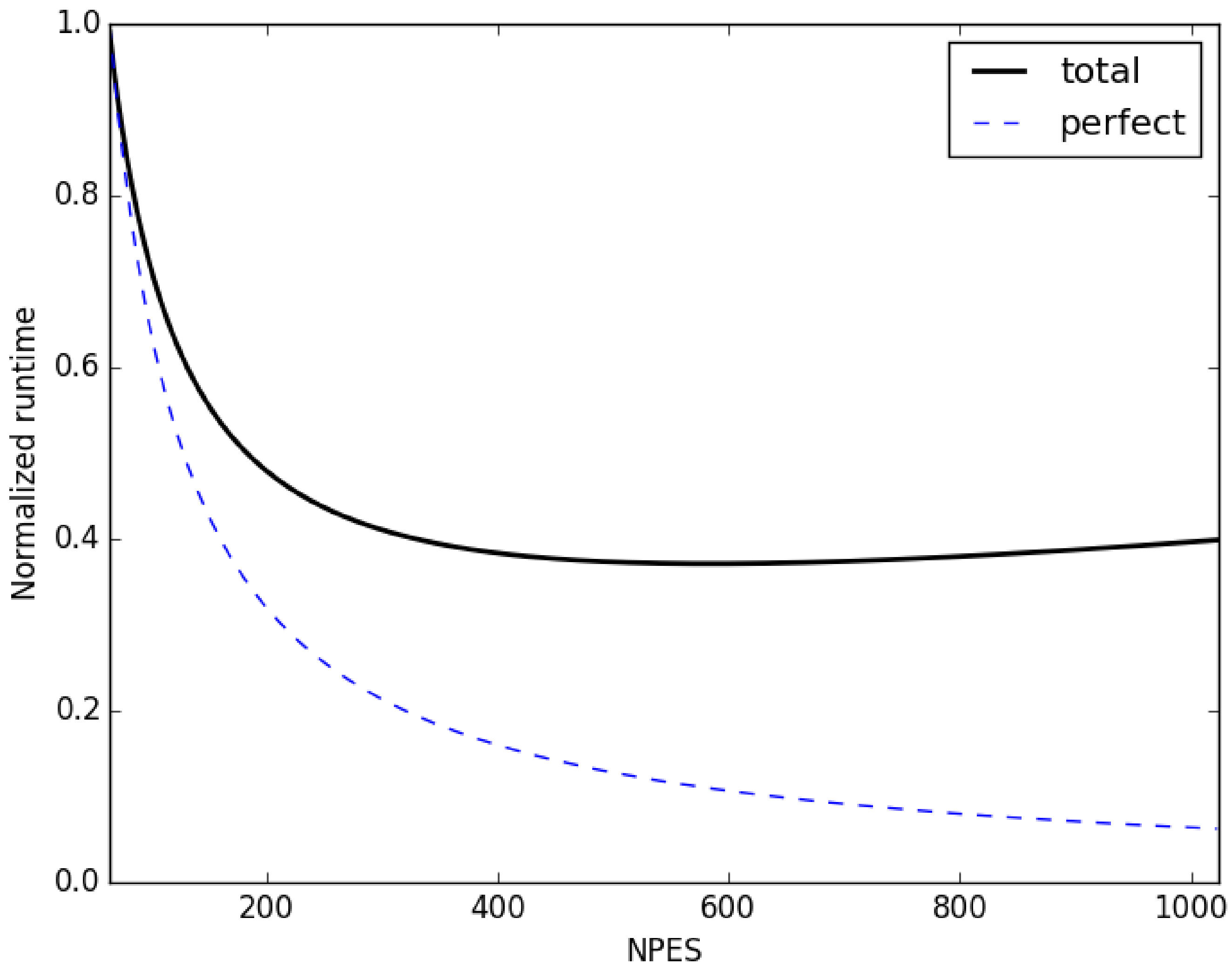
# Optimizing IFS observation processing: - a case study in scalability enhancement

Peter Lean

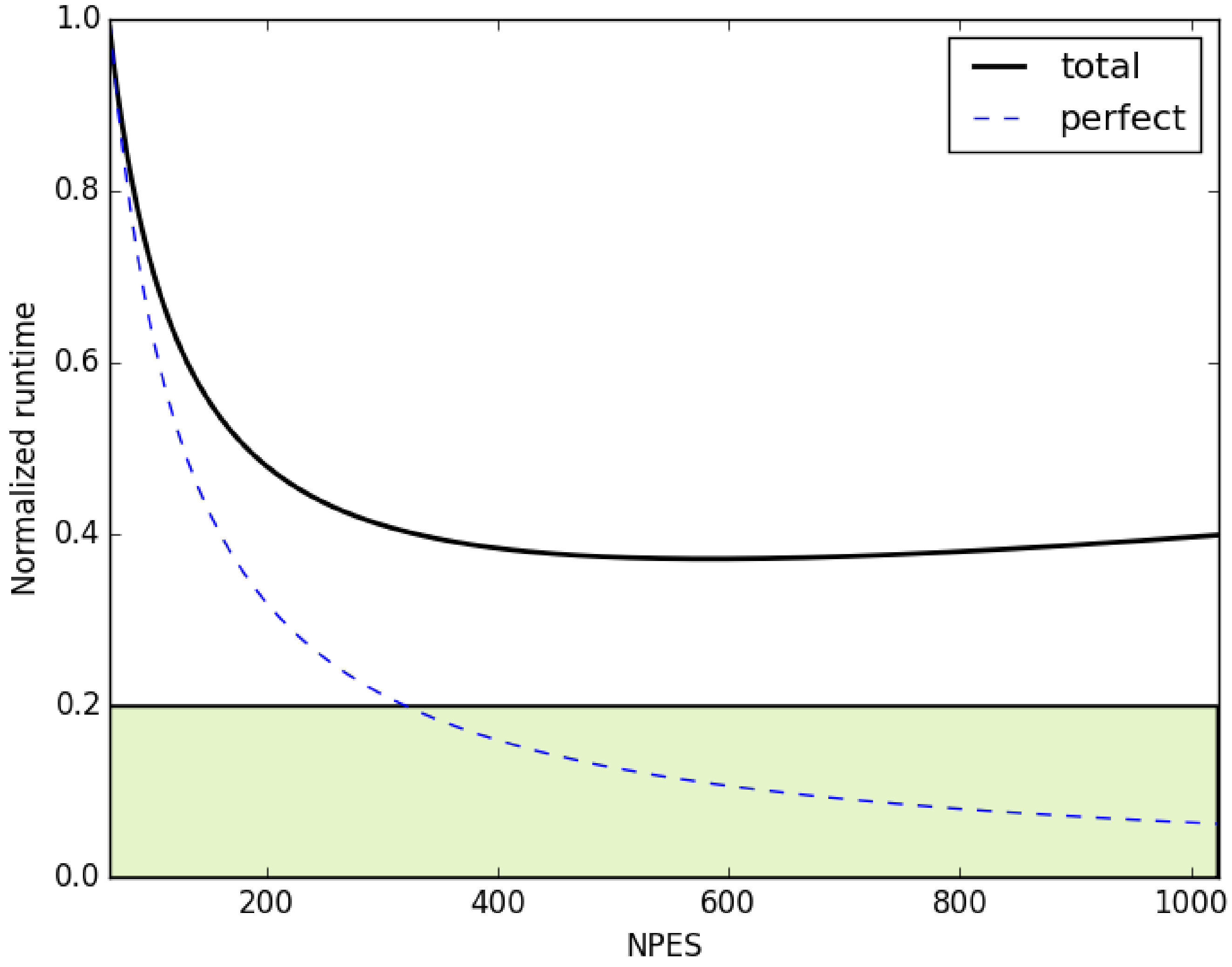
Thanks - Alan Geer, Deborah Salmond, John Hague and Niels Bormann

17th Workshop on High Performance Computing in Meteorology    October 2016

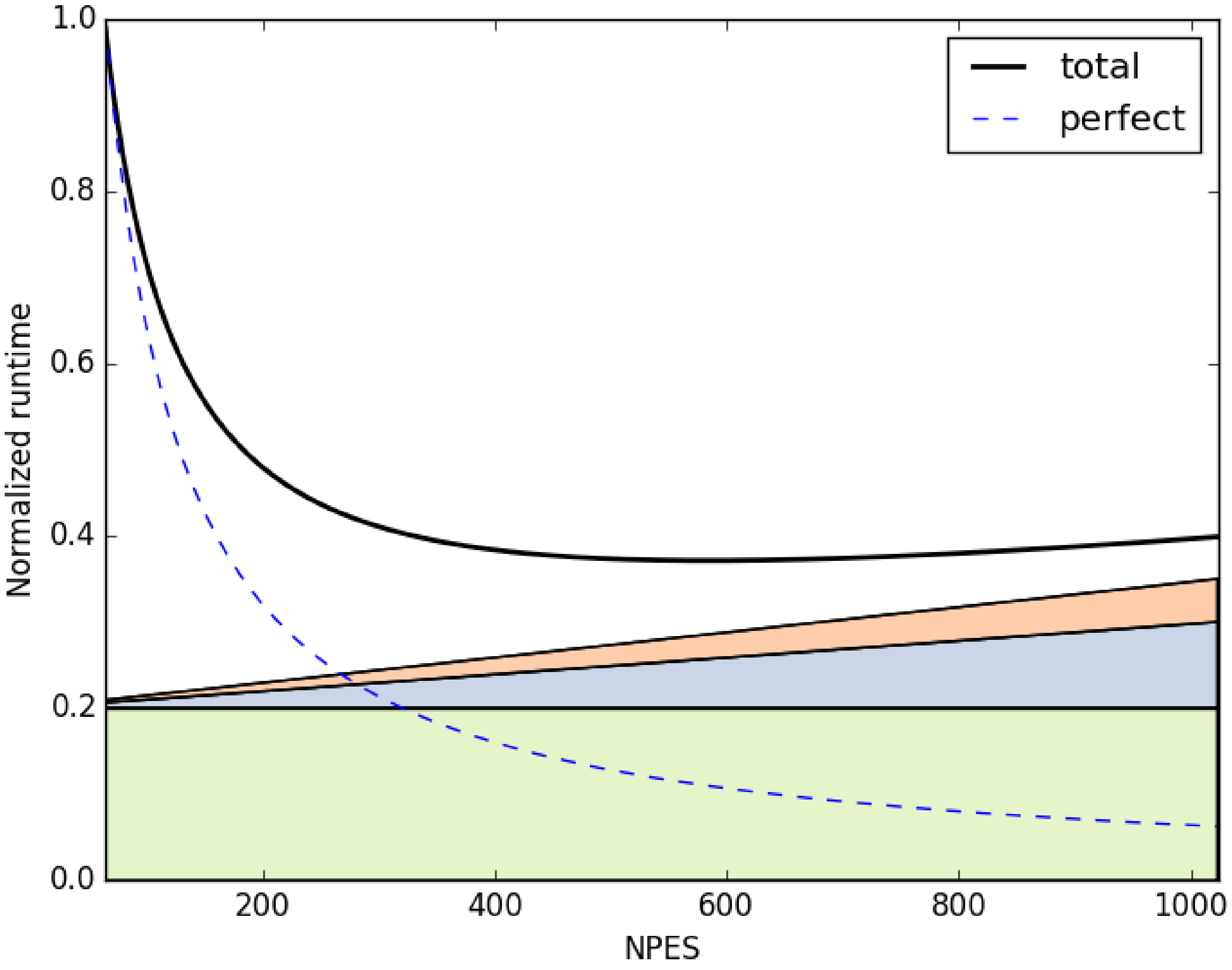
# Strong scalability:



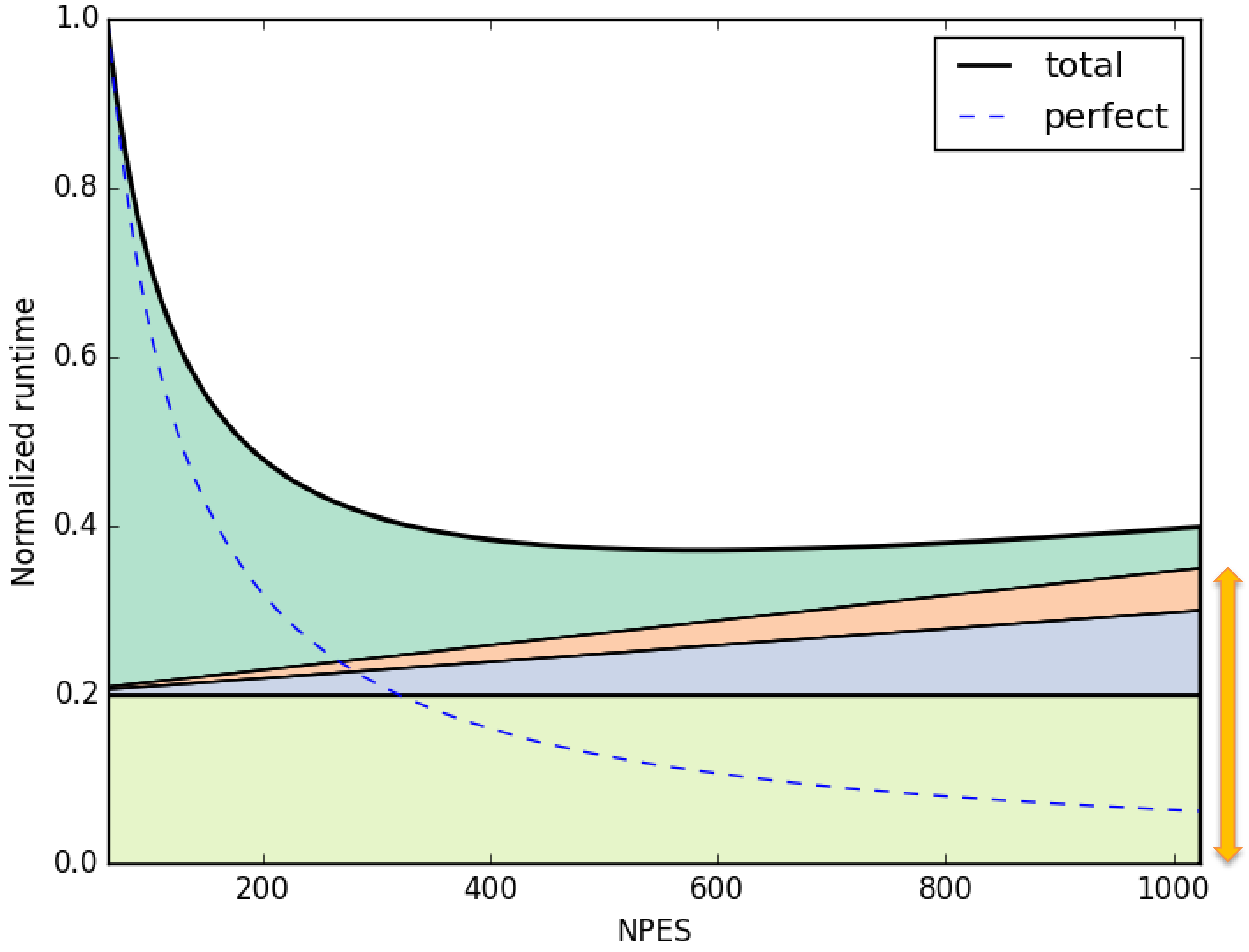
# Strong scalability:



# Strong scalability:



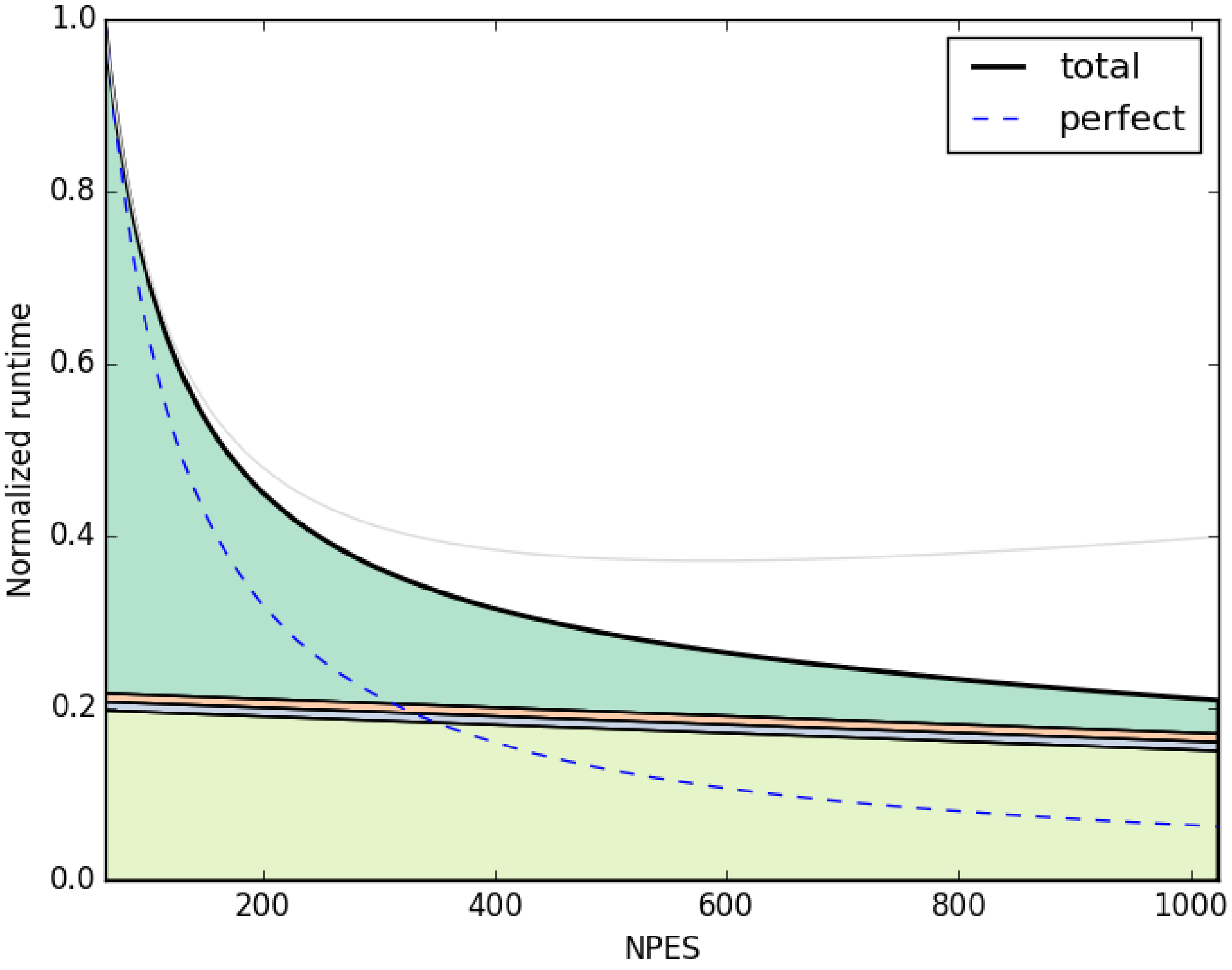
# Strong scalability:




Total runtime becomes dominated by those aspects that don't scale!

= Amdahl's law

# Strong scalability:




Significant improvement

# To improve the scalability of an application:

1. Identify the top few routines which don't scale well (or blow up)

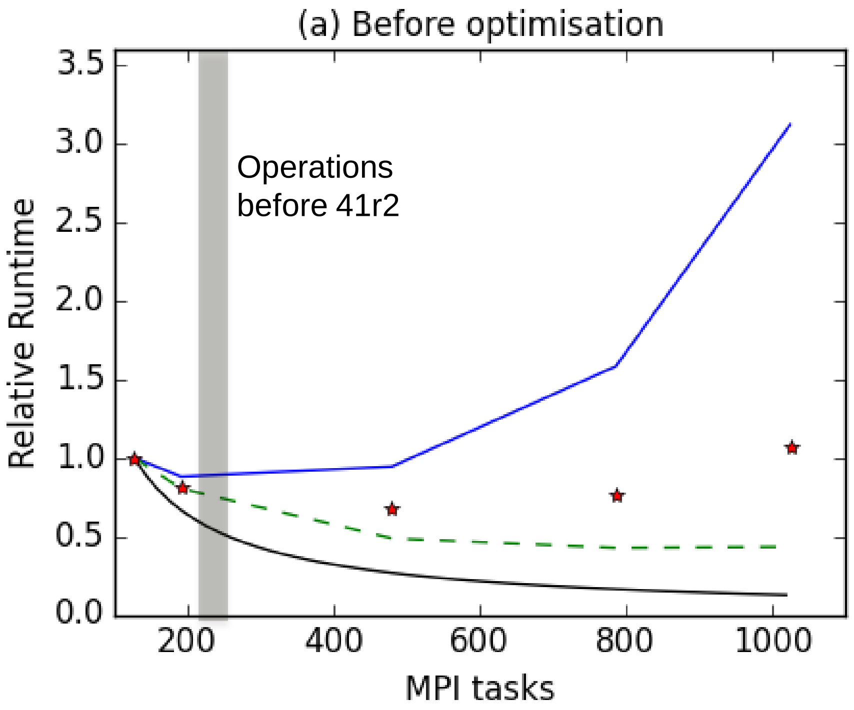
2. Fix them.

*Easy!*

# The hard part:

## Fixing scalability of the components that blow up – a case study

# Problem with scalability of observation processing on Cray XC30



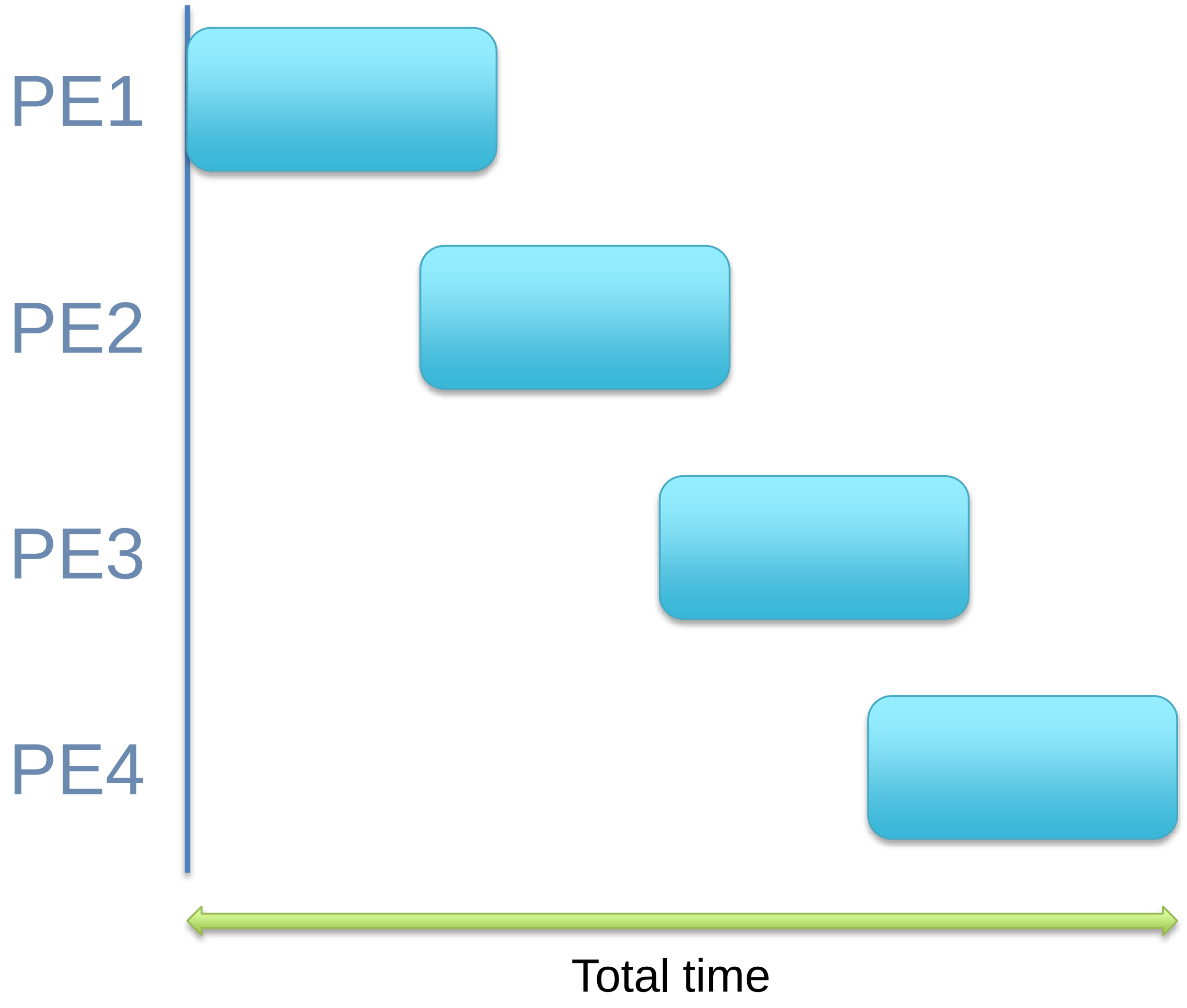

Screening trajectory — 300 Million observations

Overall 4D-Var

Minimisations — 20 Million observations

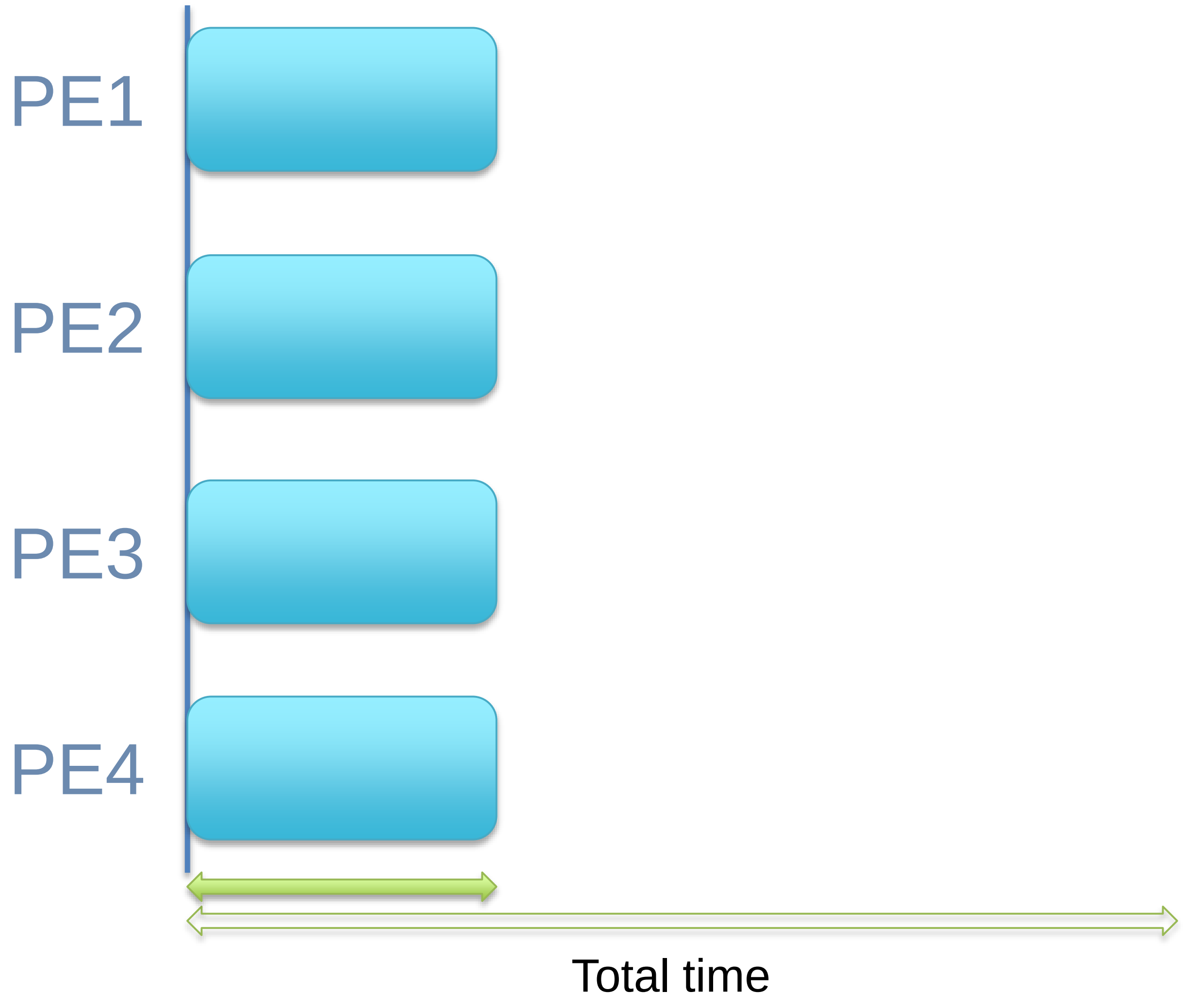
# 1. Parallel I/O in Observation Database

# Parallel I/O for Observation Database



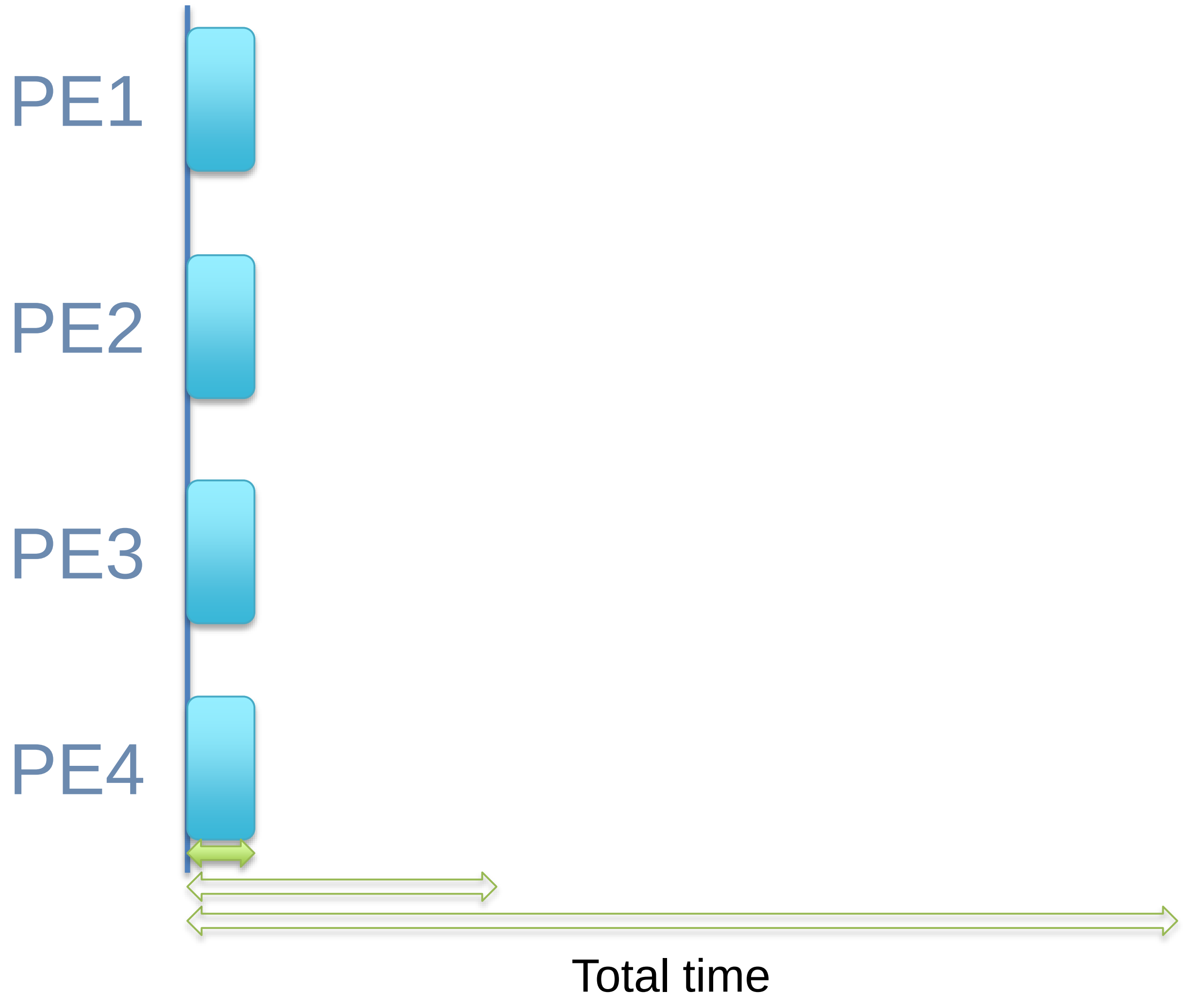

- Only subset of tasks do I/O in ODB to reduce strain on file system.
- Message passing sends the data to the PE doing I/O
- Ordering of loops in message passing and I/O code was inefficient
- Spent lots of time waiting at MPI barriers for messages to be received

# Parallel I/O for Observation Database




- Refactored the message passing and I/O loops to minimize time spent at barriers.
- Noticed that majority of time still spent on non I/O aspects.

# Parallel I/O for Observation Database




**Changed lookup table:**

- was O(n) number of tasks
- now O(1)

**Two issues:**

1. Time waiting at MPI barriers
2. Slow lookups

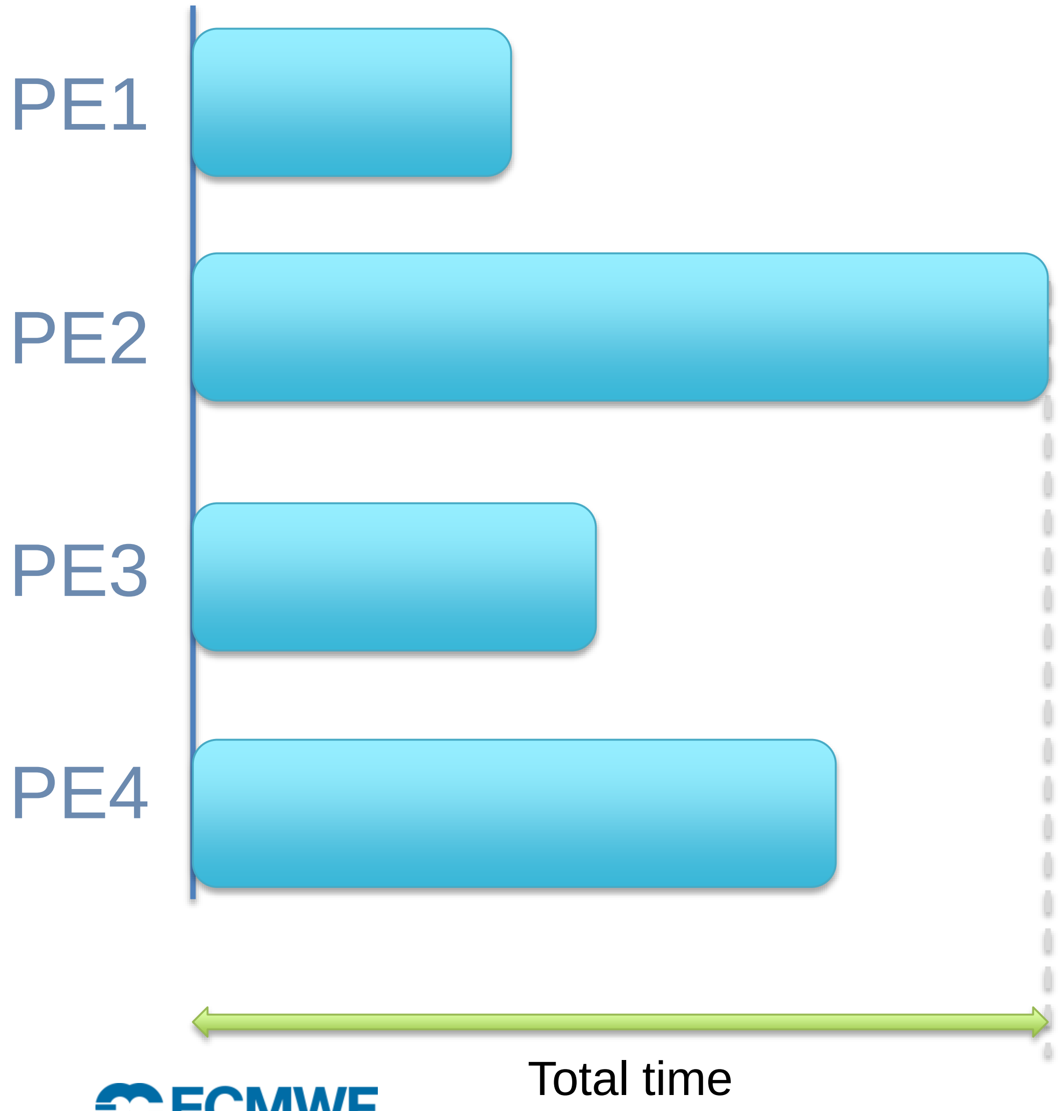
**Amplification:**

- barrier issue amplified the slow lookup issue
- lookups effectively became sequential.

e.g. 0.1s per lookup x 1000 MPI tasks = 100s!

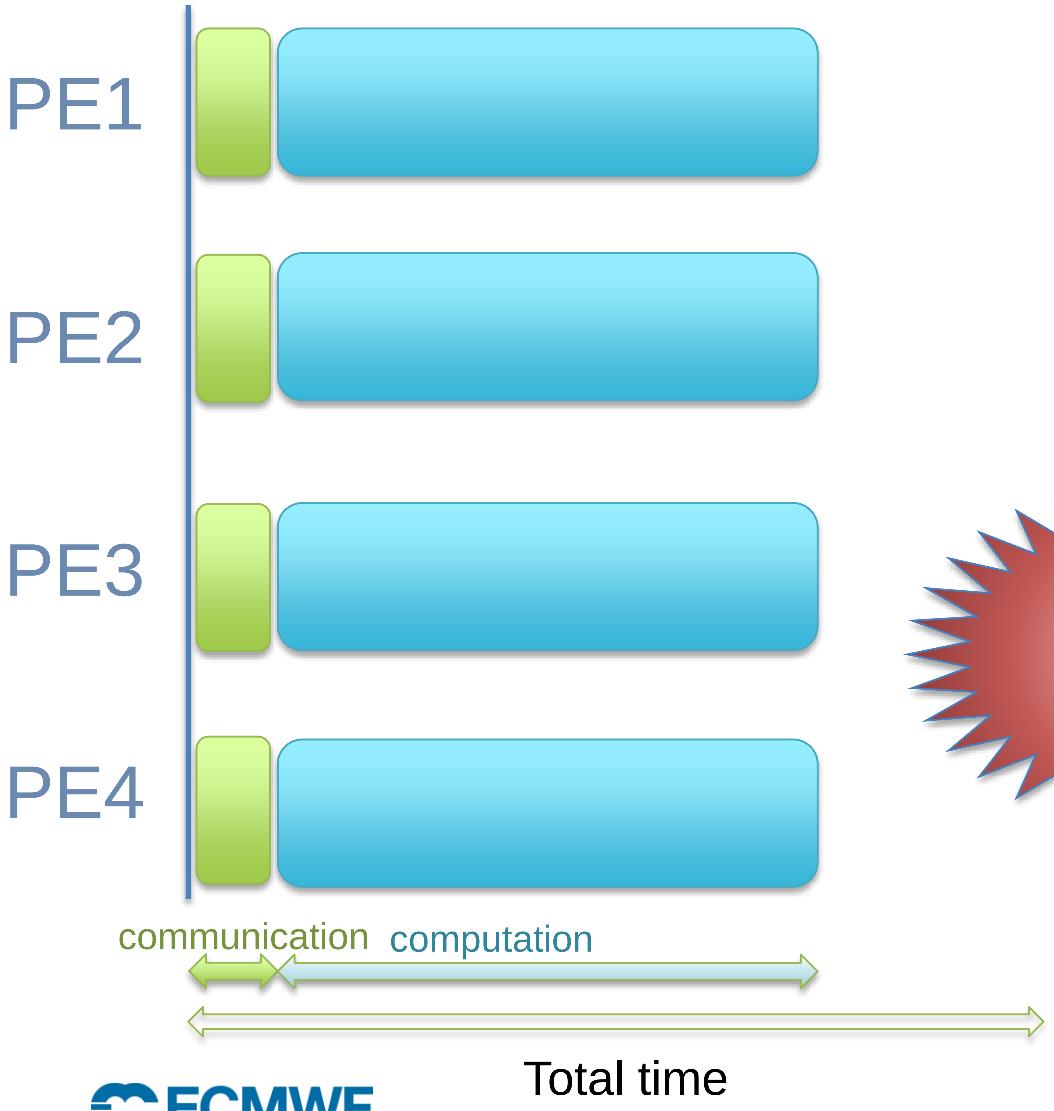
# 2. Load balancing of active observations in 4D-Var

# Load balancing of active observations in 4D-Var




- Observations are distributed across MPI tasks in ODB / IFS.
- Quality control / screening removes 90% of the original observations.
- After Quality Control, the observations may no longer be well balanced.

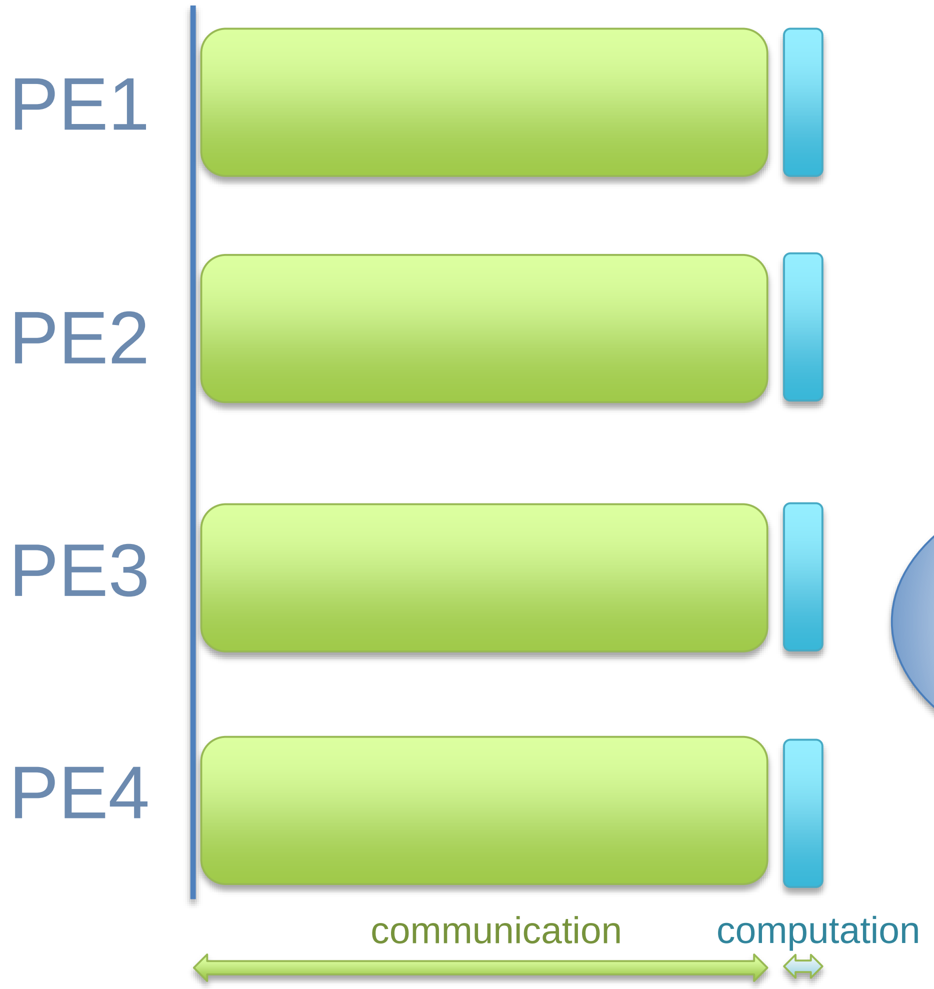
# Load balancing of active observations in 4D-Var





- Observations are distributed across MPI tasks in ODB / IFS.
- Quality control / screening removes 90% of the original observations.
- Re-distribute the active observations in 4D-Var evenly across MPI tasks:
  - Communication required
  - Computational time on each task processing the observations is more even.
  - **Overall time runtime is reduced.**

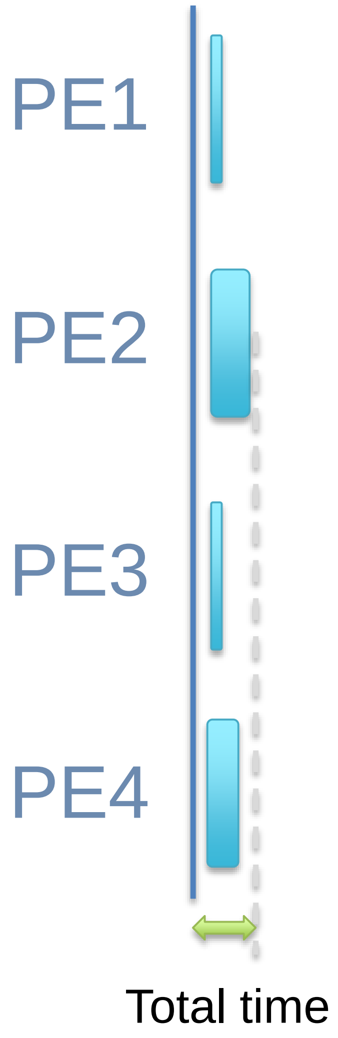
# Load balancing of active observations in 4D-Var




2016

- Observations are distributed across MPI tasks in IFS.
- Screening / quality control removes 90% of the original observations.
- Re-distribute the active observations in 4D-Var evenly across MPI tasks:
  - Communication required
  - Computational time on each task processing the observations is more even.
  - ~~Overall time runtime is reduced.~~

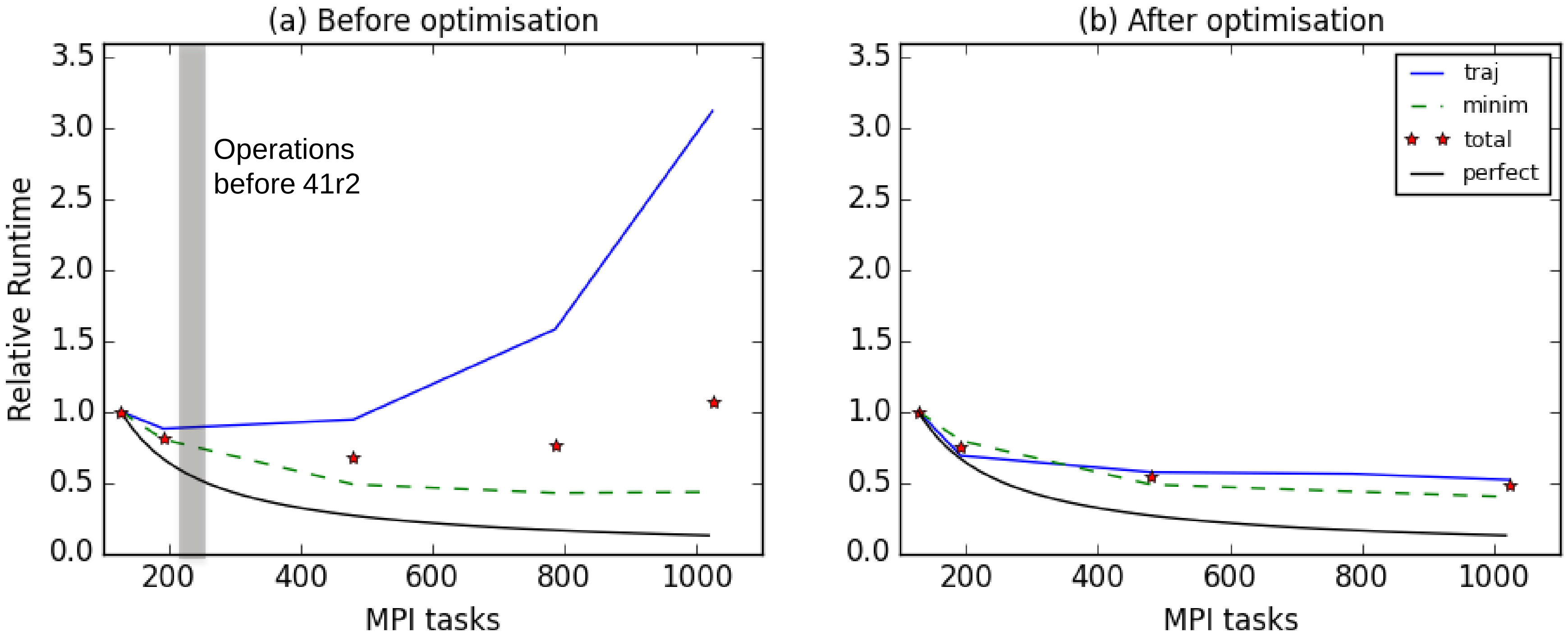
# Load balancing of active observations in 4D-Var




- In 2016, the relative cost of computation has gone down (scaled away as running on far more PEs).
- **The load balancing costs more than it saves.**

**Optimisations which gave improvements 10 years ago may no longer be giving the same benefits today!**

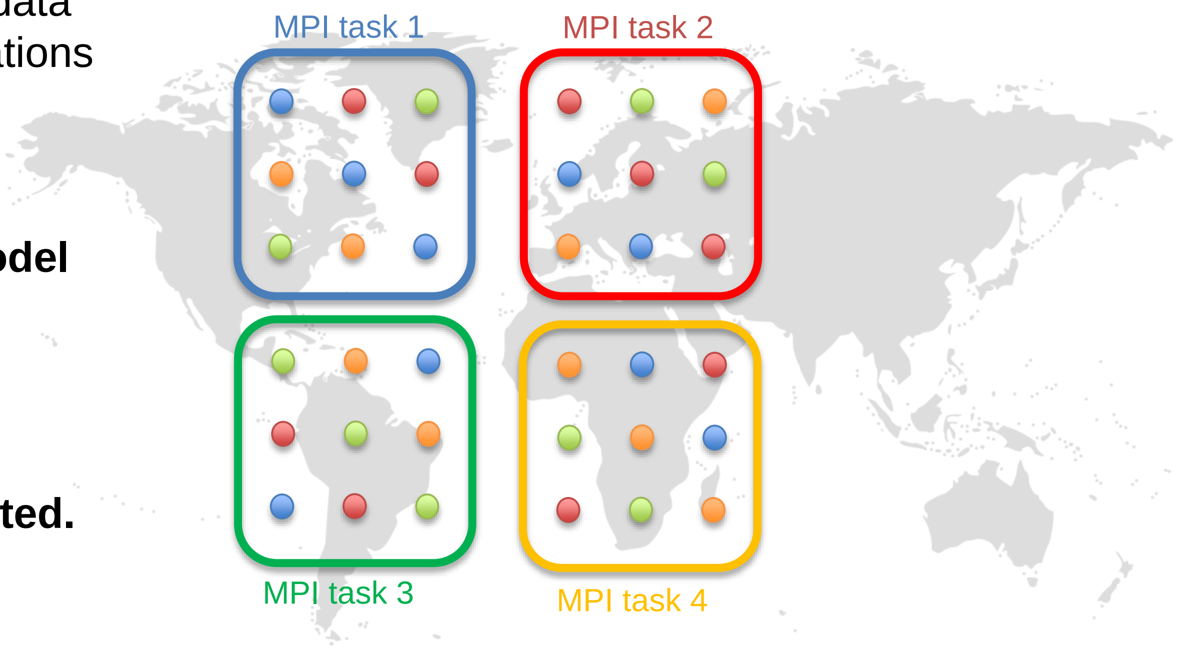
# Improved scalability:




- Scalability of first and final trajectories significantly improved
- Running on 1000 MPI tasks, 4D-Var sped up x2
- **But scalability of 4D-Var still not perfect…**

# Future directions:
## data locality → reduced communications

- Data locality: reduce costly data transfer by doing the calculations where the data resides.
- 4D-Var compares each **observation** against the **model equivalent**
  - **Two "Big" datasets**
  - **Generally not co-located.**

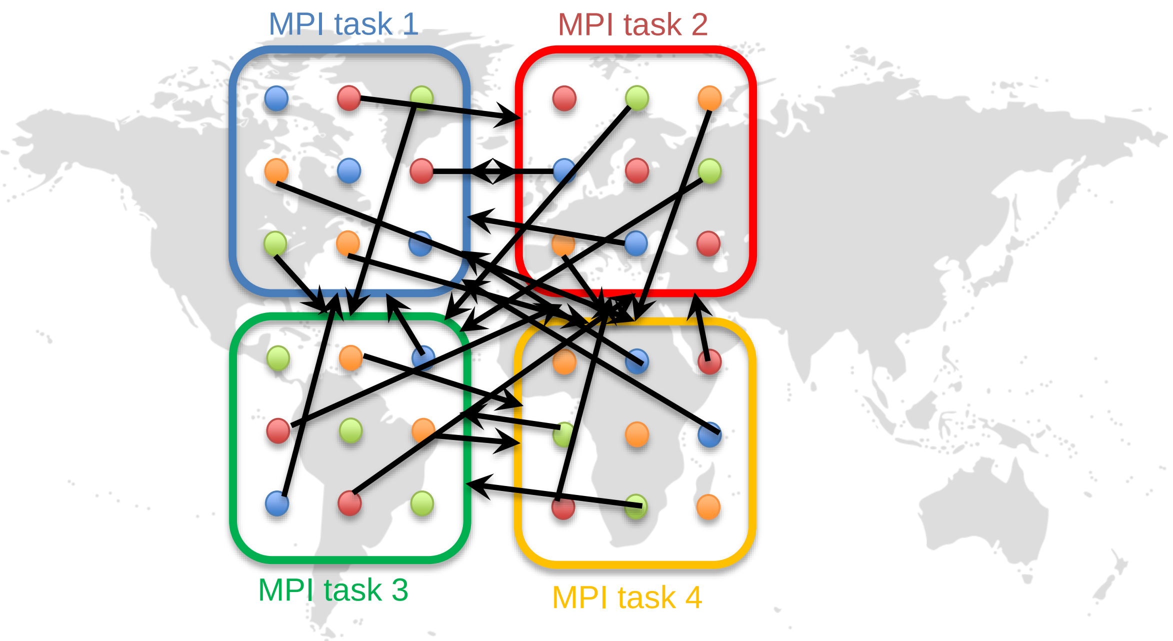



Model domain decomposed geographically across MPI tasks

ODB distributes observations randomly across MPI tasks

# Future directions:
## data locality → reduced communications

Comparison between observation and model usually requires communication



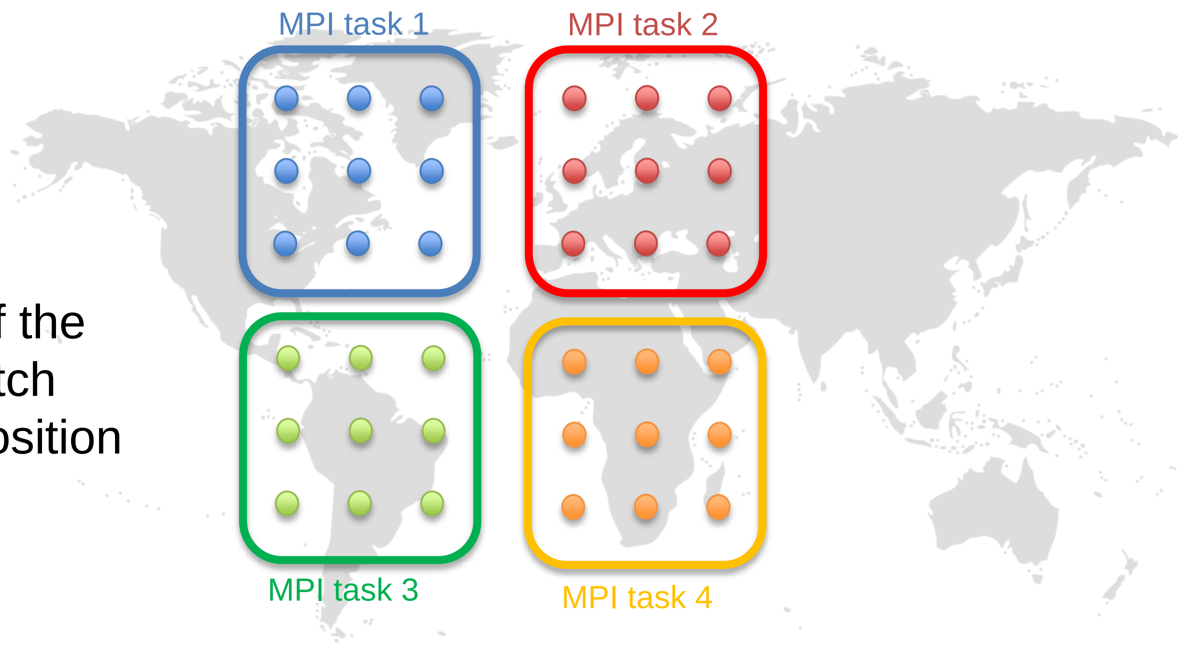

| | Runtime [s] |
|---|---|
| ODB I/O | 3s |
| Interpolation + **communication** of model profiles at observation locations | 34s |
| Observation operator (calculation of observation equivalents) | 3s |

ECMWF

# Future directions:
## data locality → reduced communications

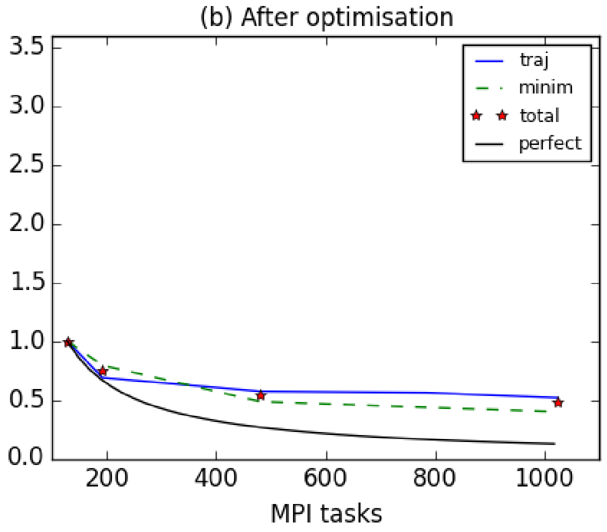
Geographical partitioning of the observations in ODB to match the model domain decomposition should reduce required communication

⇨ Faster, more scalable 4D-Var (hopefully)



# Improving scalability: short v long term perspectives

- **Short/medium term: 1-2 years**
  - Optimising existing code : **spend** ⇨ short term payoff
  - Can give us several years headroom before scalability becomes a problem and can deliver significant performance improvements today.
  - But it won't solve the major scalability challenges we face.




- **Long term: 2 years +**
  - **Invest** in major re-writes / restructuring:
    ⇨ major scalability improvements (but maybe not for another 5 years)

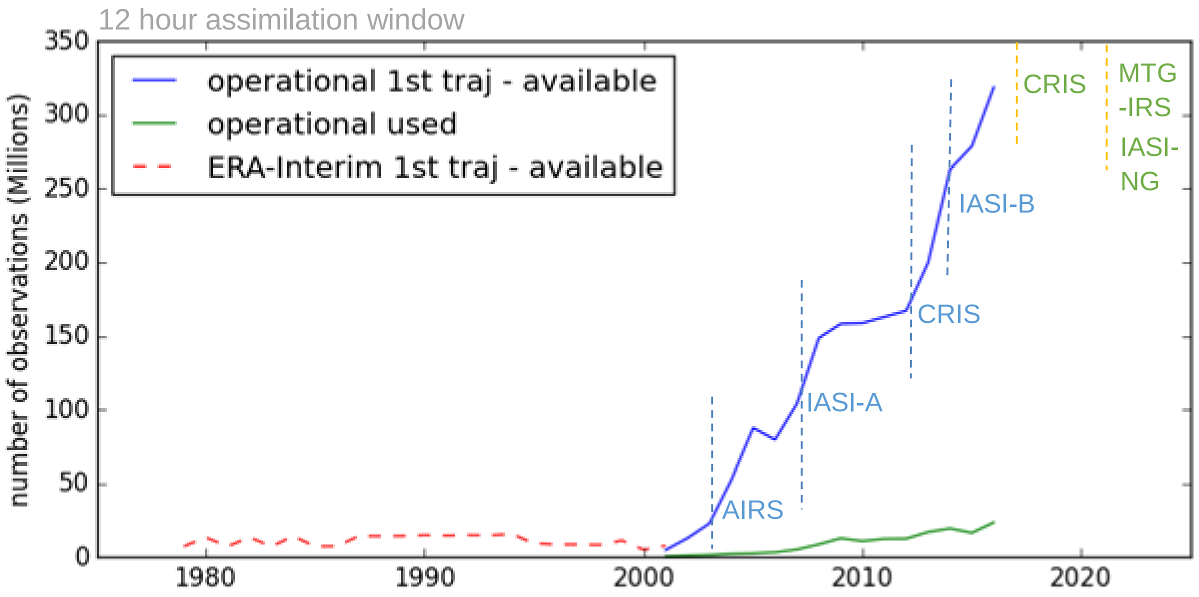
  e.g. OOPS : weak constraint time parallel 4D-Var

# Take home messages:

- Sometimes a few poorly scaling components have a disproportionate impact on overall scalability of an application.
- In this case, minimising unnecessary communications improved scalability.
- Optimisations which gave improvements 10 years ago may no longer be giving the same benefits today.
- Optimisation isn’t a one off; it should be part of business as usual.

# Thanks for listening

# Context: Trends in number of observations

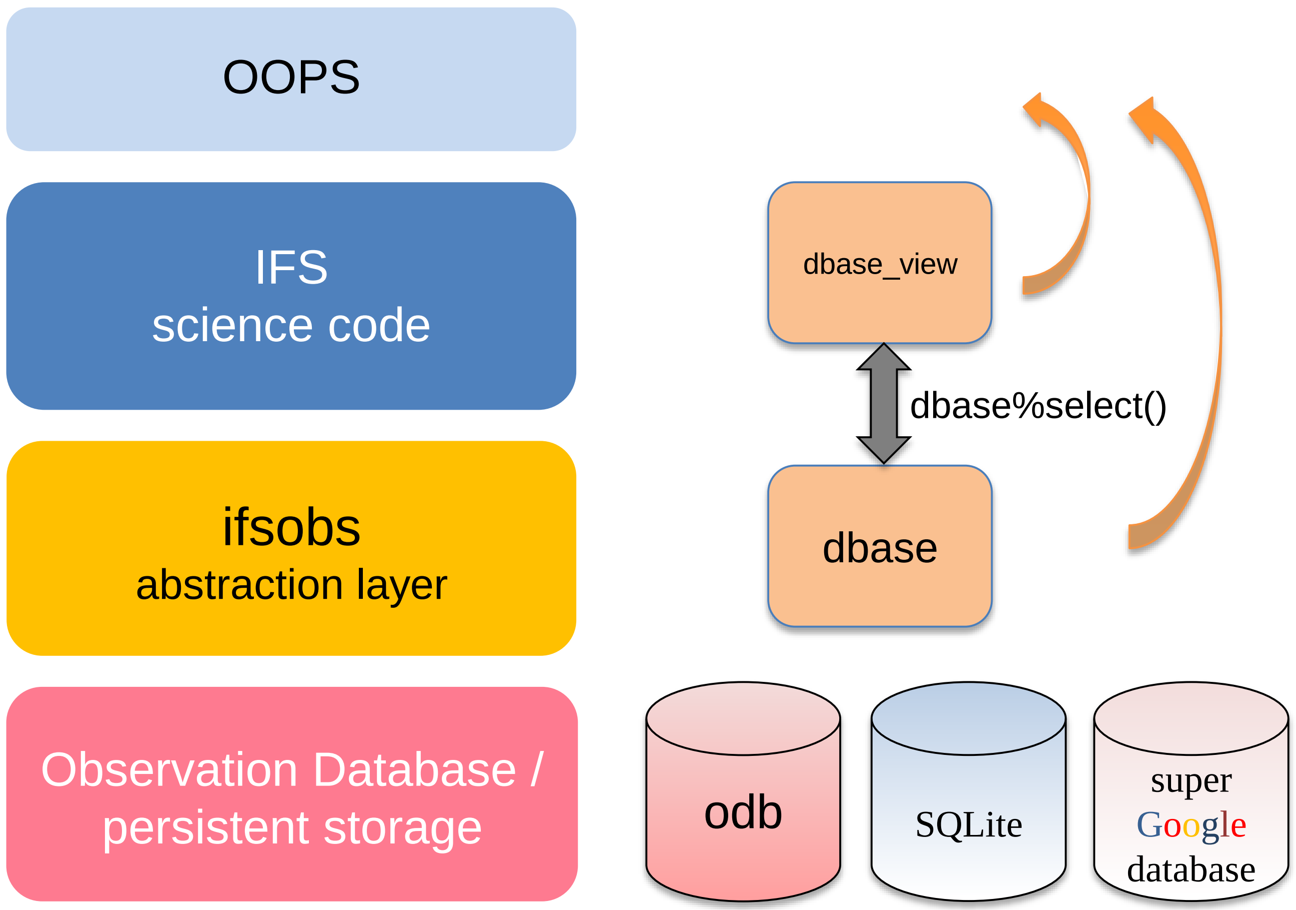



- Explosion in availability of satellite data since 2001.
- Mostly from infra-red sounders.
- Only around 5-10% are actively assimilated:
  - most are cloud-screened, thinned or blacklisted.
- Trend expected to continue with new CRIS, MTG-IRS, IASI-NG
- Projection for next 10 years:
  - x2 (current usage trends)
  - x10 (more aggressive usage)

Other factors:
- Long window 4D-Var? 5 day = x10
- Spatial error correlations? Less thinning.
- Clear sky only → all-sky.

# ifsobs

## - a data acces layer for observations in IFS:




**Decouples** IFS from the underlying database / file format;
improved **modularity** of IFS.

IFS only interacts with dbase and dbase_view objects,
**not** with the underlying database directly.