Data Assimilation on future computer architectures

Lars Isaksen

ECMWF

Acknowledgements to ECMWF colleagues:

Deborah Salmond, George Mozdzynski, Mats Hamrud, Mike Fisher, Yannick Trémolet, Jean-Noël Thepaut, Anne Fouilloux and Massimo Bonavita

and also to a set of computer vendors

Data Assimilation on future computer architectures

("The future" is considered to be something like 2020)

- Data Assimilation scalability issues on todays computer architectures - using 4D-Var at ECMWF as an example
- How will the future computer architectures look?
- Will we be able to use future parallel computers efficiently for Data Assimilation?
- Can we modify our Data Assimilation methods to utilize future computer architectures better?

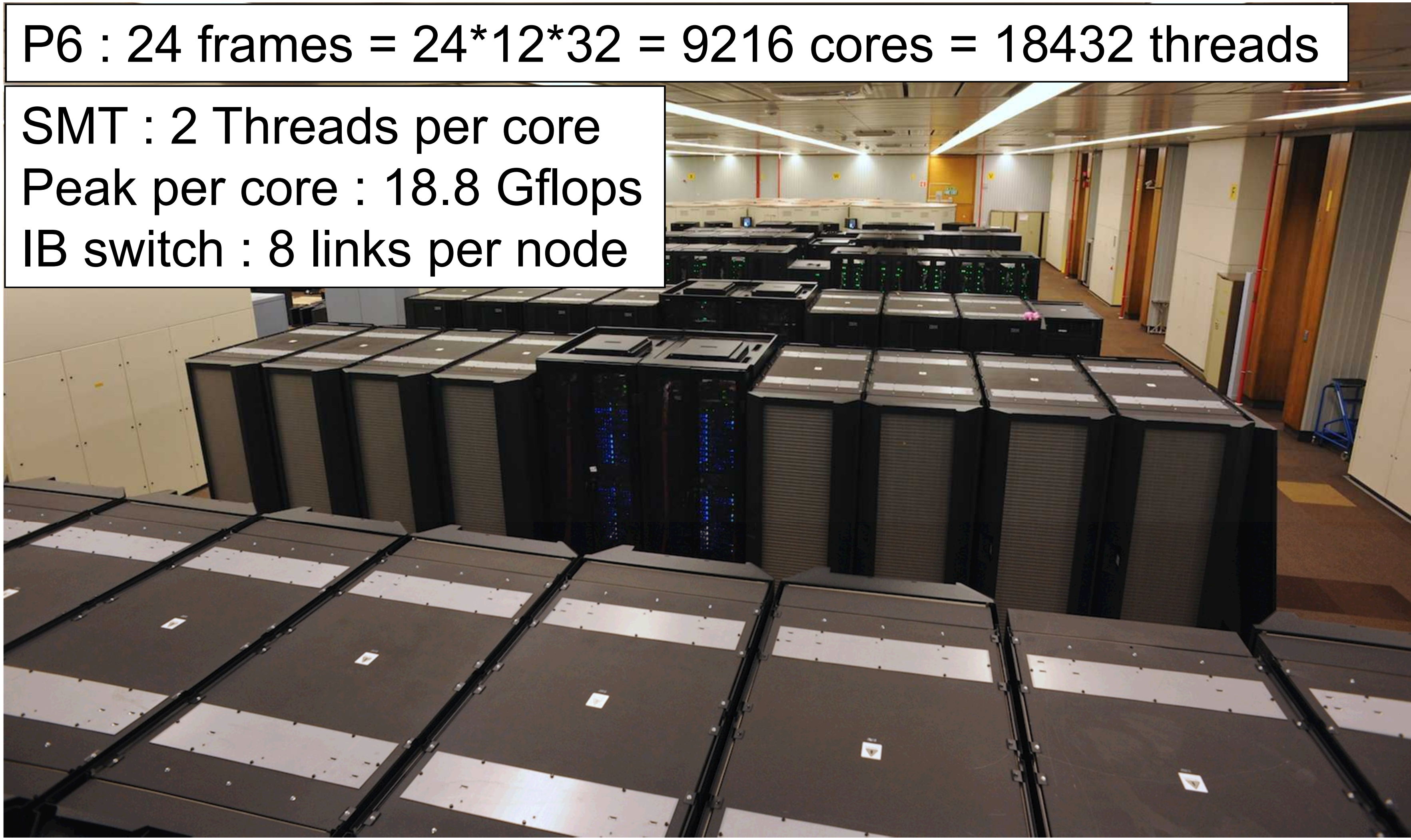
- ECMWF HPC systems
  - At the moment IBM Power6 (2x9200 cores)
  - Will soon be upgraded to IBM Power7 (2x24400 cores)

- Operational Forecast and 4D-Var assimilation configuration
  - We are using the IFS - Integrated Forecast System
  - 10-day T1279L91 Forecast (16 km horizontal grid)
  - 12 hour 4D-Var T1279 outer loop T255/T159 inner loop
  - Operational Ensemble of Data Assimilations (EDA)
    - 10 member 4D-Var T399 outer and T95/T159 inner loop

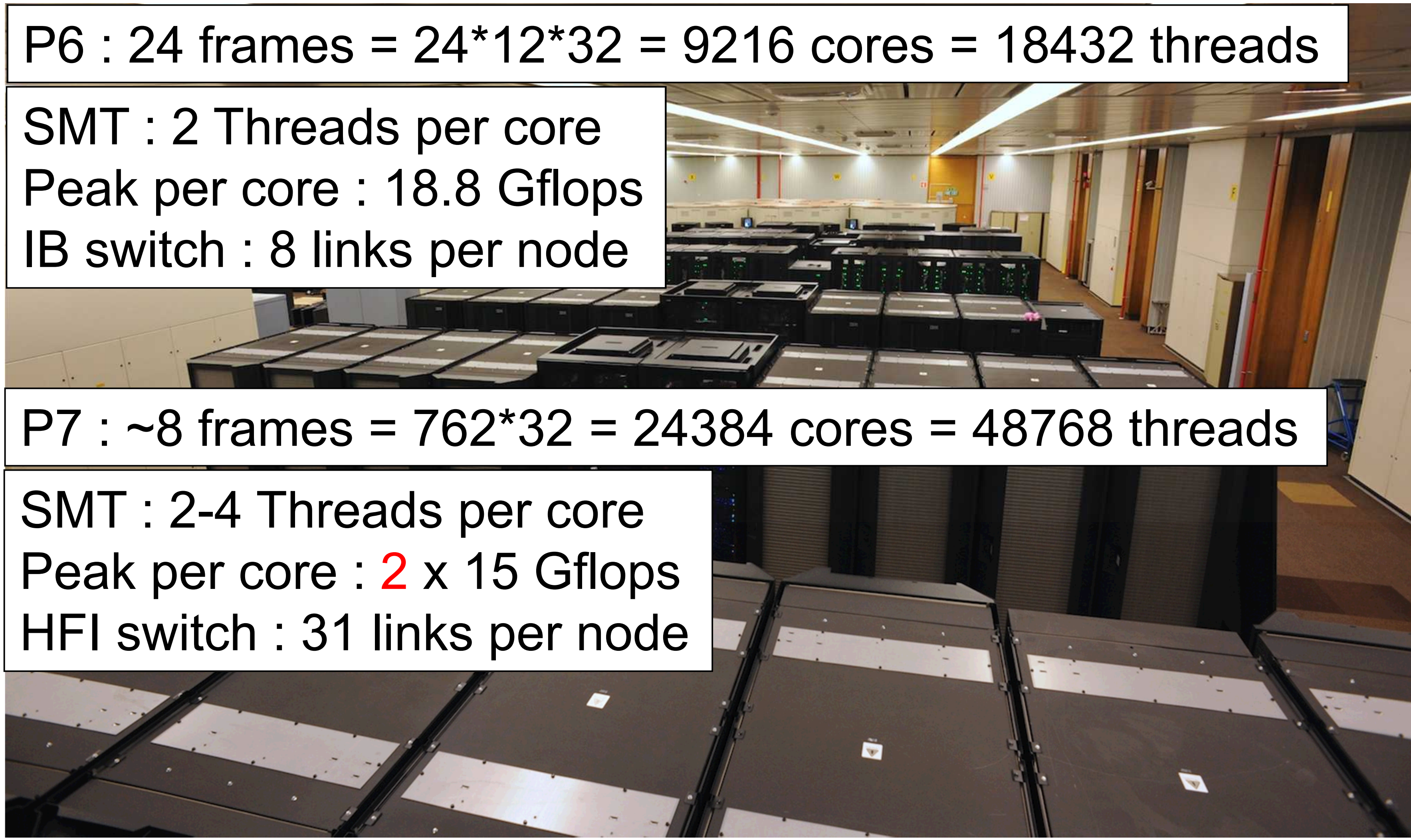
One of ECMWF’s two IBM Power6 clusters

P6 : 24 frames = 24*12*32 = 9216 cores = 18432 threads

SMT : 2 Threads per core
Peak per core : 18.8 Gflops
IB switch : 8 links per node



2011/12 : ECMWF will install two Power7 clusters



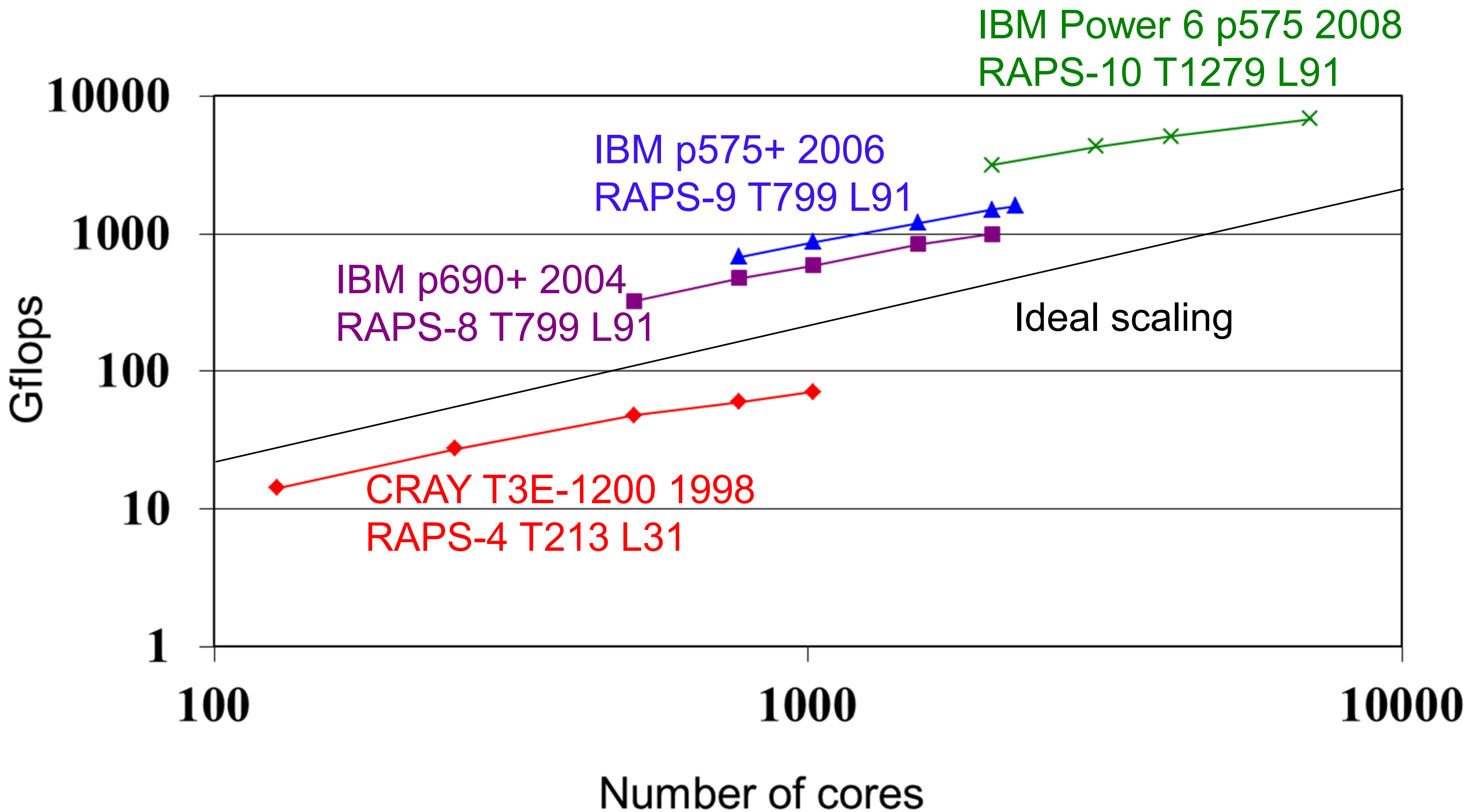
P6 : 24 frames = 24*12*32 = 9216 cores = 18432 threads

SMT : 2 Threads per core
Peak per core : 18.8 Gflops
IB switch : 8 links per node

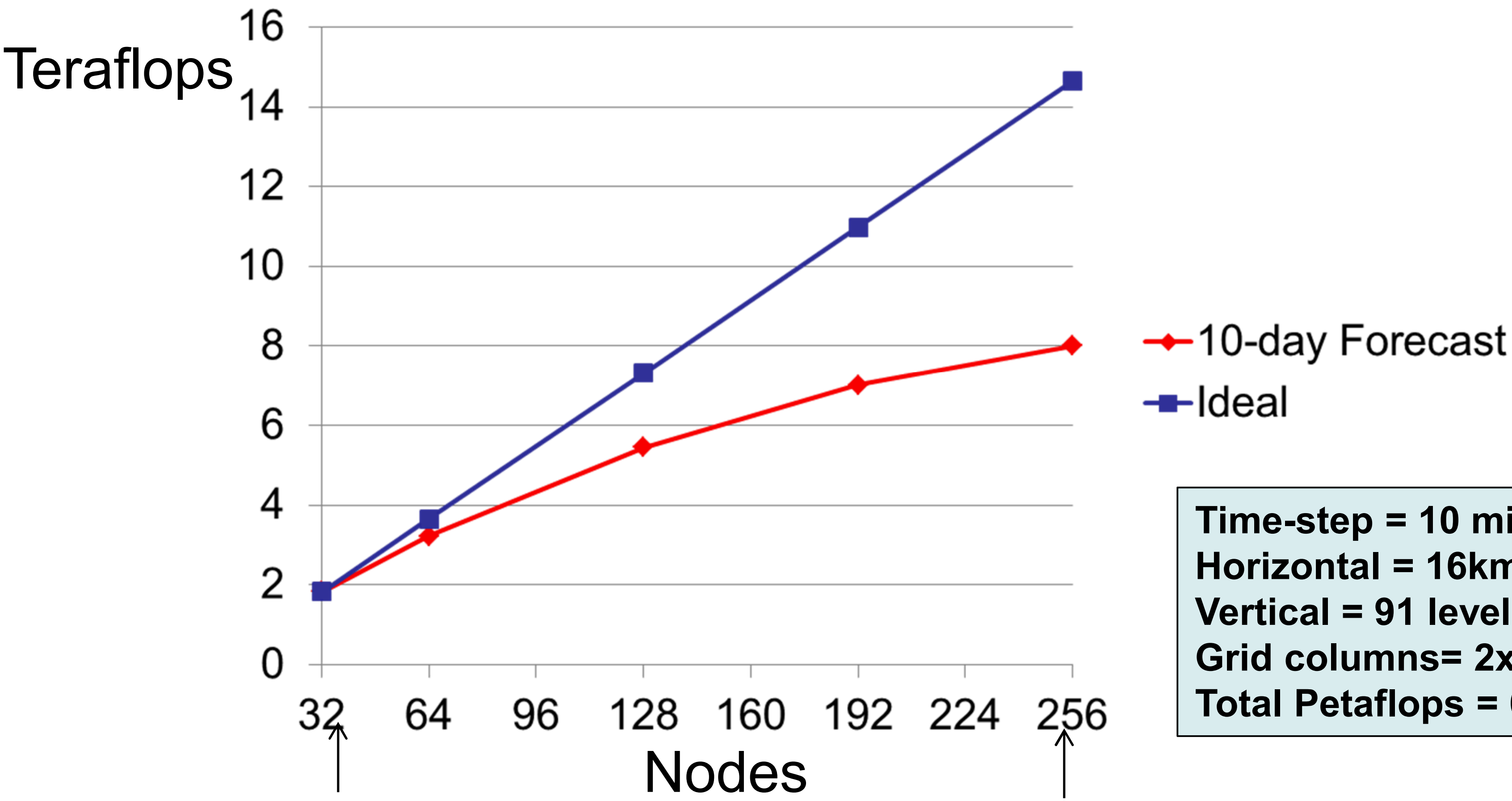
P7 : ~8 frames = 762*32 = 24384 cores = 48768 threads

SMT : 2-4 Threads per core
Peak per core : 2 x 15 Gflops
HFI switch : 31 links per node

History of IFS scalability



T1279 L91 Forecast runs up to whole IBM P6 cluster

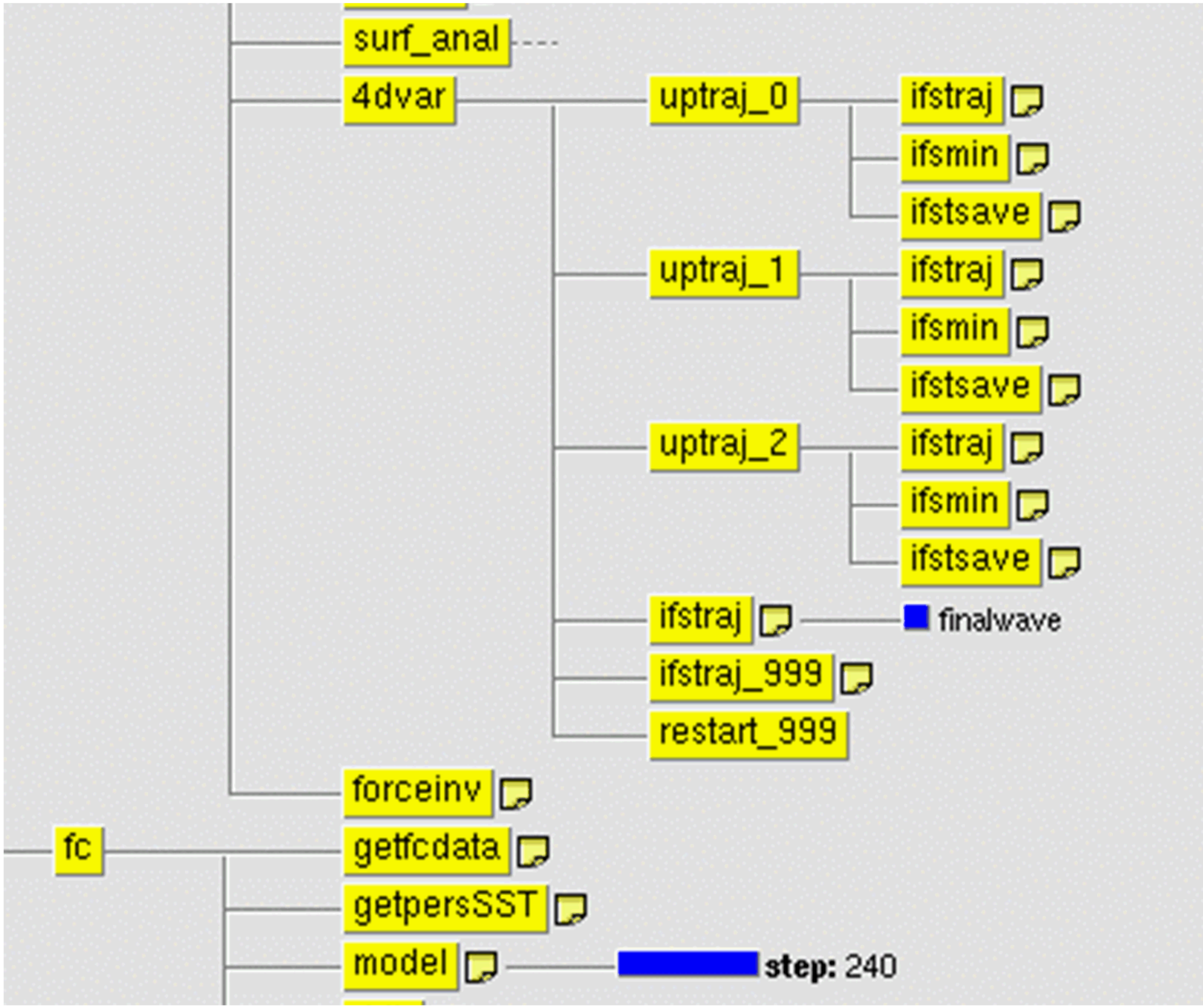


Time-step = 10 min
Horizontal = 16km
Vertical = 91 levels
Grid columns= $2x10^6$
Total Petaflops = 6.5

9% peak
2048 threads

5% peak
16384 threads

4D-Var and 10-day forecast



4D-var time window is 12 hours

Forecast & Outer loop trajectories:

(Traj_0,1,2) are using T1279 resolution
Grid columns = 2×10^6

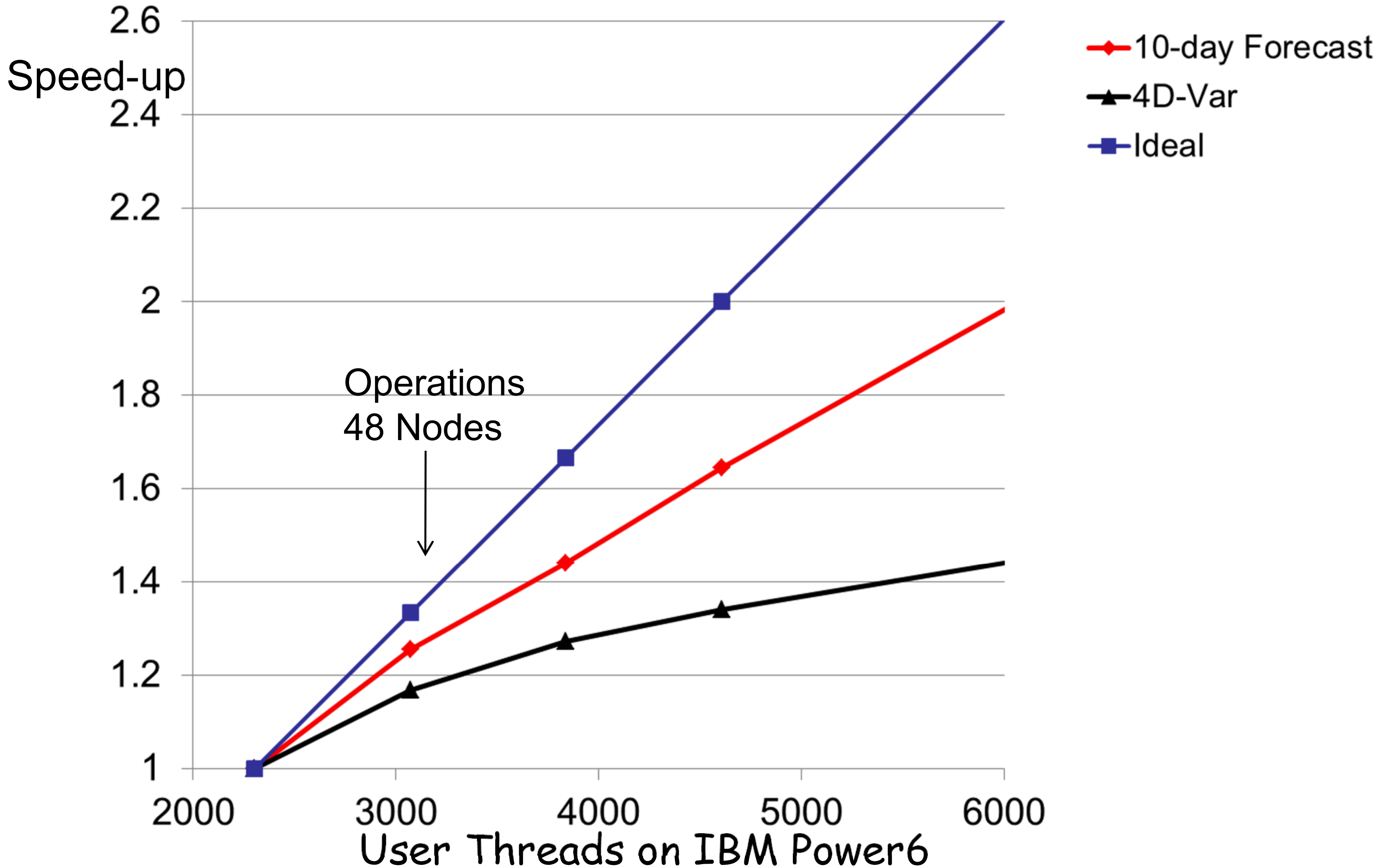
Three minimizations:

Min_0 : T159
Grid columns = 36000

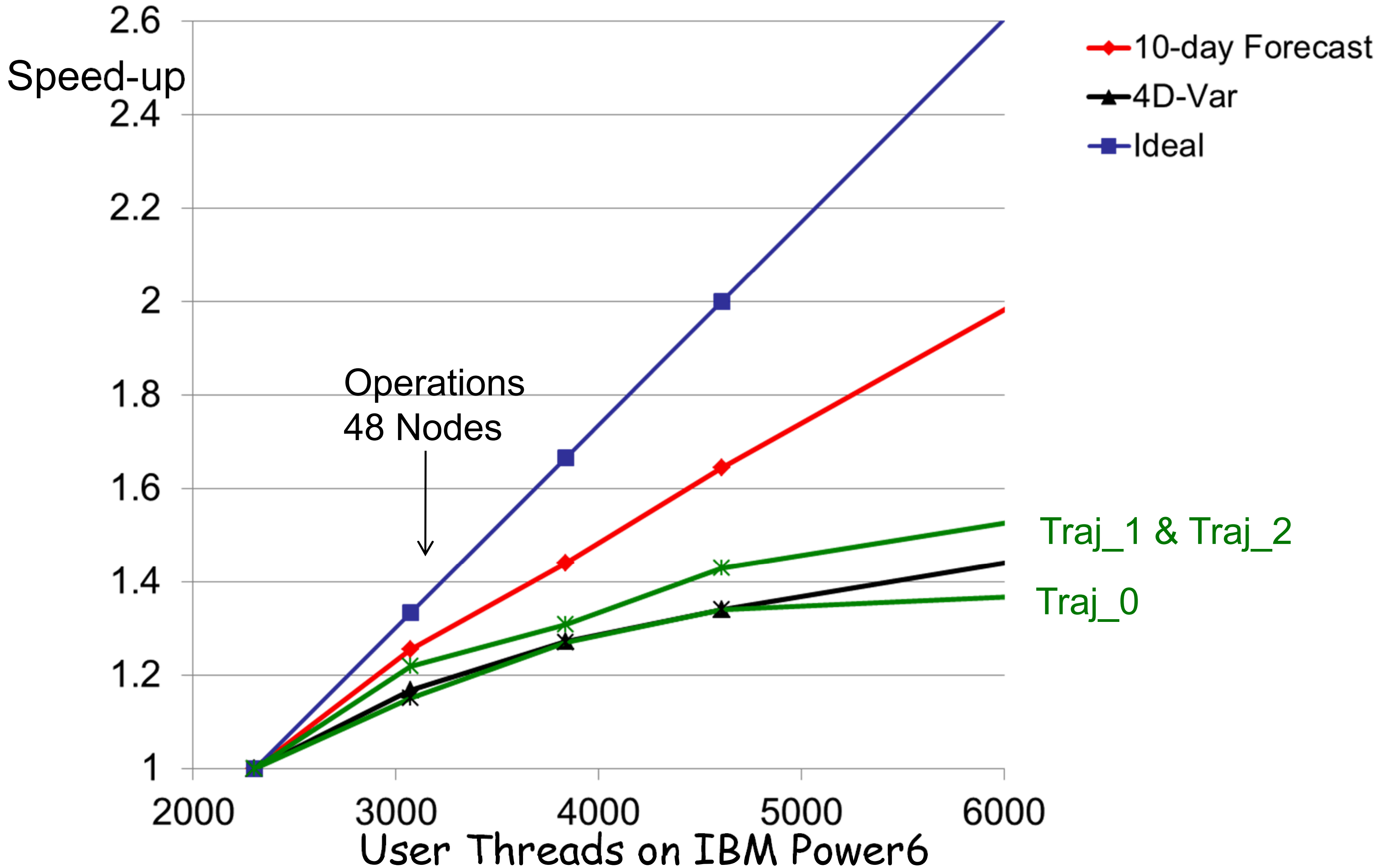
Min _1 & 2 : T255
Grid columns = 89000

Vertical = 91 levels

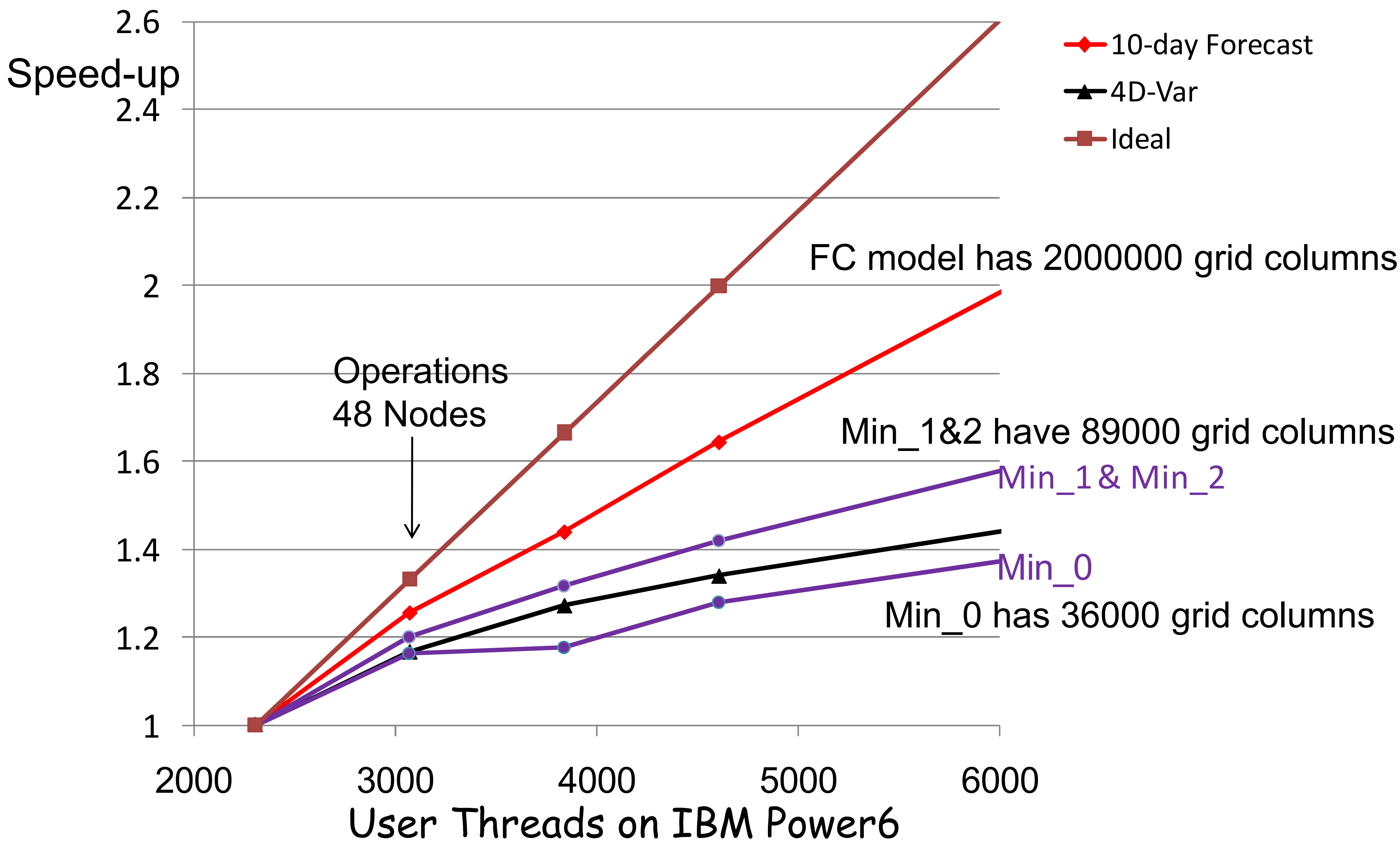
Scalability of T1279 Forecast and 4D-Var



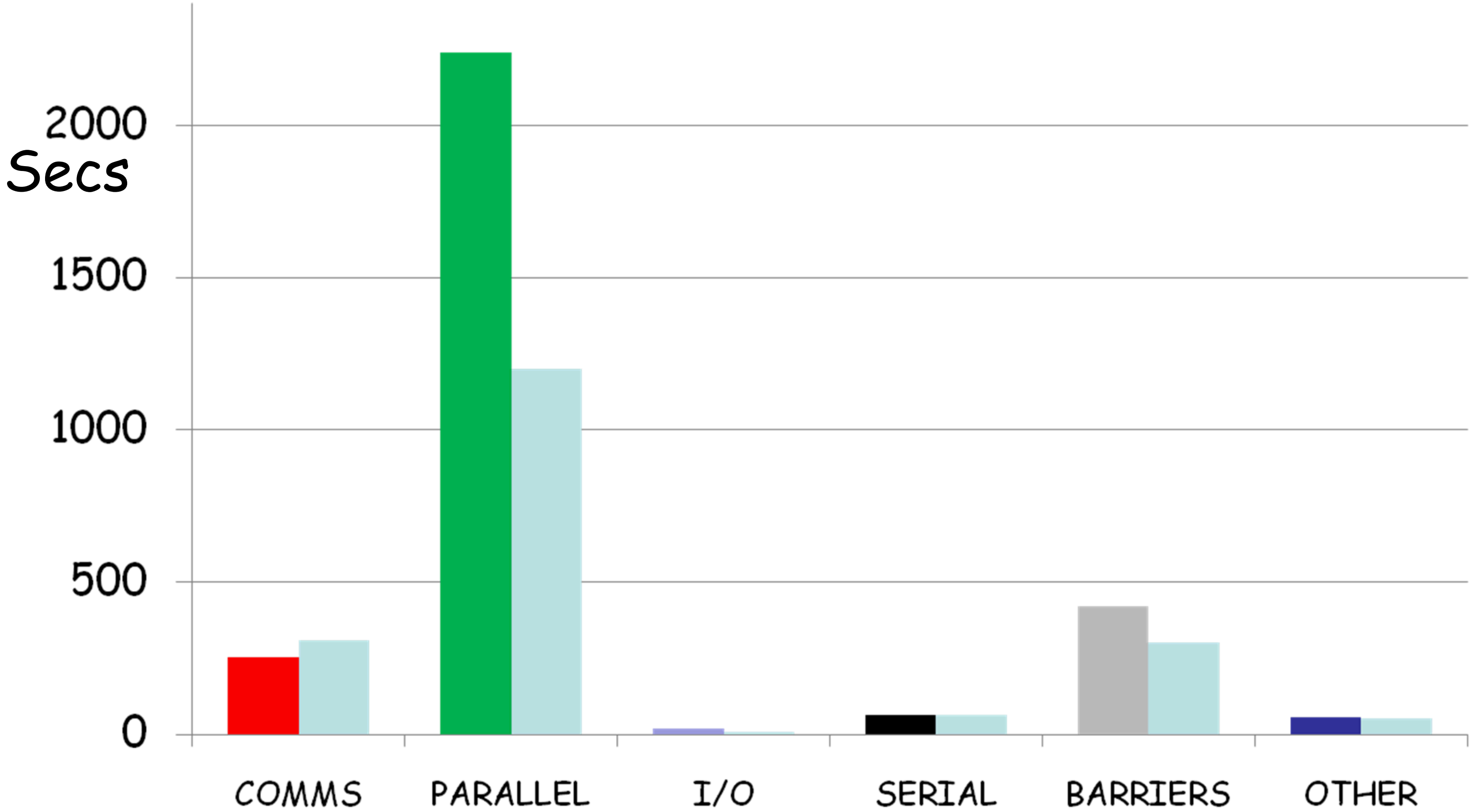
Scalability of T1279 Forecast and 4D-Var



Scalability of T1279 Forecast and 4D-Var



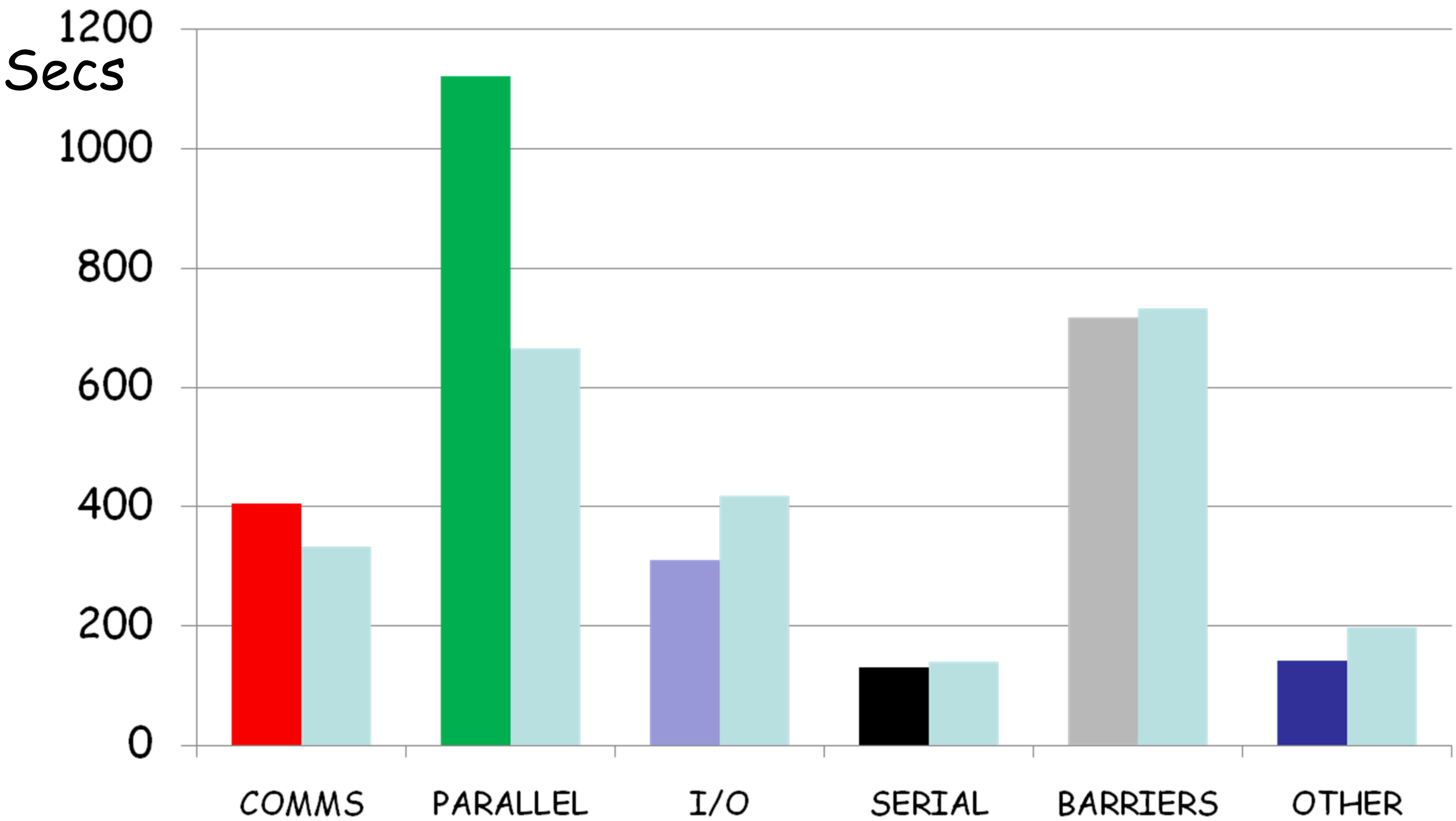
Scalability of T1279 Forecast: 48 to 96 nodes



* BARRIERS inserted for timing purposes = load imbalance + jitter

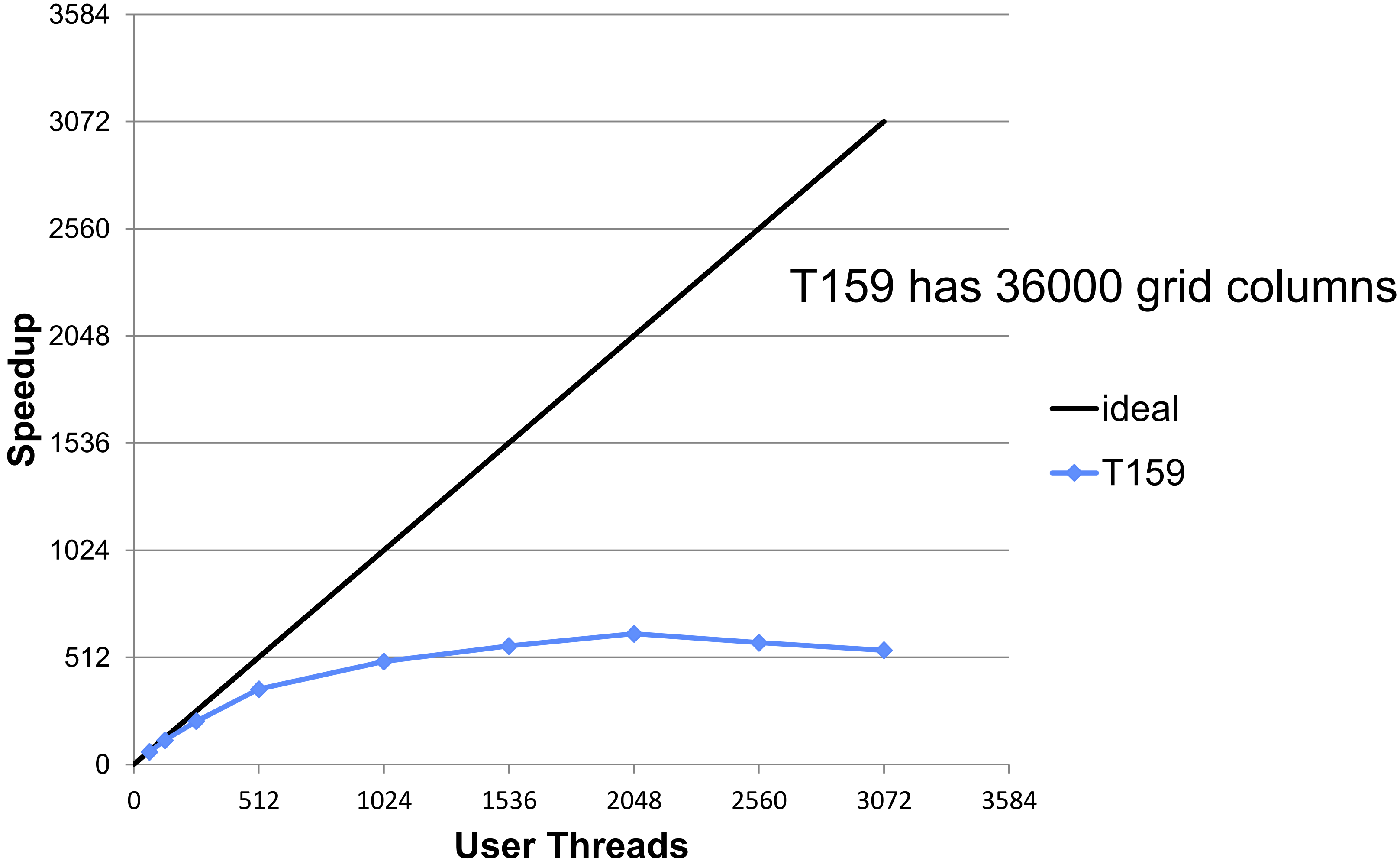
ECMWF

Scalability of 4D-Var: 48 to 96 nodes

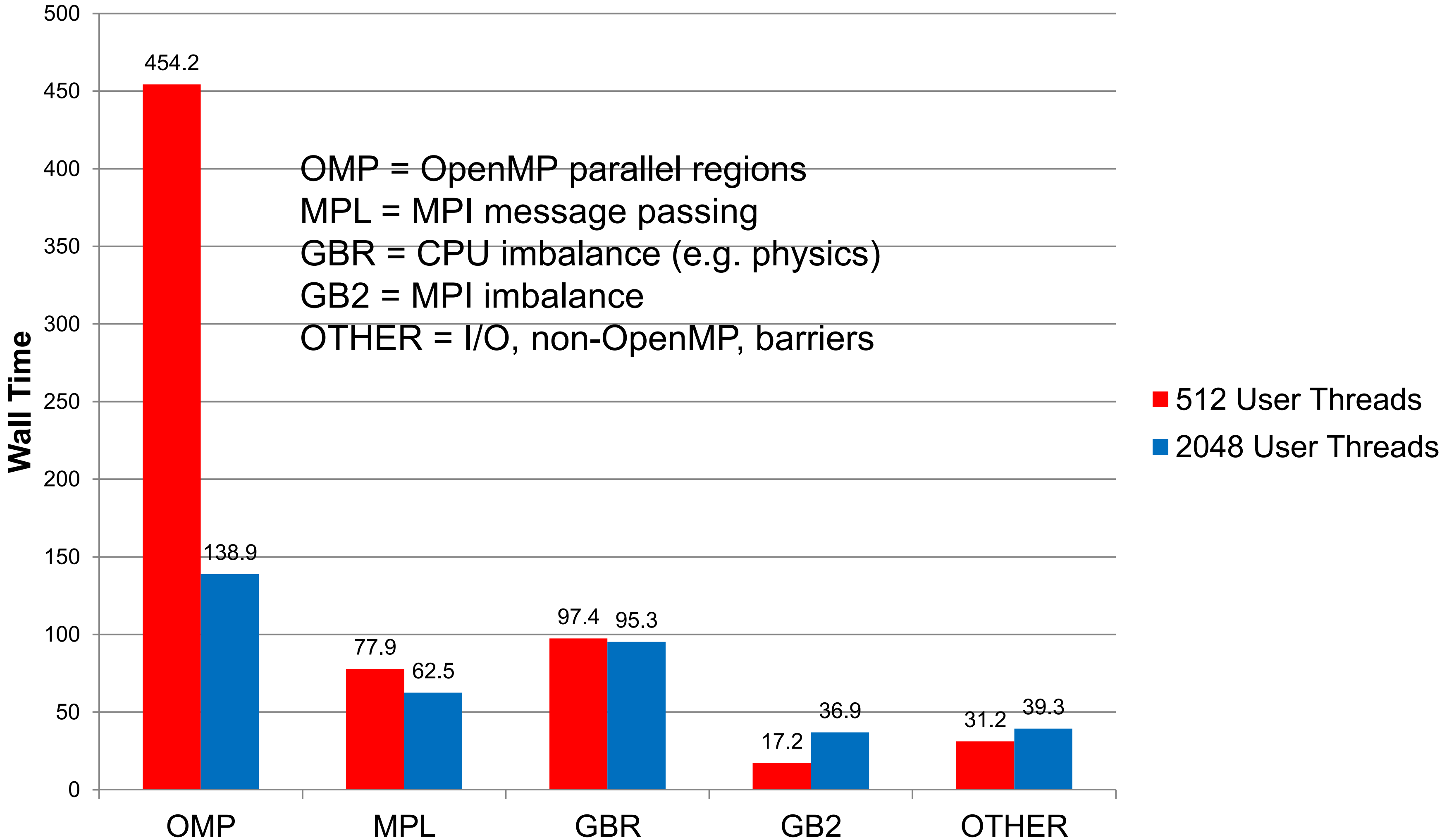


* Several types of I/O including Observational Data Base

T159 model scaling: small model with ‘large’ number of user threads



T159 model – scalability



ECMWF

Future HPC architectures

Hardware and Software issues

Challenges in Application Scaling In an Exascale Environment

14th Workshop on the Use of
High Performance Computing
In Meteorology

November 2, 2010

Dr Don Grice
IBM

Fujitsu's Approach
to
Application Centric Petascale Computing

2nd Nov. 2010

Motoi Okuda
Fujitsu Ltd.

The next-generation supercomputer and NWP system of JMA

Masami NARITA, Keiichi KATAYAMA

Numerical Prediction Division,
Japan Meteorological Agency

Using GPUs to Run Weather Prediction Models

Mark Govett
Tom Henderson, Jacques Middlecoff,
Paul Madden, Jim Rosinski

Environment Canada Environnement Canada

Canada

HPC at the Canadian Meteorological Centre

Luc Corbeil

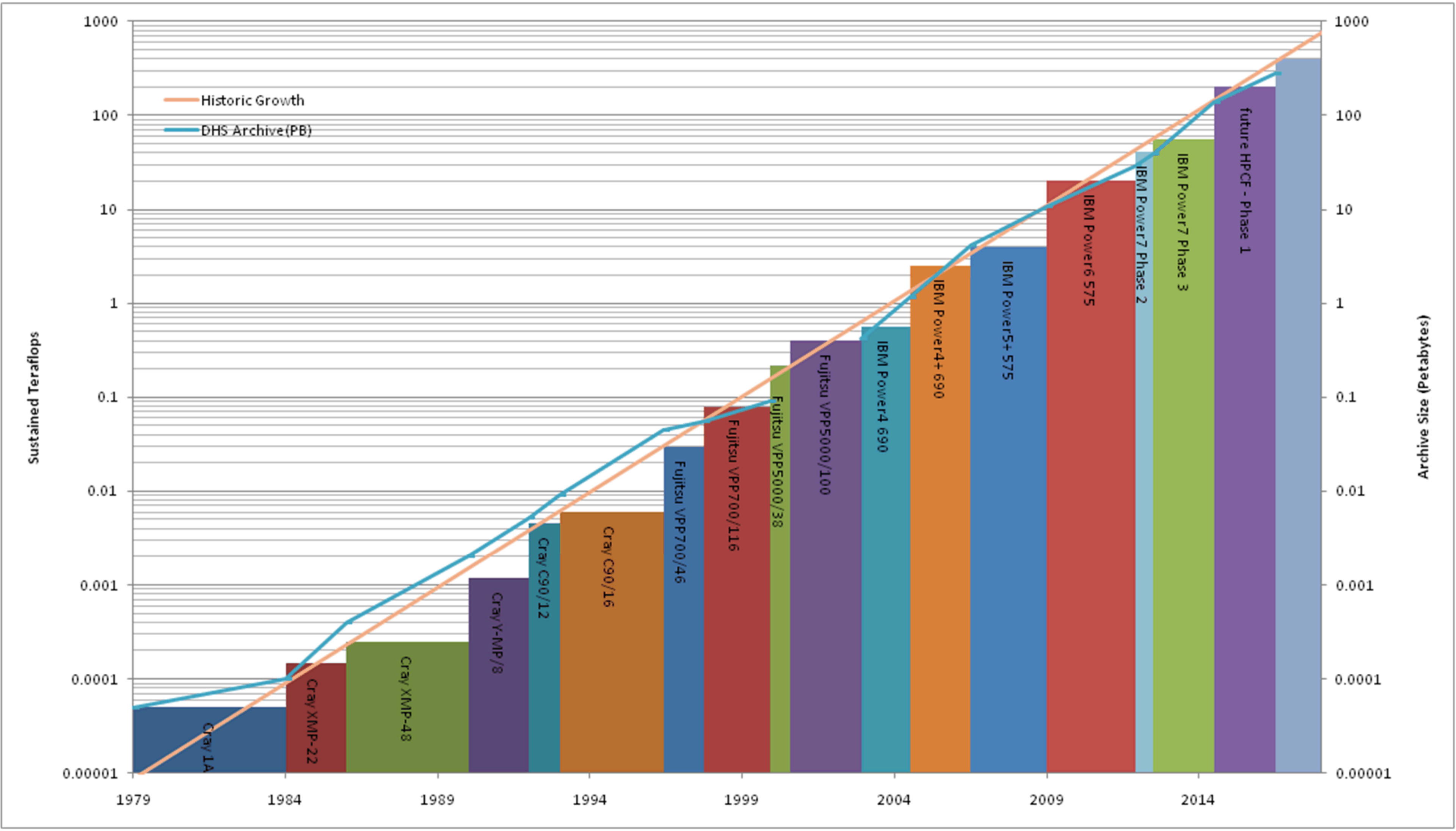
Chief, Supercomputer, Systems and Storage

Bertrand Denis

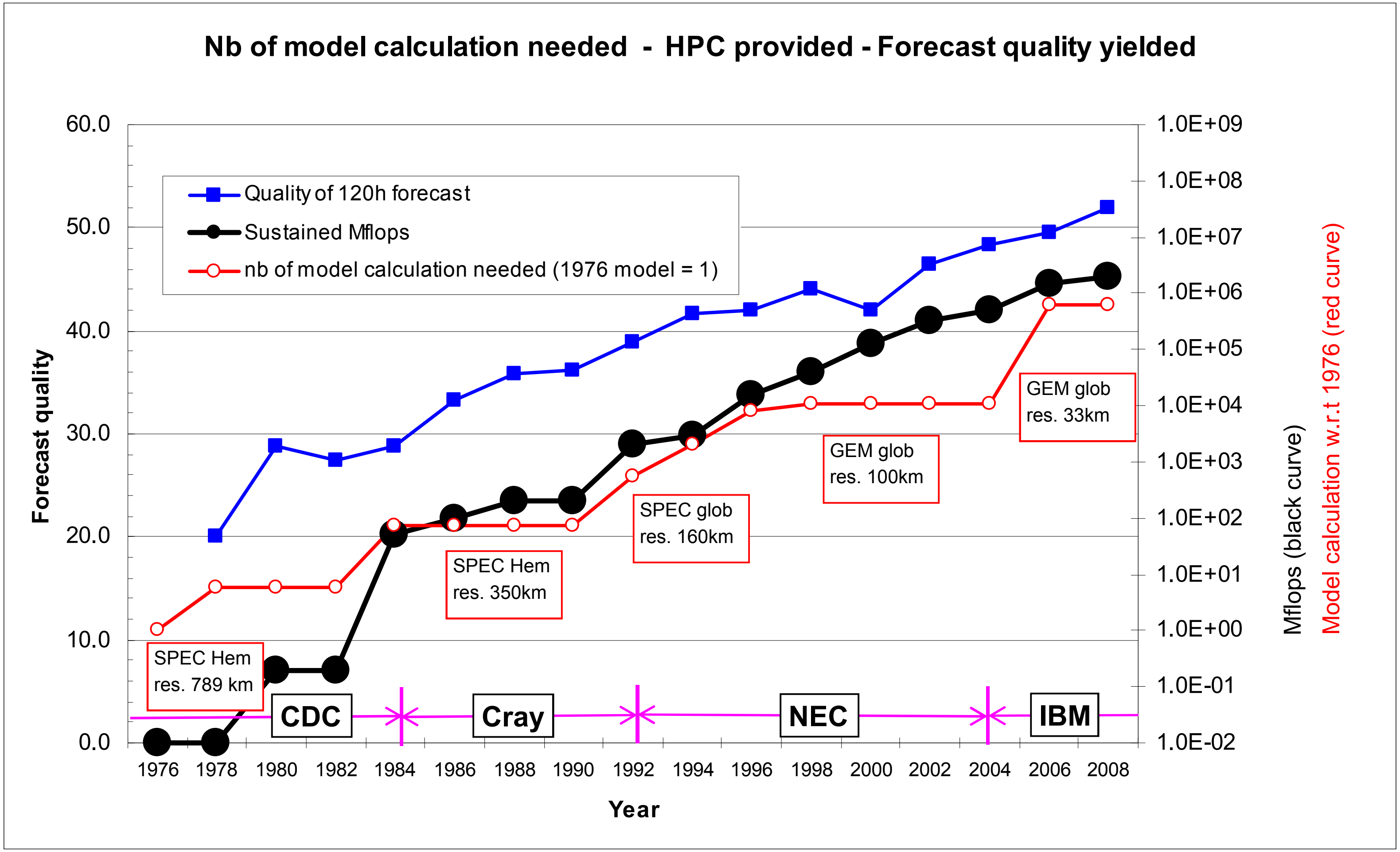
Chief, Numerical Weather Prediction Section

Fourteenth Workshop on Use of High Performance Computing in Meteorology
1 – 5 November 2010, ECMWF, Reading, UK

ECMWF sustained historic computer growth

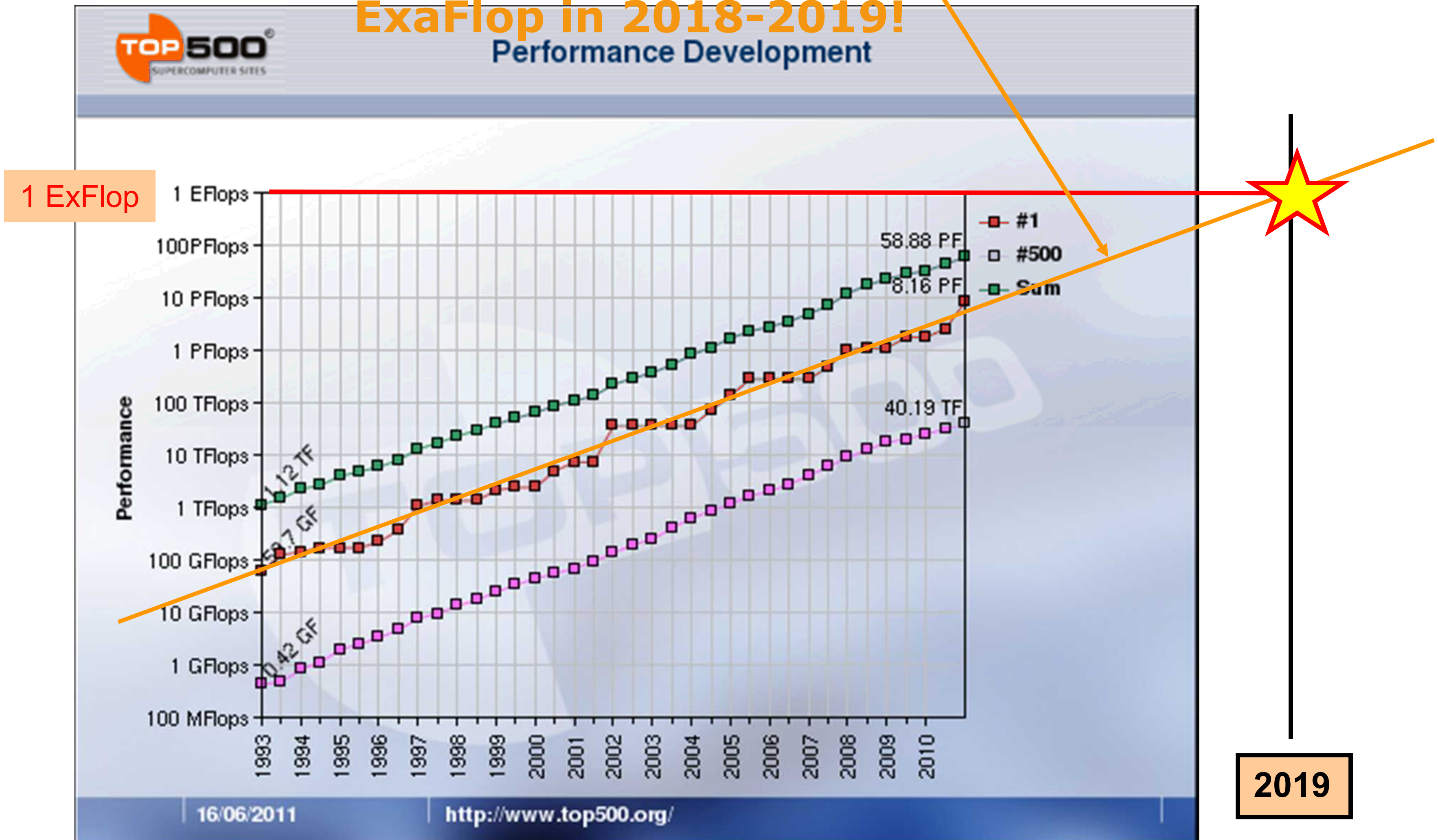


Historical HPC evolution and forecast quality at CMC



Growth of the Largest Computers by year
1000x every 10-11 years

ExaFlop in 2018-2019!



(Slide updated by me)

Clock speed and power per chip has stopped increasing

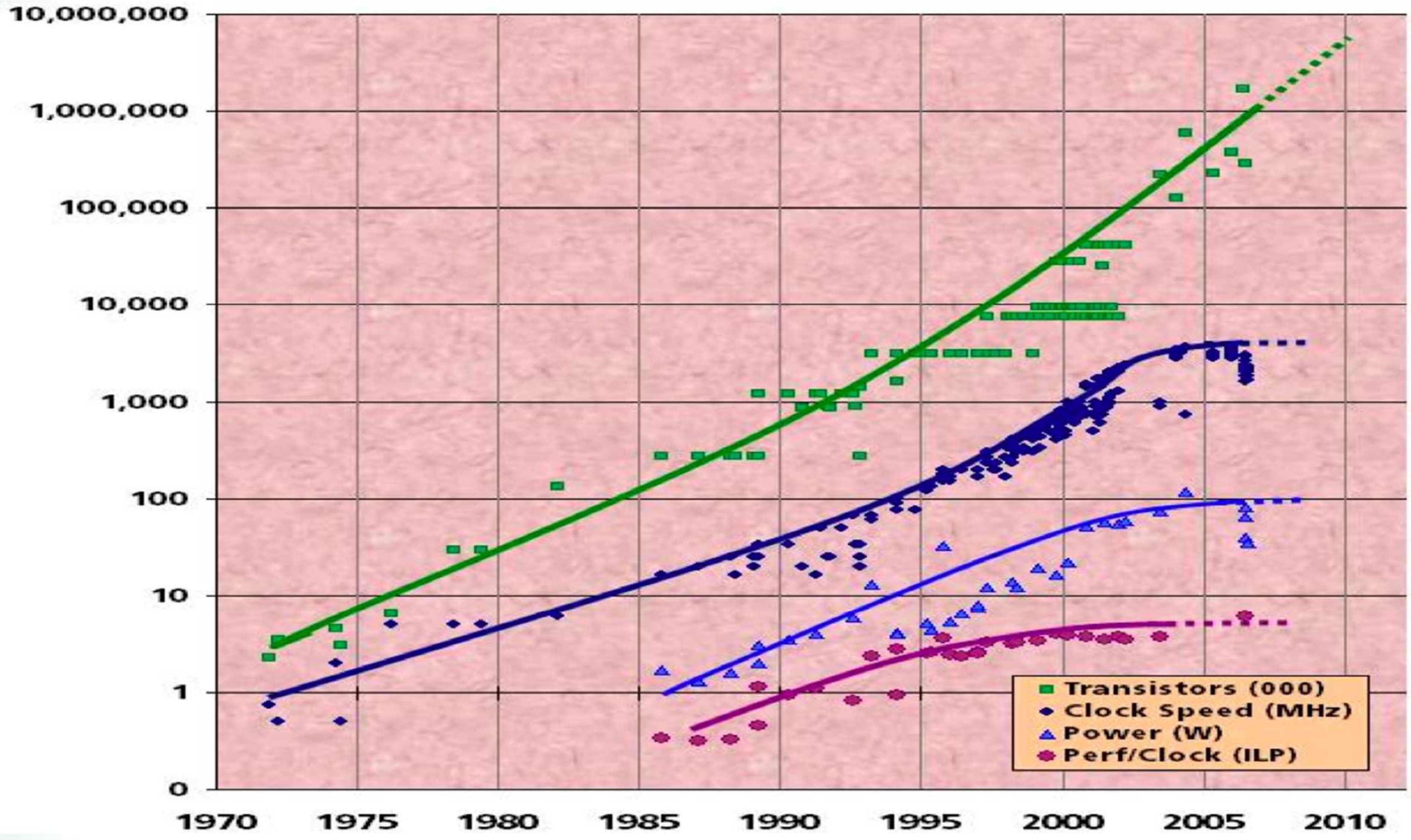
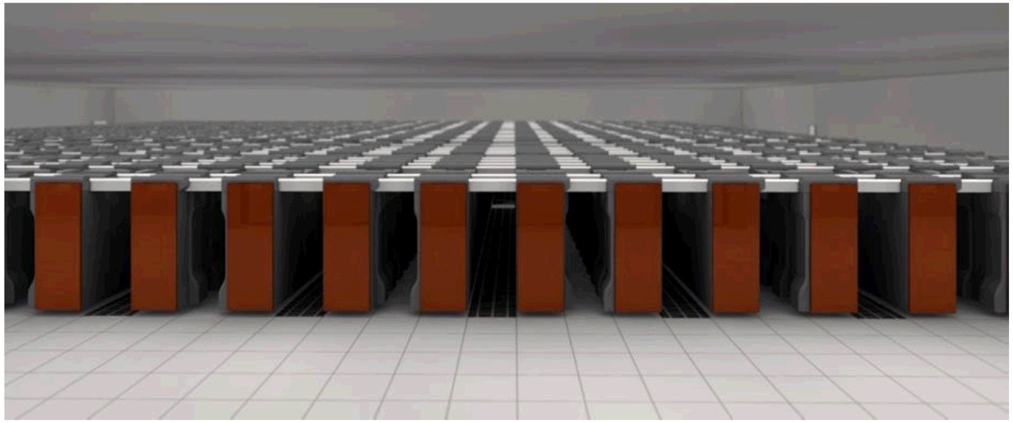


Figure courtesy of Kunle Olukotun, Lance Hammond, Herb Sutter, and Burton Smith

Japanese K-computer

- **10 PFLOPS** Peak Performance in **2012**
  - The Japanese word "**Keisoku**" means **10 petaflops**.
- National Leadership (Initiative by MEXT)
- Next-Generation Supercomputer project
  - Carried out by **RIKEN**
  - Fujitsu **SPARC64** VIIIfx **80,000 CPUs**
- 112 billion yen ($1.3 billion)
- #1 system on TOP500



Interconnect for Petascale Computing

topology / Characteristics	Cross bar	Fat-Tree/ Multi stage	Mesh / Torus
Performance	Best	Good	Average
Operability and Availability	Best	Good	Weak
Cost and Power consumption	Weak	Average	Good
Topology uniformity	Best	Average	Good
Scalability	Hundreds nodes Weak	Thousands nodes Ave.-Good	>10,000 nodes Best
Example	Vector Parallel	x86 Cluster	Scalar Massive parallel

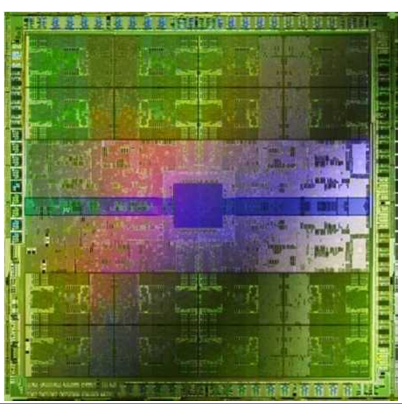
- Which type of the topology can scale up over 100,000 node?

➡ Improvement of the performance, operability and availability of mesh/torus topology are our challenge

GPU / Multi-core Technology

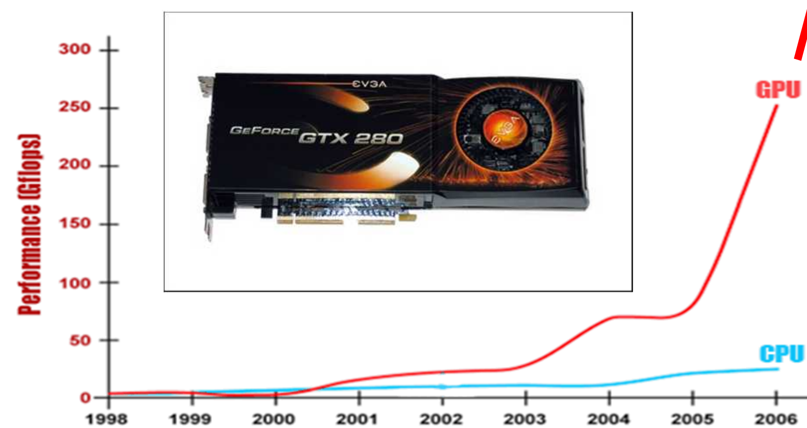
- NVIDIA: Fermi chip first to support HPC
  - Formed partnerships with Cray, IBM on HPC systems
  - #2, #4 systems on TOP500 (Fermi, China)
- AMD/ATI: Primarily graphics currently
  - #7 system on TOP500 (AMD-Radeon, China)
  - Fusion chip in 2011 (5 TeraFlops)
- Intel: Knights Ferry (2011), 32-64 cores

NVIDIA: Fermi (2010)



✧ 1.2 TeraFlops
✧ 8x increase in double precision
✧ 2x increase in memory bandwidth
✧ Error correcting memory

NVIDIA: Tesla (2008)



GPU: 2008
933Gflops
150W

CPU:2008
~45 Gflops
160W

ECMWF

CPU – GPU Comparison

CHIP TYPE	CPU Nahalem	GPU NVIDIA Tesla	GPU NVIDIA Fermi
Cores	4	240	480
Parallelism	Medium Grain	Fine Grain	Fine Grain
<u>Performance</u> Single Precision Double Precision	85 GFlops	933 GFlops 60 GFlops	1040 GFlops 500 GFlops
Power Consumption	90-130W	150W	220W
Transistors	730 million	1.4 bilion	3.0 billion

Next Generation Weather Models

- Models being designed for global cloud resolving scales (3-4km)
- Requires PetaFlop Computers

DOE Jaguar System

- 2.3 PetaFlops
- 250,000 CPUs
- 284 cabinets
- 7-10 MW power
- ~ $100 million
- **Reliability in hours**

Equivalent GPU System

- 2.3 PetaFlop
- 2000 Fermi GPUs
- 20 cabinets
- 1.0 MW power
- ~ $10 million
- **Reliability in weeks**

- Large CPU systems (>100 thousand cores) are unrealistic for operational weather forecasting
  - Power, cooling, reliability, cost
  - Application scaling

Valmont Power Plant ~200 MegaWatts Boulder, CO

ECMWF

Fortran GPU Compilers

- General Features
  - Do not support all Fortran language constructs
  - Converts Fortran into CUDA for further compilation
- CAPS – HMPP
  - Extensive set of parallelization directives to guide compiler analysis and optimization
  - Optionally generates OpenCL
- PGI
  - **ACCELERATOR** – directive-based accelerator
  - **CUDA Fortran** – Fortran + language extensions to support Kernel calls, GPU memory, etc
- F2C-ACC
  - Developed at NOAA for our models
  - Requires hand tuning for optimal performance

Different Scaling Trends for Different Technologies Compute Ratios will Change

- Driven by Cost and Energy Usage
- Circuit-Flop Densities will Continue to Improve
- I/O BWs and Power will not improve as quickly
  - Technology Improvements may help this
  - Costs may still be limiters
- Memory Volume Costs (and BWs) may be Limiting

The Big Leap from Petaflops to Exaflops

- Technology disruptions in many areas driven by Power and Cost Concerns.
- All Impact System Balance and Application Optimization
  - Silicon power scaling:
    - Frequency Plateau – more threads needed
    - Impacts Application Scaling, Power Usage, and RAS
  - Memory technology – Volume and BW
    - Bytes/Flop ratios decreasing (Locality Counts)
  - Interconnect BW
    - Bytes/Flop ratios decreasing (Locality Counts)
  - Packaging technology – I/O Switching Costs
    - Relative Amount of Power needed to move Data Increasing
- Need to be able to exploit machines. Not just about flops.
  Flop metric promises to be an even poorer predictor of sustained performance than it is now

Scaling Limitations

Not all applications will scale to exascale with their current structure due to things like:

- Parallelism
  - O (10**11) Threads required
  - Load Imbalance and Serial Portions
- Locality
  - Vertical (temporal)
    - Data Reuse is not always possible
    - Movement in the Memory Hierarchy Occurs
  - Horizontal (data decomposition)
    - Excessive Movement uses Energy
    - Introduces Latency and Serialization Issues
- Bottlenecks in Memory, Communications, and I/O BWs

Conclusion

- Fundamental Programming Style not likely to change much
  - Multi-Threaded ‘MPI tasks’ will be the norm
  - New Languages are emerging to help with extreme scale
- A Shared Memory model at the Task level will still exist
- Amount of Threading will have to increase
- ‘More Science’ will be a way to use cycles
- Optimization Points will change – Computing is ‘free’
- New Tools are emerging to help create applications ‘at scale’

Scaling issues: today

- Static Load Imbalance
  - per MPI task, per OpenMP thread
- Dynamic Load Imbalance
  - e.g. physics computations, semi-Lagrangian comms
- Jitter
- MPI Comms Latency, Topology
- OpenMP overheads, NUMA
- Input/Output
- Shell scripts

Scaling to 100K - 1M threads ?

- Next 5 to 10 years ?
- Can this still be done with MPI + OpenMP ?
- Partitioned global address space (PGAS) languages?
- Fortran 2008 coarrays
- Jitter free systems
- Need comms to speed up with the increase in cores
- Overlap compute and comms?
- Tools (debuggers, profilers)
  - That work reliably and fast at high core counts
  - That work with large applications

Can we modify DA methods to use future computer architectures better?

Data Assimilation improvements

There are many areas where data assimilation can be improved without adding considerable computational resources (but they may require large human resources):

Improve data assimilation methods:

- Improve representation of model error
- Improve representation of analysis uncertainty
- Improve handling of biases

Extract more information from observations

Use more observations

Improve the forecast model

Enhance diagnostics of the assimilation system

Is ECMWF’s DA plan computationally viable?

Hybrid DA system: Use EDA information in 4D-Var

- Flow dependent background error variances and covariances in 4D-Var

Long-window weak-constraint 4D-Var

Unified Ensemble of Data Assimilations (EDA) and Ensemble Prediction System

- For estimation of analysis and short range forecast uncertainty that will benefit the deterministic 4D-Var
- For estimation of long range forecast uncertainty (the present role of the EPS)

Note: The EDA is a ‘stochastic EnKF’ with an expensive 4D-Var component. It may be replaced or supplemented by an LETKF system, if beneficial.

Is ECMWF's DA plan computationally viable?

A 50 member Ensemble Prediction System (EPS) is more scalable than the high resolution forecast model

A 10 (an of course also a 50) member EDA is more scalable than the high resolution deterministic 4D-Var

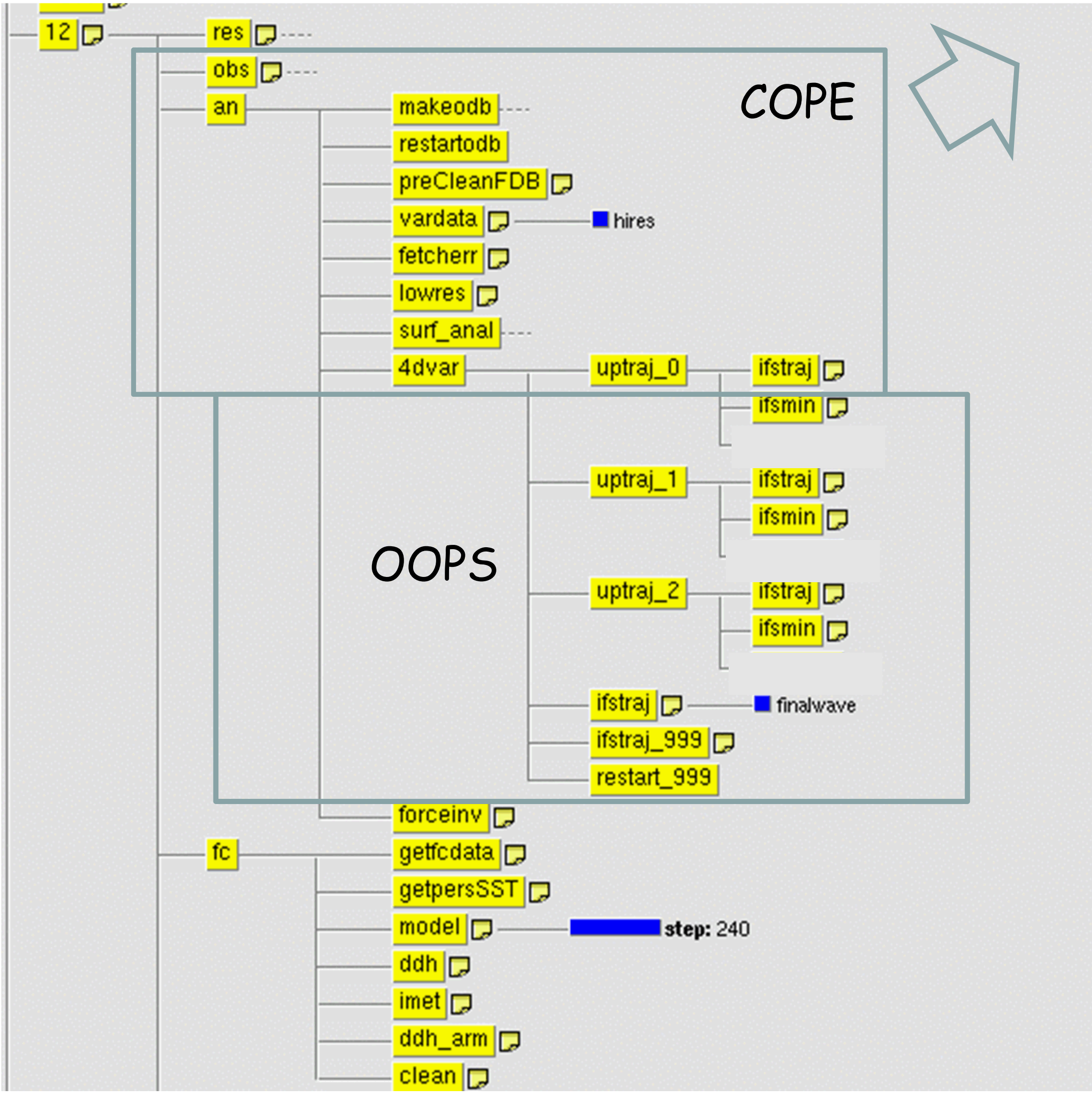
EnKF is more scalable than the EDA, but possibly requires more CPU cycles to achieve the same

EnKF (and EDA) are less scalable than the EPS system, because of observation handling, I/O and sequential parts of the analysis step

En-4D-Var is likely to be as scalable as EnKF/EDA, but I/O and memory BW is an issue

The main question to answer: Is deterministic 4D-Var scalable?

Improving scalability of time critical DA suite



4D-var time window is 12 hours

Forecast & Outer loop trajectories:

(Traj_0,1,2) are using T1279 resolution
Grid columns = 2×10^6

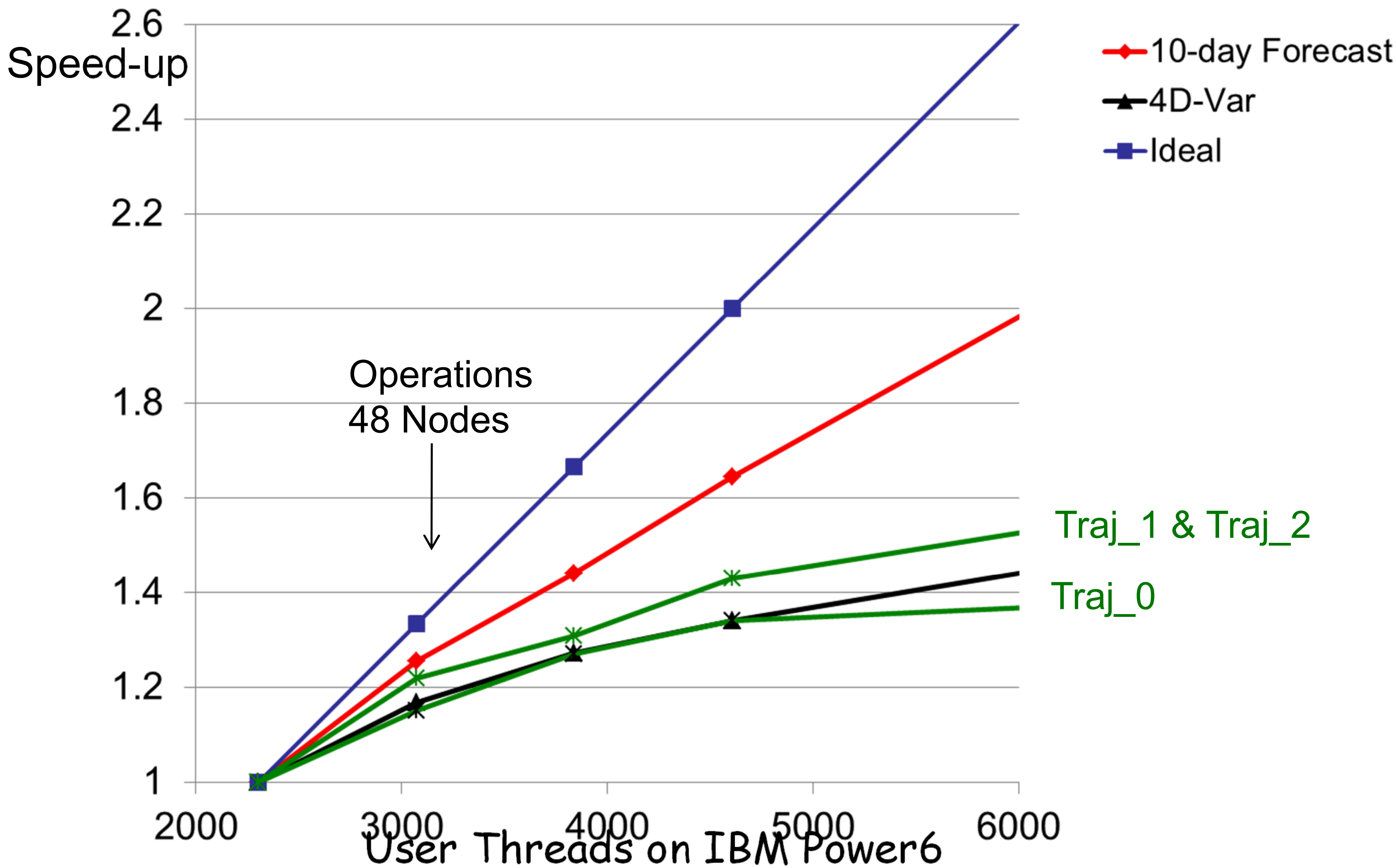
Three minimizations:

Min_0 : T159
Grid columns = 36000

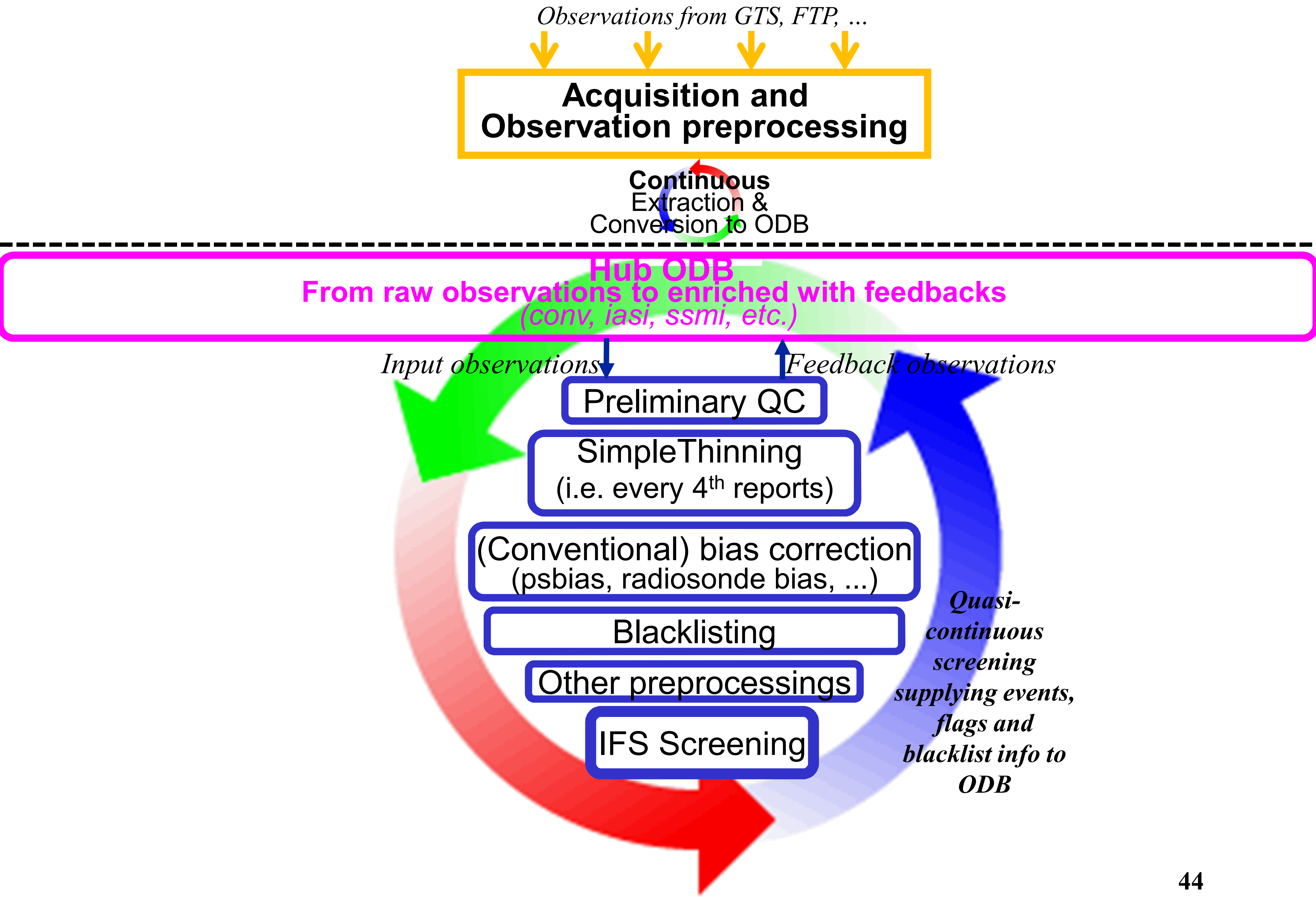
Min _1 & 2 : T255
Grid columns = 89000

Vertical = 91 levels

Scalability of T1279 Forecast and 4D-Var



Continuous Observation Processing Environment (COPE)



Continuous Observation Processing Environment (COPE)

- Hub ODB serving as an interface to all our observation processing (screening, monitoring, analysis, diagnostics) and “continuously” fed by arriving observations
- Shortens the time critical path by performing observation pre-processing and screening quasi-continuously as data arrive
- Reduce risk of potential failures in the operational analysis during the time critical path and allow for early response when observation problems occur
- Enables near real-time quality control and monitoring of observations
- More modular and simplified quality control, bias correction and screening

The 5 Dimensions of 4D-Var

- The bulk of the 4D-Var algorithm comprises 5 nested loop directions:

  1. Minimisation algorithm iterations (inner and outer),
  2. Time stepping of the model (and TL/AD),
  3. Latitude, `NPROMA`
  4. Longitude, `NPROMA`
  5. Vertical.

- Only two are parallel!

- We need to look at the other directions for more parallelism, for example:

  - Minimisation algorithm:
    - Parallel search directions,
    - Parallel preconditioner and less iterations,
    - Observation space algorithms, saddle point algorithms.
  - Time stepping:
    - Weak constraint 4D-Var.

- Scalability cannot be improved solely by technical or local optimizations!

Object-Oriented Prediction System – The OOPS project

- Data Assimilation algorithms manipulate a limited number of entities (objects):
  - x (State), y (Observation),
  - H (Observation operator), M (Model), H*& M*(Adjoints),
  - B & R (Covariance matrices), etc.

  - To enable development of new data assimilation algorithms in IFS, these objects should be easily available & re-usable

- More Scalable Data Assimilation

- Cleaner, more Modular IFS

OOPS → More Scalable Data Assimilation

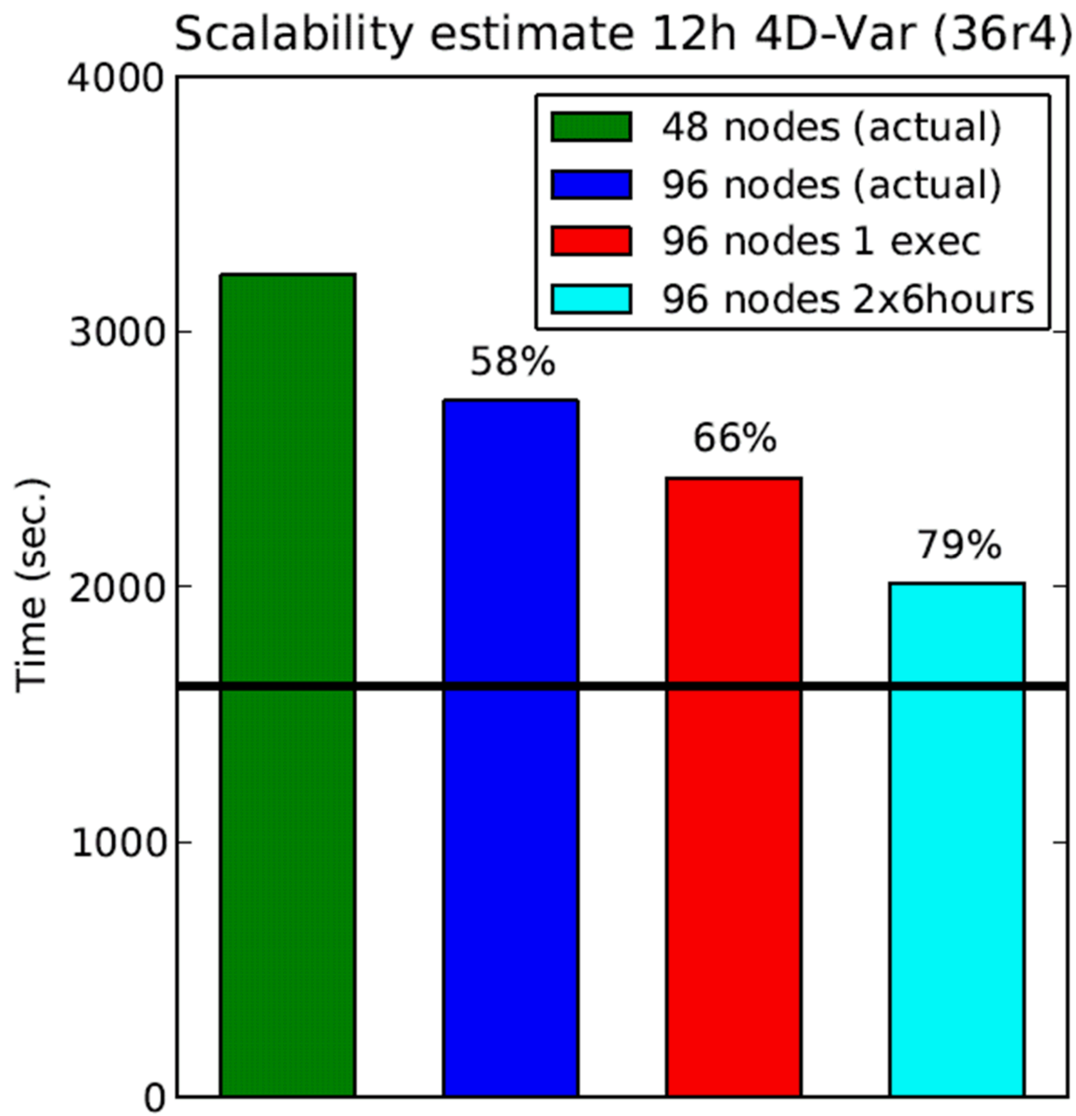
- One execution instead of many will reduce start-up - also I/O between steps will not be necessary

- New more parallel minimisation schemes
  - Saddle-point formulation

  (Only OOPS has made it possible to for Mike to implement the saddle-point formulation so quickly!!)

- For long-window, weak-constraint 4D-Var: Minimization steps for different sub-windows can run in parallel as part of same execution

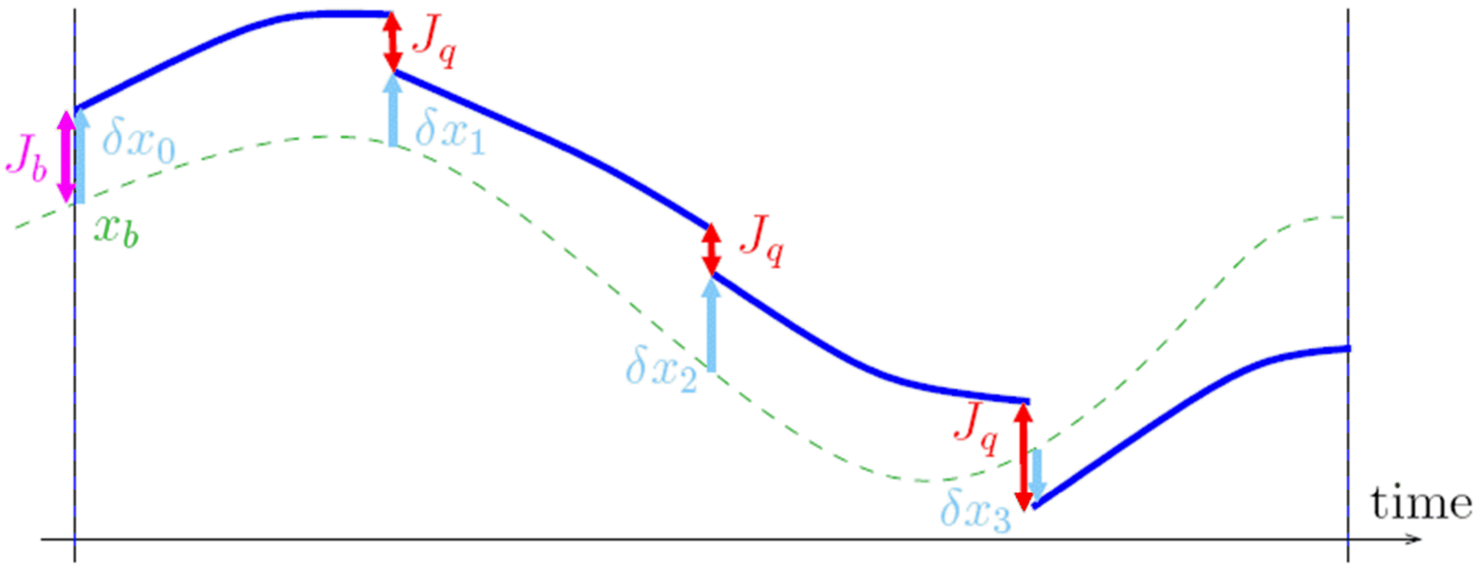
An Example of Better Scalability in 4D-Var



- One executable instead of 7 reduces I/O and start-up costs.
- Weak constraints 4D-Var with a split window gives access to more parallelism.

Figure from Deborah Salmond

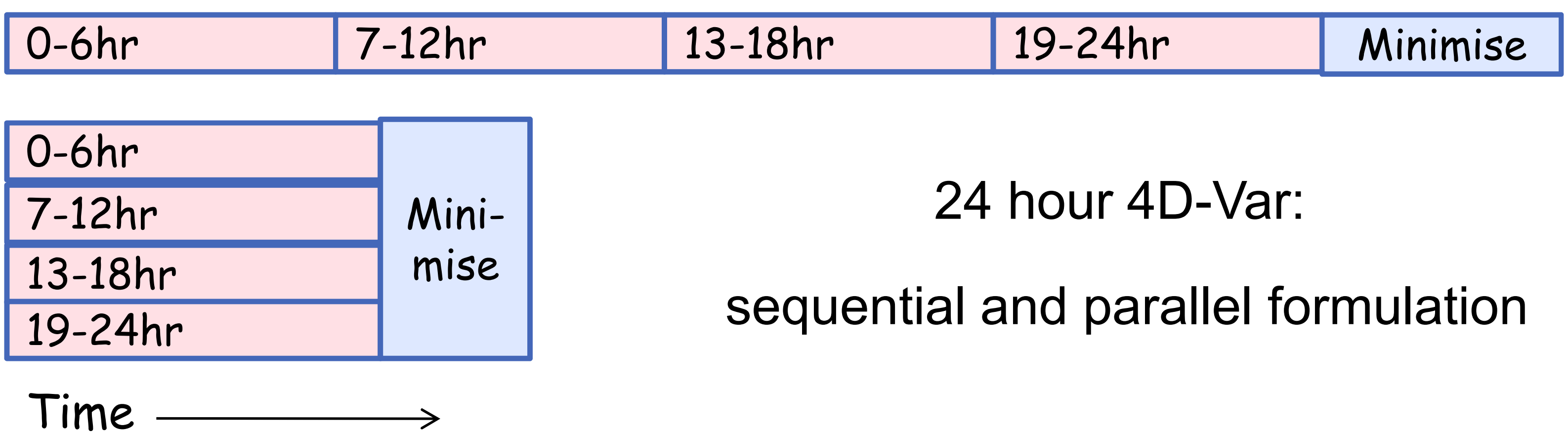
Weak Constraint 4D-Var



- Model integrations within each time-step (or sub-window) are independent:
  - Information is not propagated across sub-windows by TL/AD models,
  - $\mathcal{M}$ and $\mathcal{H}$ can be run in parallel over the sub-windows.
- Several shorter 4D-Var cycles are coupled and optimised together.
- 4D-Var becomes an elliptic problem and preconditioning becomes more complex.

OOPS → More Scalable Data Assimilation

- For long-window, weak-constraint 4D-Var: Minimization steps for different sub-windows can run in parallel as part of same execution as shown by Mike yesterday
- This ‘parallel formulation’ will make 4D-Var very scalable
- In the limit 4D-Var will become more scalable than the forecast model, because the sequential time integration no longer is required
- The ‘sequential formulation’ will not be scalable, but is still expected to be acceptable for the operational ECMWF configuration until 2018.



24 hour 4D-Var:

sequential and parallel formulation

Conclusions

- Significant efforts are required to ensure scalability of data assimilation systems in the future
- Removing as much as possible from the time critical part is essential and will become more important in the future (COPE)
- With optimizations, reduction in I/O, and reducing the number of start ups via one executable it is possible to extend the scalability of ECMWF data assimilation system by some years. Some of this will be done as part of OOPS.
- But 4D-Var is not scalable or viable as the operational system (beyond 2018?) at ECMWF if a ‘sequential formulation’ is retained.
- A ‘parallel formulation’ of 4D-Var, as being developed in OOPS, will make 4D-Var scalable and viable for the next two decades
- EDA and EnKF will be scalable, but IO and memry BW an issue