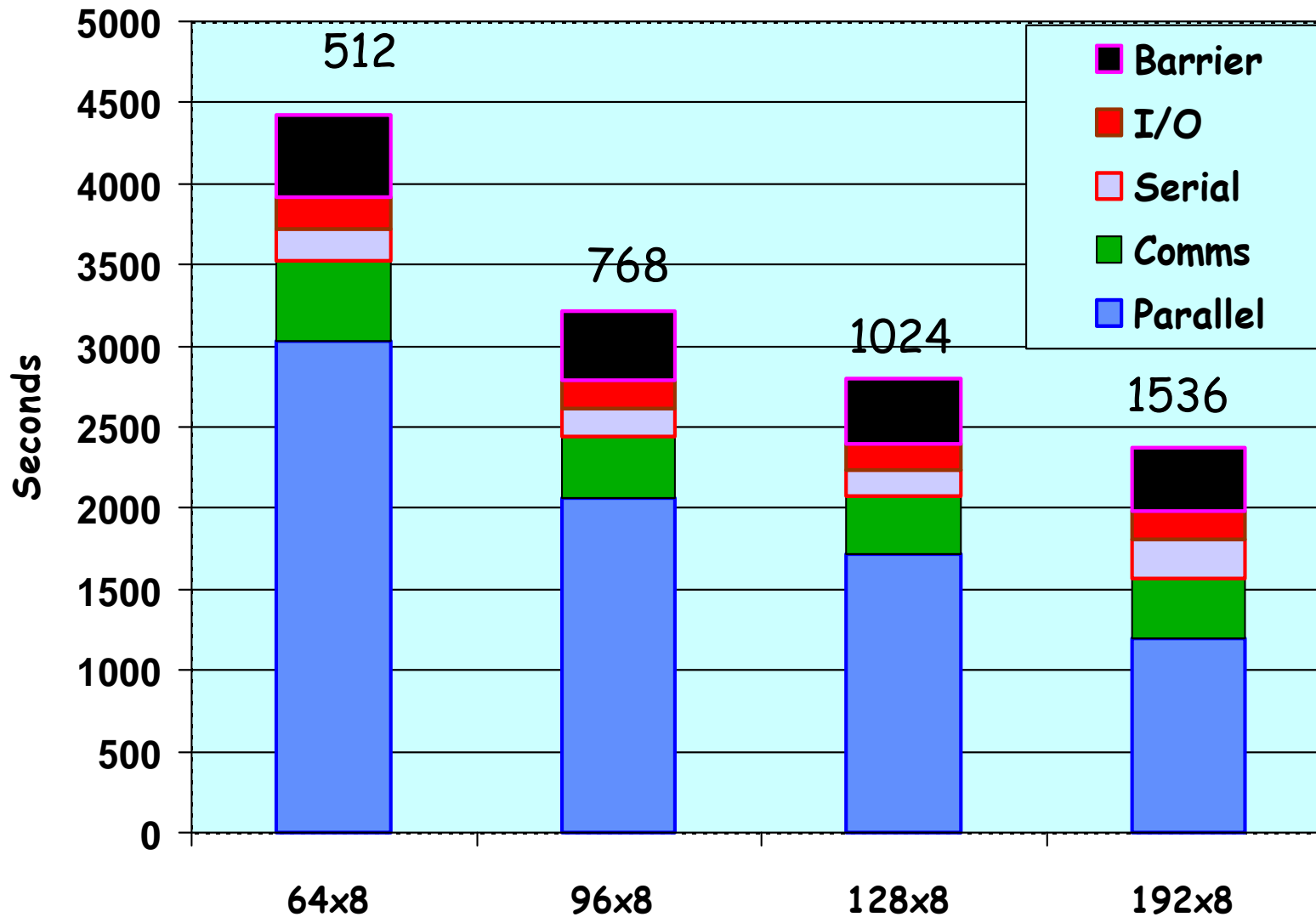
IFS RAPS-8: 4D-Var run on p690+ (hpcd)

T799L91 / T95 / T255

MPIx OpenMP	TOTAL (secs)	% of peak
64 x 4	5950	7.9%
128 x 4	3547	6.7%
96 x 8	2738	5.7%
128 x 8	2359	5.0%



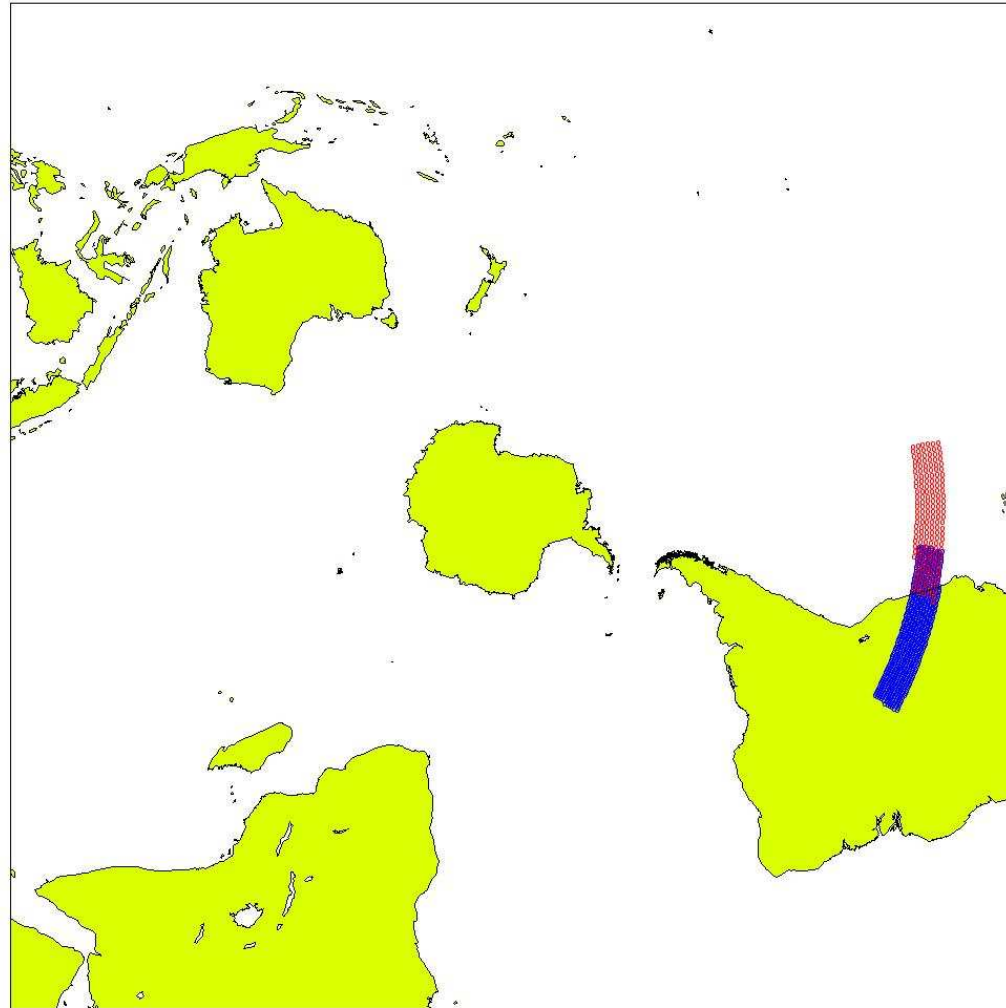
4D-Var T799/T95/T255 L91 on p690+ hpcd (with barriers for more accurate timing)



T_L511 Model / T_L255 Radiation Grid

- Radiation computations are expensive
- To save on cost we
 - Run radiation computations every hour (every 4th time step)
 - Run radiation computations on a courser grid
 - Requires interpolation
- Two interpolation possibilities
 - Gather global fields to different tasks (non-scalable)
 - Perform interpolation with only local communication for halo (scalable)
 - But there was a problem ...

Example, 512 tasks, task=293

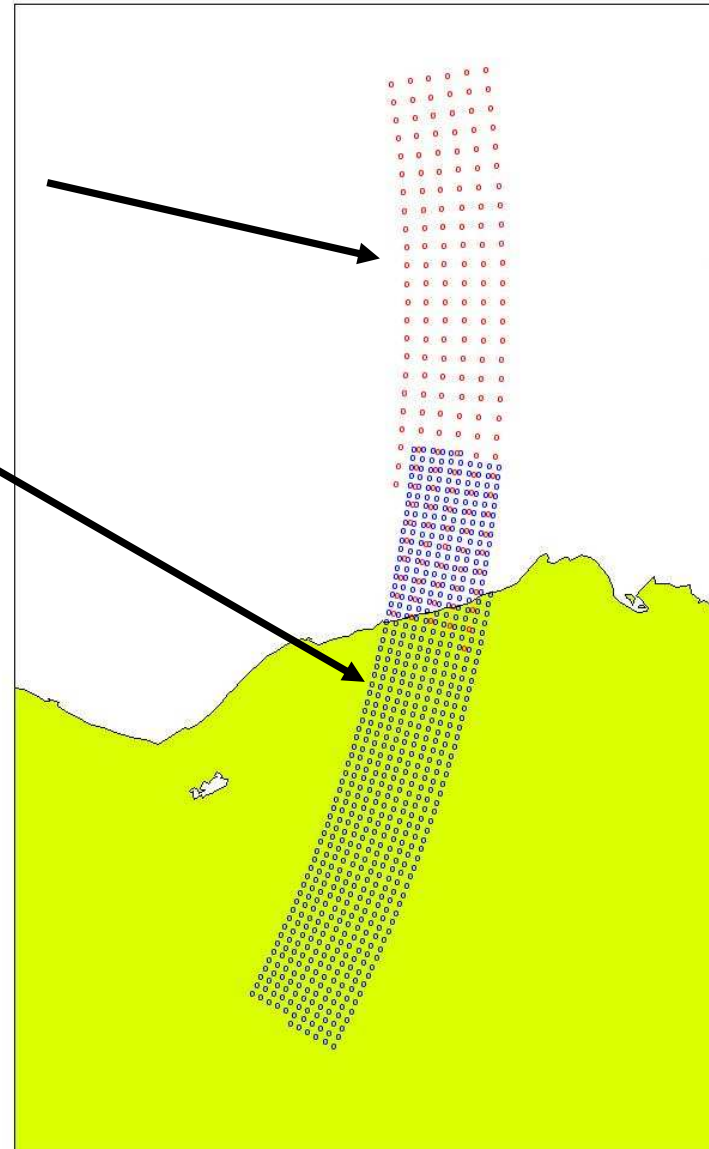


PE=293, Radiation Grid T_L255

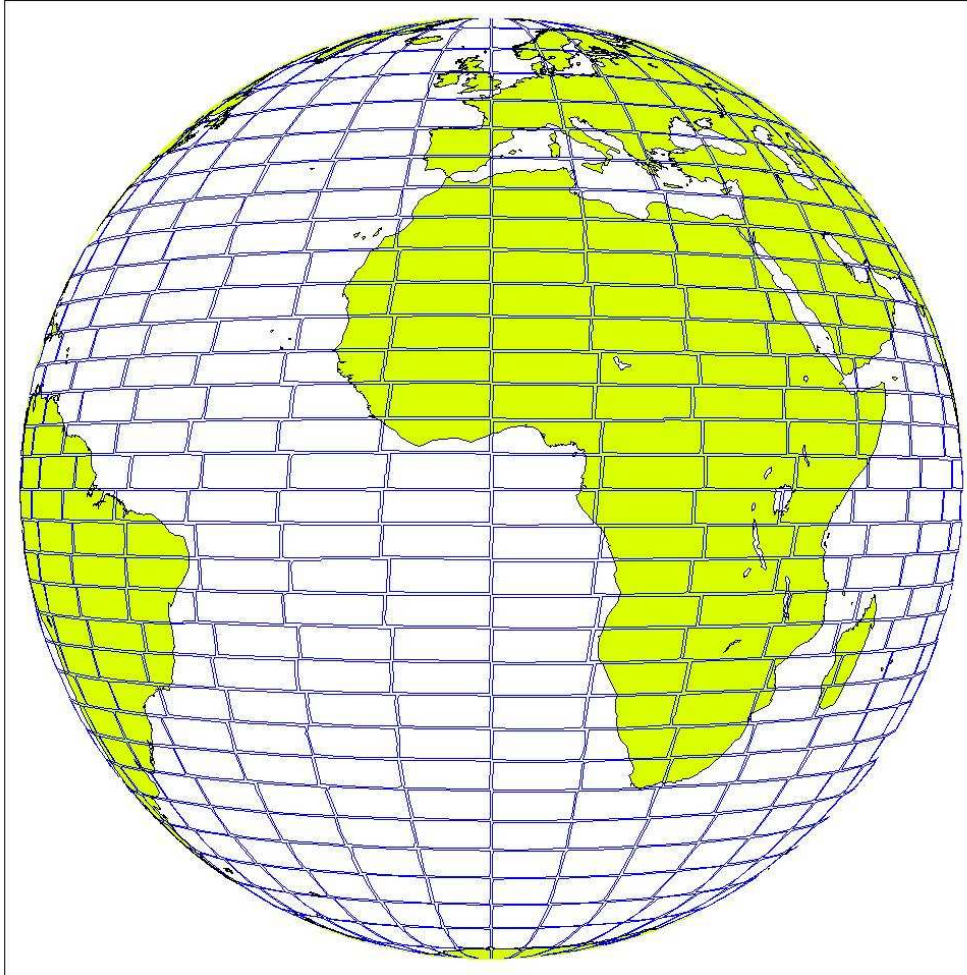
PE=293, Model Grid T_L511

Model and Radiation grids for same partition are offset geographically, because

- Use of Gaussian Grid (Linear)
- T_L255 is not a projection of T_L511
- Long thin partitions make matters worse (need for new partitioning strategy?)

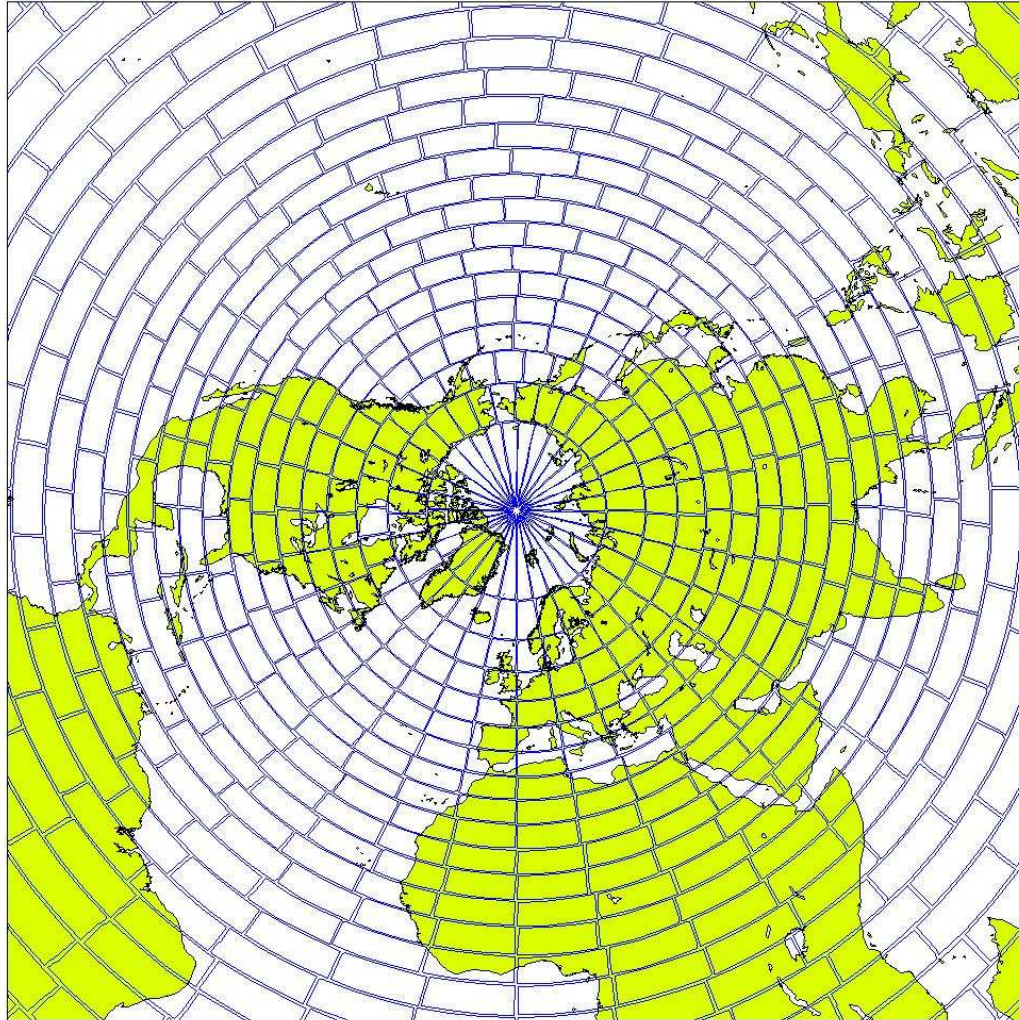


T_L799 1024 task 2D partitioning c.1994



Of course, because IFS uses OpenMP on IBM HPCF we **only** need **256 tasks** x 4 threads today!

T_{L799} 1024 Tasks



2D partitioning
results in non-optimal
Semi-Lagrangian
comms requirement
at poles and equator!

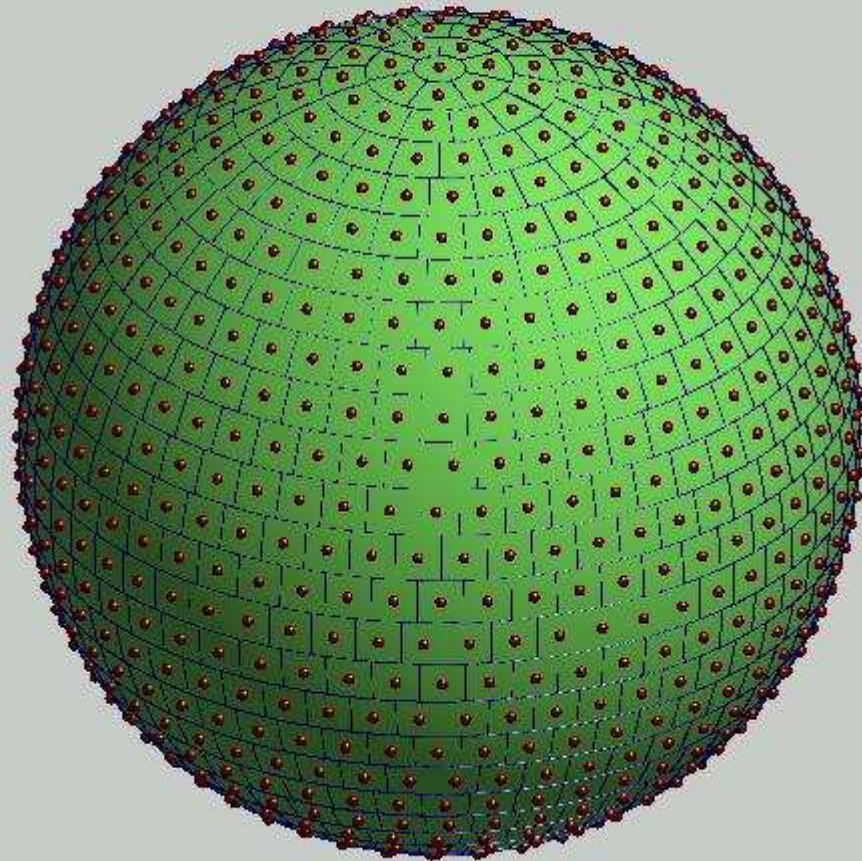
Square is best!

EQ_REGIONS algorithm
developed by Paul
Leopardi et al. , Univ. of
New South Wales

To be implemented in
IFS 2H2006.

Above EQ_REGIONS
algorithm will be adapted
for IFS gaussian grid to
give ideal load balance of
grid points per partition.

Recursive zonal equal area partition of S^2
into 1024 regions, showing the center point of each region.





QUESTIONS?