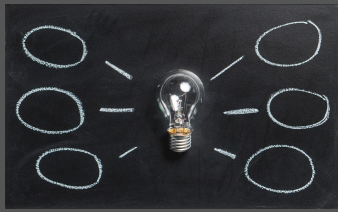
Department of Computer Science

HPS

University of Reading

Challenges and Approaches for Extreme Data Processing



Limitless Storage
Limitless Possibilities

https://hps.vi4io.org

Julian M. Kunkel

EPSRC Centre for Doctoral Training
Mathematics of Planet Earth

March 2020

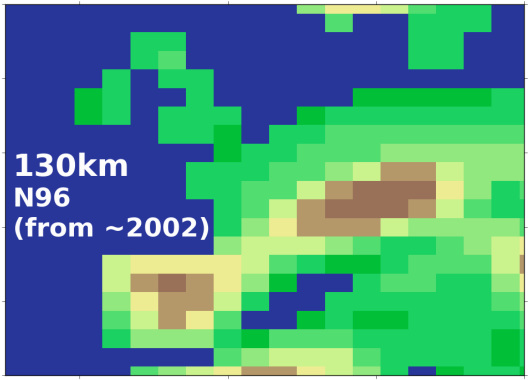
LIMITLESS **POTENTIAL** | LIMITLESS **OPPORTUNITIES** | LIMITLESS **IMPACT**

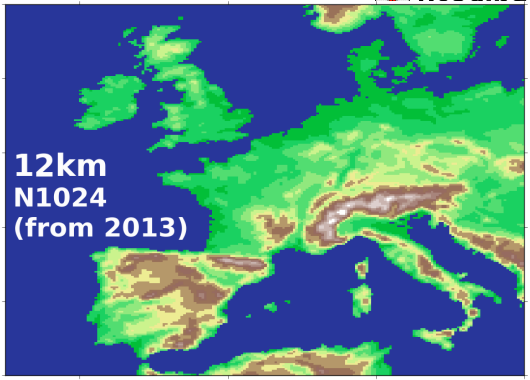
Outline

University of Reading

Climate/Weather Simulation: Volume . . .

University of Reading





One “field-year”: 26 GB

1 field, 1 year, 6 hourly, 80 levels

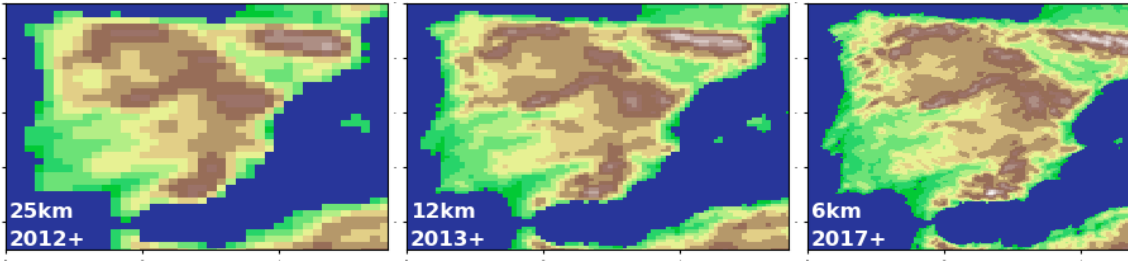
1 x 1440 x 80 x 148 x 192

One “field-year”: 6 TB

1 field, 1 year, 6 hourly, 180 levels

1 x 1440 x 180 x 1536 x 2048

Volume — The Reality of Global 1km Grids



1 km is the current *European Network for Earth System Modelling (ENES)* goal!

Consider N13256 (1.01km, 26512x19884):

- 1 field, 1 year, 6 hourly, 180 levels
- 1 x 1440 x 180 x 26512 x 19884 = 1.09 PB
- but with 10 variables hourly: $>$ 220 TB/day!

Requires a massively parallel environment

High-Performance Computing (HPC)

University of Reading

Definitions

- HPC: Field providing massive compute resources for a computational task
  - Task needs too much memory or time for a normal computer
  - $\Rightarrow$ Enabler of complex scientific simulations, e.g., weather, astronomy
- Supercomputer: aggregates power of many compute devices

Example Supercomputers

DKRZ: Mistral

- Compute: 3,000 dual socket nodes
  - Linpack: 3 Petaflop/s (10^{15})
- Storage: 52 Petabyte
  - 10k HDDs, 300 servers
  - Cost: 6 M€
- Energy: 1.5 MW

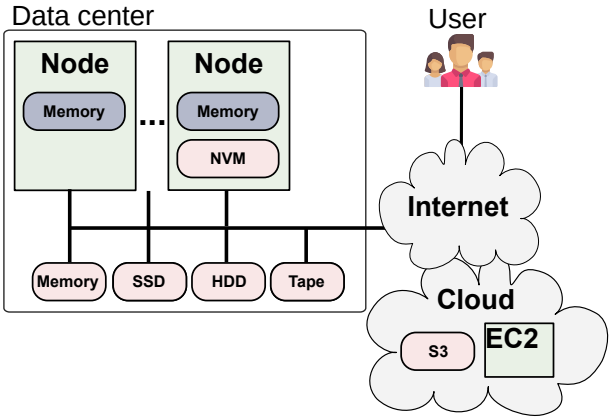
Oak Ridge National Laboratory: Summit

- Compute: 4,608 dual socket nodes
  - 6 GPUs per Node
  - Linpack: 148 Petaflop/s
- Storage: 200 Petabyte
  - NVM: 1.6 TB/Node
- Energy: 13 MW

UK will spend £1.2 Billion next 10 years for weather and climate supercomputer(s)!

Supercomputers & Data Centers

University of Reading



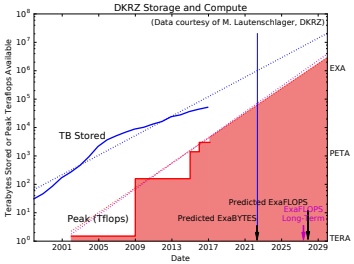
Juwels Cluster

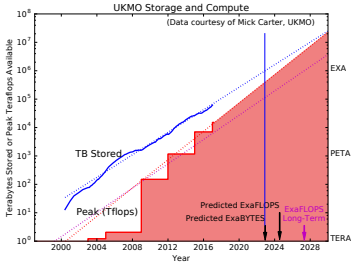
Copyright: Forschungszentrum Jülich

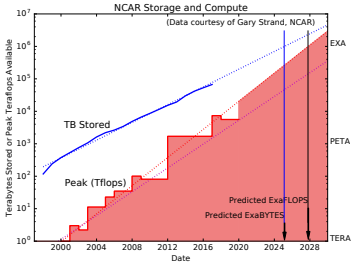
The Exabyte Challenge in Climate and Weather

University of Reading

- We reach Exabyte (10^{18} Bytes) long before Exascale (10^{18} FLOPs)
- Comparing German Climate Computing Center, MetOffice and NCAR storage

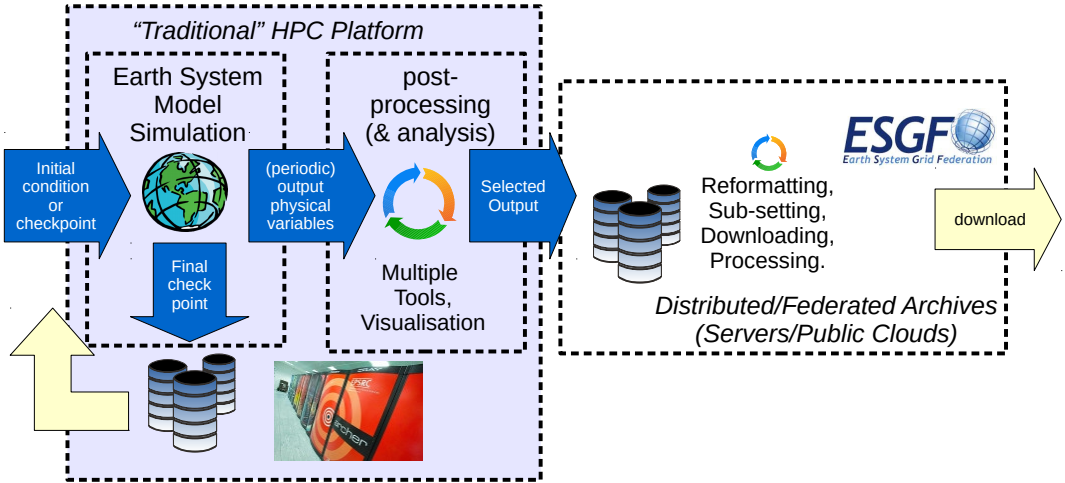






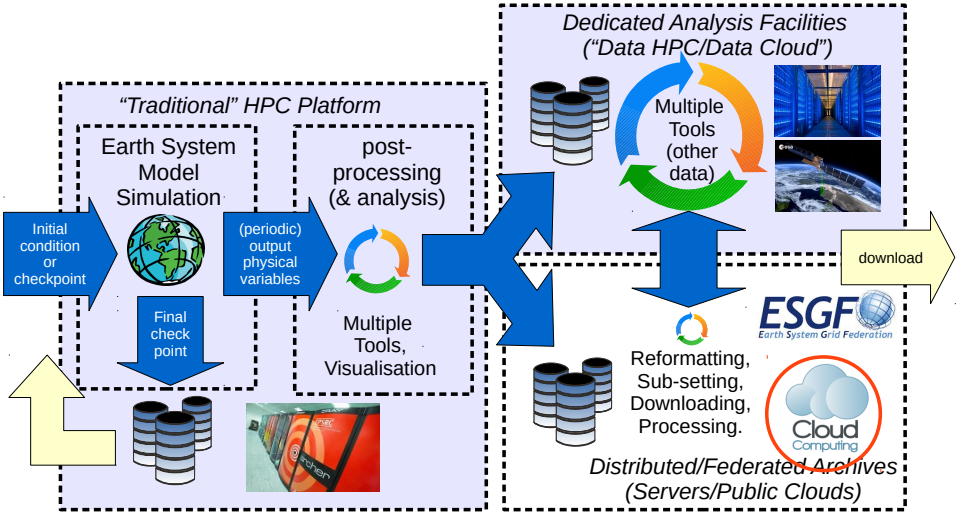
Long-term predictions uses historical data (before 2000)

How we used to do it: From Supercomputer to Download



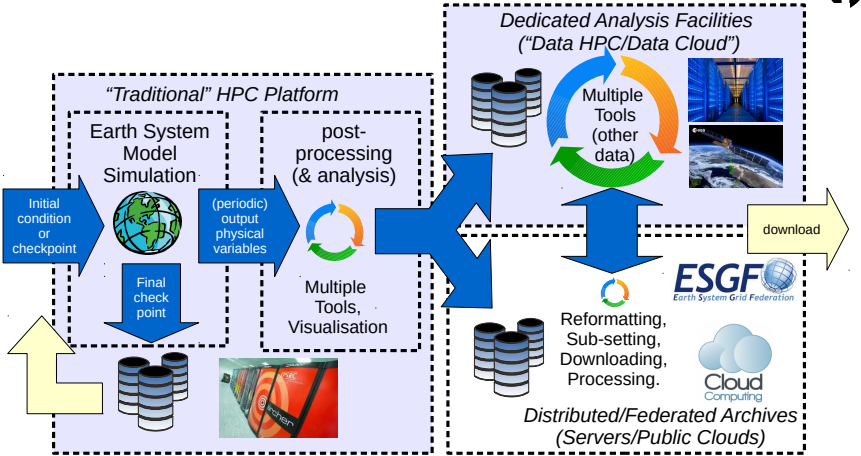
Now: Many Different Supercomputing Environments

University of Reading



Now: Many Different Supercomputing Environments

University of Reading



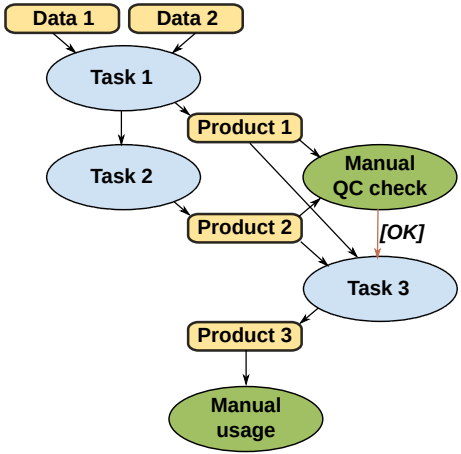
Multiple Roles, at least:

Model Developer, Model Tinkerer, Runner, Expert Data Analyst, Service Provider, Data Manager, Data User

Workflows

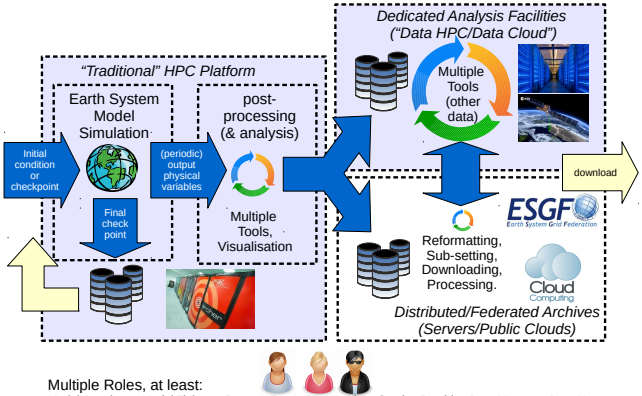
University of Reading

- Insight: What users are interested in
- Consider workflow from 0 to insight
  - Needs input
  - Produces output data
  - Uses tasks
    - Parallel applications
    - Big data tools
    - Manual analysis / quality control
  - A task may run on one of aforementioned systems
  - May need month to complete
  - Manual tasks are unpredictable



Discussion

- Time: 2 minutes (2 minutes discussion)
- List **compute/data challenges** you face or you expect in such an environment



Climate/Weather Workflows

University of Reading

Challenges Related to Data and IO

- Programming of efficient workflows
- Efficient analysis of data
- Data management: Organizing data sets
- Ensuring reproducability of workflows/provenance of data
- Meeting the compute/storage needs in future complex hardware landscape

Boundary Conditions: Expected Data Characteristics in 2020+

- Velocity: Input 5 TB/day (for NWP; reduced data from instruments)
- Volume: Data output of ensembles in PBs of data
- Variety: Various file formats, input sources
- Usability: Data products are widely used by 3rd parties

General I/O Challenges

University of Reading

- Large data volume and high velocity
- Data management practice does not scale and is not portable
  - Cannot easily manage file placement and knowledge of what file contains
  - Hierarchical namespaces does not reflect use cases
  - Individual strategies at every site
- Data conversion/merging is often needed
  - To combine data from multiple experiments, time steps, ...
- The storage stack becomes more inhomogeneous
  - Non-volatile memory, SSDs, HDDs, tape
  - Node-local, vs. global shared, partial access (e.g., racks)
- Suboptimal performance & performance portability
  - Users cannot properly exploit the hardware / storage landscape
  - Tuning for file formats and file systems necessary at the *application* level

Data Management

University of Reading

High-Level questions relevant to scientists

- What experiments did I run yesterday?
- Show me the data of experiment X, with parameters Z...
- Cleanup unneeded temporary data from experiment X
- How can I free some of the occupied space to save 10% money?
- Compare the mean temperature for one experiment across different model versions
- Actually: Where is my data?

Data Management

University of Reading

What scientists (need to) care about right now

- Achieving good efficiency (performance) when executing a workflow
  - Taming costs (in Cloud) or prescribed usage limits (in HPC)
- Organization of the (hierarchical) namespace
- Choosing the storage location/s
  - Often (if not policy driven): when to archive/how?
- Management of the workflows and applications that generate data
- Loads of scripts to run applications and manage data
- The reproduction of experiments after their data is archived
- Sometimes: May use high-level databases to manage data
  - The allow to ingest metadata and organize data accordingly

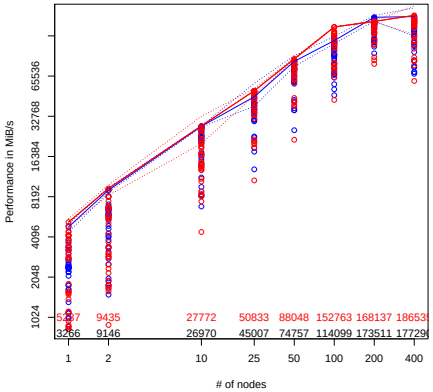
The Performance Challenge

University of Reading

- DKRZ file systems offer about 700 GiB/s throughput
  - However, I/O operations are typically inefficient: Achieving 10% of peak is good
- Influences on I/O performance
  - Application's access pattern and usage of storage interfaces
  - Network congestion
  - Slow storage media (tape, HDD, SSD)
  - Concurrent activity – shared nature of I/O
  - Tunable optimizations deal with characteristics of storage media
  - These factors lead to complex interactions and non-linear behavior

Illustration of Performance Variability

- Best-case benchmark: optimal application I/O
  - Independent I/O with 10 MiB chunks of data
  - Real-world I/O is sparse and behaves worse
- Configurations vary:
  - Number of nodes the benchmark is run
  - Processes per node
  - Read/Write accesses
  - Tunable: stripe size, stripe count
- Optimal performance:
  - Small configuration: 6 GiB/s per node
  - Large configurations: 1.25 GiB/s per node
- Best setting depends on configuration!

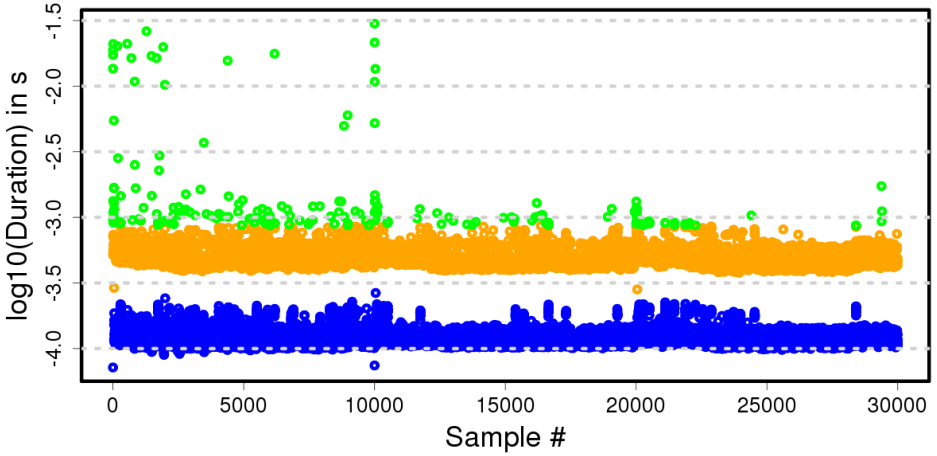


A point represents one configuration
(color: read/write)

Illustration of Performance Variability (2)

University of Reading

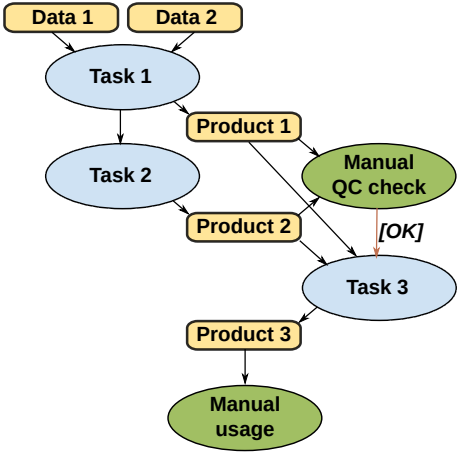
- Rerunning the same operation (access size, ...) leads to performance variation
- Individual measurements – 256 KiB sequential reads (outliers purged)



Workflows

University of Reading

- Consider workflow from 0 to insight
  - Needs/produces data
  - May need month to complete
  - Manual tasks are unpredictable
- Not well described in HPC
  - Mostly hardcoded in scripts
  - Users must know about HPC
- Can we exploit workflows?
  - Can we optimize resources?
  - Enforce requirements to preserve data



Scenario: Large Simulation

University of Reading

- Assume large scale simulation, timeseries (e.g., 1000 y climate)
- Assume manual data analysis needed (but time consuming)
- We need all 1000 y for detailed analysis!

A typical workflow execution

- Run simulation for 1000 year simulation time
  - Store various data on (online) storage
  - Keep checkpoints to allow reruns
  - Maybe backup data in **archive**
- Explore data to identify how to analyze data
- At some point: Run the analysis on all data
- Problem: Occupied storage capacity is expensive!

Alternative Workflows Done by Scientists

University of Reading

Recomputation

- Run climate simulation
  - Store checkpoints
  - Store only selected data (wrt. resolution, section, time)
- Explore data
  - Run recomputation to create needed data (e.g., last year)
- At some point: run analysis across all data needed
- This is a manual process, must consider
  - Runtime parameters
  - System configuration/available resources
  - We are trading compute cycles vs. storage
- It would be great if a system considers costs and does this automatically

Another Alternative Workflow

University of Reading

Provided by more intelligent storage and better workflows

- Run simulation
  - Store checkpoints on node-local storage
    - Redundancy: from time to time restart from another node
  - Store selected data on online storage (e.g., 1% of volume)
    - Also store high-resolution data sample (e.g., 1% of volume)
  - Store high-resolution data directly in a cold **archive**
- Explore data on snapshot
- Month later: schedule analysis of data needed
  - The system retrieves data from the **archive**
  - Performs the scheduled operations on **streams** while data is pulled in
  - Informs user about analysis progress
- Some people do this manually or use some tools to achieve similarly
  - Aim for domain & platform independence and heterogenous HPC landscapes

Outline

Advanced Computing for Environmental Science (ACES)

University of Reading

The ACES research group conducts cutting-edge research in computer science to accelerate environmental science

Research themes address the future of relevant computing and data systems

- Developing software to aid scientific programming
- Exploiting machine learning in environmental science
- Handling the high volume and high velocity data
- Processing in next-generation compute environments (Cloud, HPC+)

Faculty Members (in alphabetical order)

- Julian Kunkel (CS)
- Bryan Lawrence (NCAS + CS)
- Christopher Maynard (MetOffice + CS)

Research Activities & Interest

University of Reading

High-performance storage for HPC

- Efficient I/O
  - Performance analysis methods, tools and benchmarks
  - Optimizing parallel file systems and middleware
  - Modeling of performance and costs
  - Tuning of I/O: Prescribing settings
  - Management of workflows
- Data reduction: compression library, algorithms, methods
- Interfaces: towards domain-specific solutions and novel interfaces

Other research interests

- Application of big data analytics (e.g., for humanities, medicince)
- Domain-specific languages (for Icosahedral climate models)
- Cost-efficiency for data centers in general

Long Term Vision for Data Management

University of Reading

Decisions made by users: What scientists should care about

- Scientific metadata
- Declaring workflows
  - Covering data ingestion, processing, product generation and analysis
  - Data life cycle (and archive/exchange file format)
  - Constraints on: accessibility (permissions), ...
  - Expectations: completion time (interactive feedback human/system)
- Modifying workflows on the fly
- Interactive analysis, e.g., Visual Analytics
- Declaring value of data (logfile, data-product, observation)

Accessing Data via a Semantic Namespace

- Allow to explore data based on user metadata
  - Similar to an MP3 library (search by Genre/Year/Artist/...)
- User-defined properties but provide means to validate schemas

Vision: Separation of Concerns (2)

University of Reading

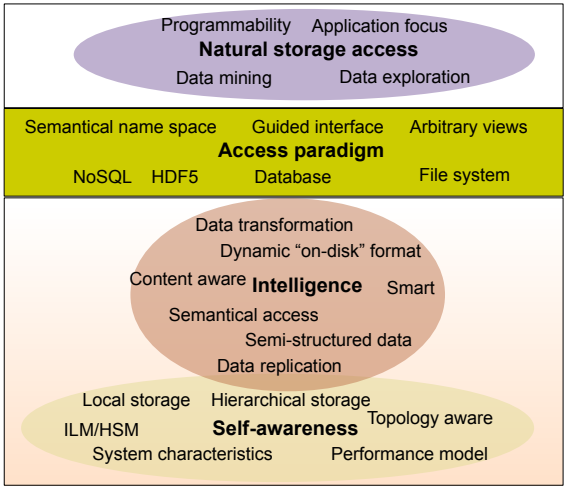
Programmers of models/tools

- Decide about the most appropriate API to use (e.g., NetCDF + X)
- Register compute snippets (analytics) to API
- Do not care **where** and **how** compute/store

Decisions made by the (compute/storage) system

- Where and how to store data, including file format
- Complete management of available storage space
- Performed data transformations, replication factors, storage to use
- Including scheduling of compute/storage/analysis jobs (using, e.g., ML)
- Where to run certain data-driven computations (**Fluid-computing**)
  - Client, server, in-network, cloud, your connected laptop

Personal Vision: Towards Intelligent Storage Systems and Interfaces



- Abstract data interfaces
- Enhanced data management
- Integrate compute/storage engine
- Flexible views on data
- Smart hardware/storage
  - Self-aware systems
  - AI optimized placement
  - Bring-your-own-behavior-model
- Cross sites and cloud

Support Activities

University of Reading

- Community building
  - Bootstrapped: The Virtual Institute for I/O https://www.vi4io.org
  - Supporting: European Open File System (EOFS) https://www.eofs.eu/
  - Organizing: Various I/O workshops
- Awareness: co-created the IO-500 list http://io-500.org
- Beyond teaching:
  - Online teaching platform for C-Programming (ICP project)
  - A **HPC certification program** https://hpc-certification.org
- Standardization:
  - Compression interfaces (AIMES project)
  - Next-Generation I/O Interfaces (https://ngi.vi4io.org)

Ongoing Activity: Earth-Science Data Middleware

University of Reading

- Part of the ESiWACE Center of Excellence in H2020
  - Centre of Excellence in Simulation of Weather and Climate in Europe

  `https://www.esiwace.eu`

ESDM provides a transitional approach towards a vision for I/O addressing

- Scalable data management practice
- The inhomogeneous storage stack
- Suboptimal performance and performance portability
- Data conversion/merging

Earth-System Data Middleware

University of Reading

Design Goals of the Earth-System Data Middleware

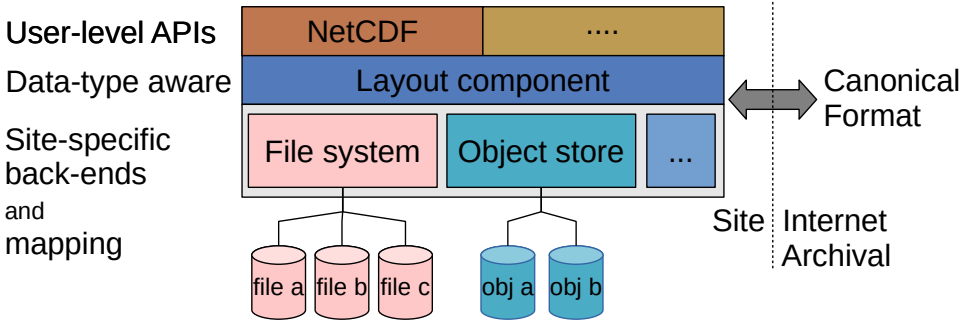
1. Relaxed access semantics, tailored to scientific data generation
   - Understand application data structures and scientific metadata
2. Site-specific (optimized) data layout schemes
   - Based on site-configuration and performance model
   - Site-admin/project group defines mapping
3. Ease of use and deployment particularly configuration
4. Enable a configurable namespace based on scientific metadata

Architecture

University of Reading

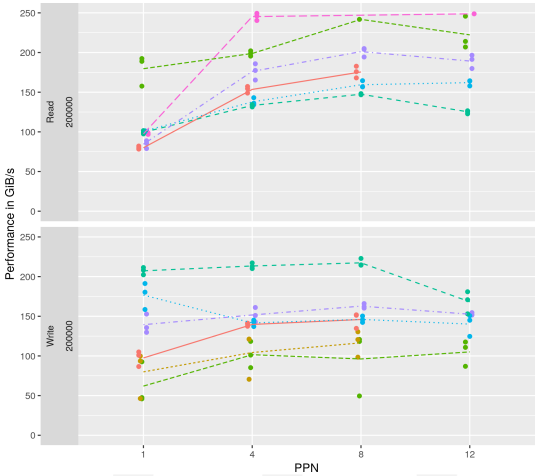
Key Concepts

- Middleware utilizes layout component to make placement decisions
- Applications work through existing API (e.g., NetCDF library)
- Data is then written/read efficiently; potential for optimization inside library



Measured Performance

University of Reading



- IOR serves as baseline (optimal IO)
- Chunking into files increases performance
- Usage of multiple Lustre fs +25%
- Can utilize various storage “tiers”
- *We are still working on it*

Outline

Performance Analysis

University of Reading

Problem

Assessing observed time for I/O is difficult.

What best-case performance can we expect?

Support for analysis – my involvement

- Models and simulation
  - Trivial models: using throughput + latency
  - PIOSimHD: MPI application + storage system simulator
- Tools to capture and analyze system statistics and I/O activities
  - HDTrace – tracing tool for parallel I/O (+ PVFS2)
  - SIOX – tool to capture I/O on various levels
  - Grafana – Online monitoring for DKRZ (support)
- Benchmarks – on various levels, e.g., Metadata (md-workbench, IOR)
- **Statistic model** to determine likely cause based on time

I/O Modeling and Diagnosing Causes with Statistics

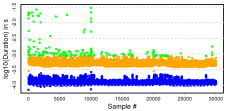
University of Reading

Issue

- Measuring the same operation repeatedly results in different runtime
- Reason: Sometimes a certain optimization is triggered, shortening the I/O path
  - Example strategies: read-ahead, write-behind
  - If data is available in cache, copy it; otherwise, read it directly/store in cache
- Consequence: Non-linear access performance, time also depends on access size
- It is difficult to assess performance of even repeated measurements!

Goal

- Predict likely reason/cause-of-effect by just analyzing runtime
- Estimate best-case time, if optimizations would work as intended



Groupwork

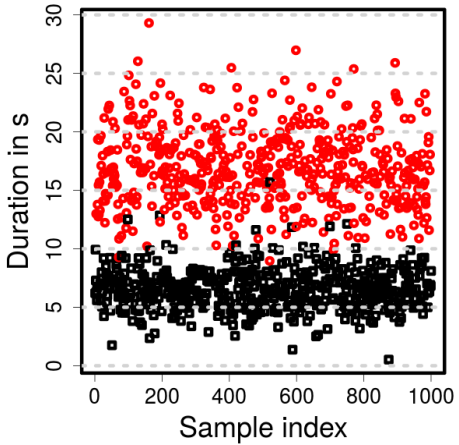
University of Reading

How can we identify relevant processes?

- For arbitrary many (unknown) processes?
- Just by looking at the observation?
- Time: 5 Minutes

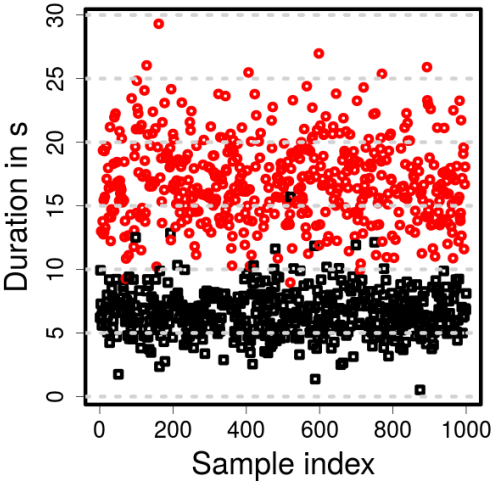
Simulation of Typical IO Behavior

- Assume we have two components
  - Component A is faster than B
  - Either A or B transfer data
  - Cache miss of A leads to transfer for B
- Overlaying three stochastic processes:
  - Two gamma distributions with scale=1
  - Normal distribution (little impact)

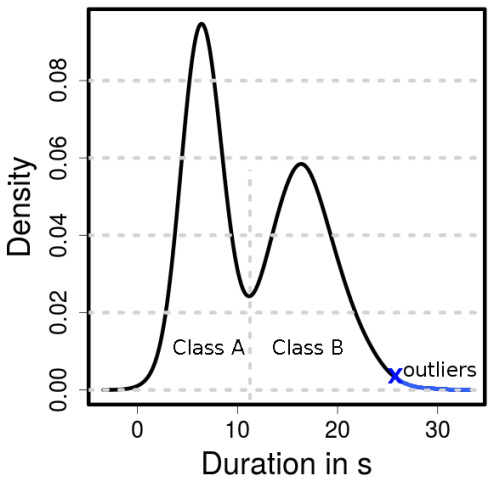


Resulting time for 1000 measurements

Simulated Access Time and Resulting Density

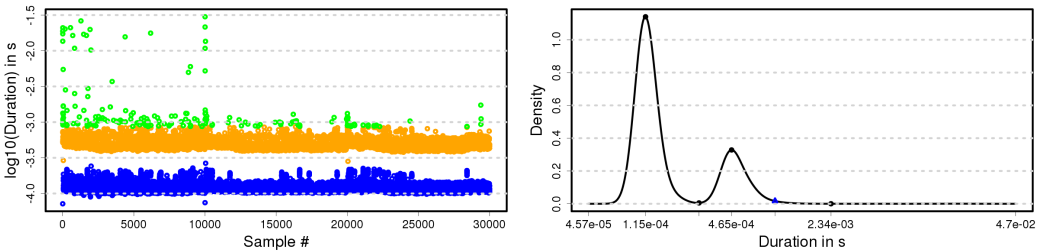


(a) Timeline



(b) Density reveals two classes

Comparing Density Plot with the Individual Data Points

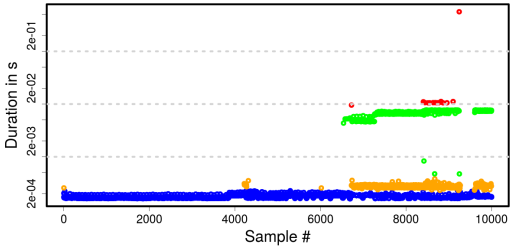


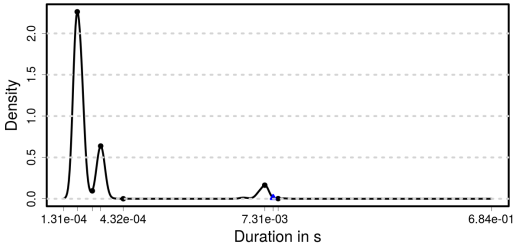
Duration for sequential reads with 256 KiB accesses (off0 mem layout)

Algorithm for determining classes (color schemes)

- Create density plot with Gaussian kernel density estimator
- Find minima and maxima in the plot
- Assign one class for all points between minima and maxima
- Rightmost hill is followed by cutoff (blue) close to zero $\Rightarrow$ outliers (unexpected slow)

Write Operations





Results for one write run with sequential 256 KiB accesses (off0 mem layout).

Known optimizations for write

- Write-behind: cache data first in memory, then write back
- Write back is expected to be much slower

This behavior can be seen in the figure

Approach

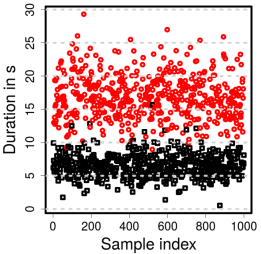
Assumptions

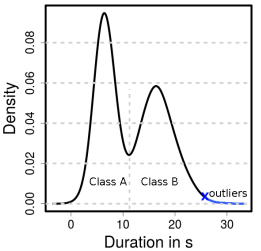
- Each “class” is caused another optimization/technology
  - Assign an observation to the likely class
  - This may lead to (tolerable) errors
- Behavior not visible on the density plot is irrelevant

⇒ The strategy identifies relevant “performance factors”

Concept

1. Repeatedly measure time for I/O with a given size
2. Construct the density graph and identify clusters
3. A class is caused by (at least) one performance factor
4. Build a model to assign the cluster across “sizes“
5. Optional: Identify the root cause for the cluster
6. Assign appropriate names, e.g., “client-side cached”





Approach: Models

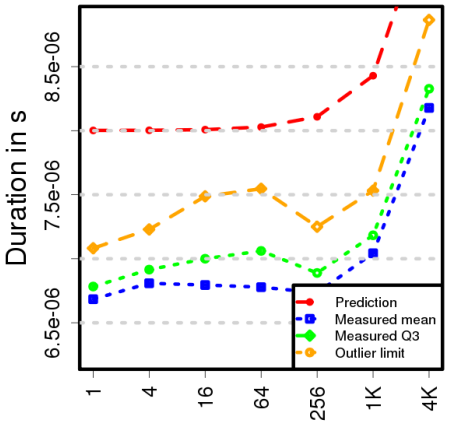
University of Reading

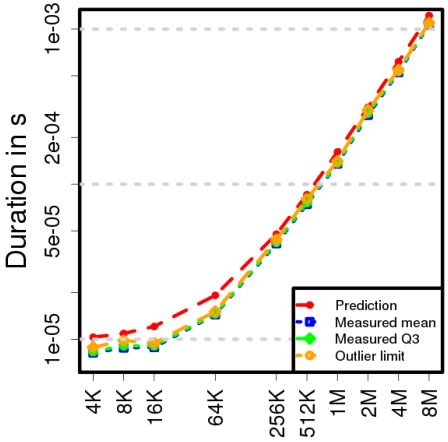
- Apply a family of linear models predicting time; $lm(size) = c + f(size)$
  - Assume time correlates to the operation's size
  - Each model represents a condition C (cached, in L1, ...)
  - $t_C(size) = lm(size) + lm'(size) + ...$ and check $min(|t_C - \hat{t}|)$
- Assume the conditions for the closest combination are the cause
- Ignore the fact of large I/O requests with mixed conditions
  - i.e., some time of the operation may be caused by C and some by C'

Example models

- $t(size) = m$: Data is discarded on the client or overwritten in memory
- $t(size) = m + c(size)$: Data is completely cached on the client ...

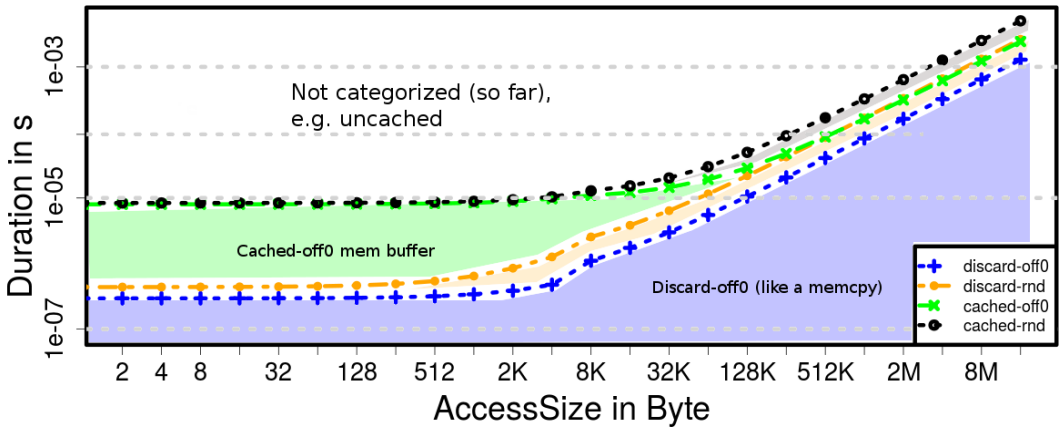
Model for Reading Cached Data





Model accuracy for reading cached data (off0 locality in memory and file). Other figures look similar

Resulting Performance Models for Read

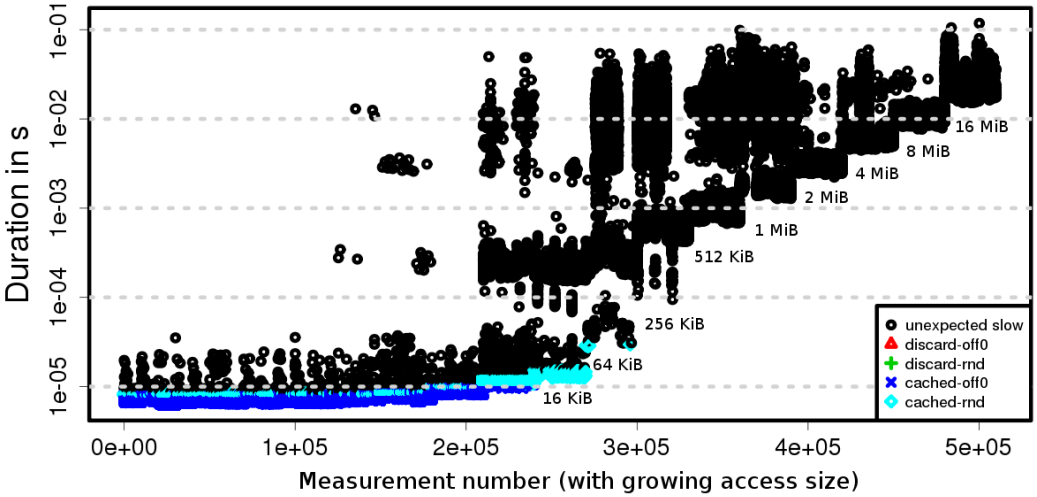


Read models predicting caching and memory location.

Using the Model to Identify Anomalies

University of Reading

Using the model, the figure for reverse access shows slow-down (by read-ahead)



Validation: Classify Different Patterns

University of Reading

Experiment state-mem-file	cached off0	cached rnd	discard off0	discard rnd	uncached
D-reverse-off0 R	46	54	0.3	0.03	0.004
C-off0-off0 R	0	34	60	6.1	0.29
C-seq-off0 R	0	0	52	47	0.31
C-seq-reverse R	0	0	42	4.3	54
C-seq-rnd8 R	0	0	30	44	26
C-seq-rnd R	0	0	26	5.6	68
C-seq-seq R	0	0	48	9.5	42
C-seq-stride8,8 R	0	0	28	8.8	63
C-off0-rnd R	0	2e-04	18	1.9	80
U-off0-rnd R	0	0	0.01	0.15	100
U-seq-seq R	0	0	57	6.1	37
C-off0-rnd W	0	0	0	0.003	100
C-off0-seq W W	0	0	40	17	42
C-seq-seq W	0	0	40	12	48
C-off0-reverse W	0	0	71	14	15

Model predictions classes in % of data points for selected memory & file locations – access size is varied.

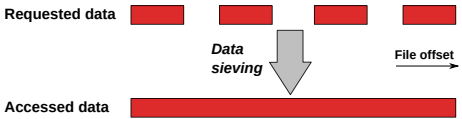
Outline

University of Reading

Prescriptive Analysis: Learning Best-Practises for DKRZ

University of Reading

- Performance benefit of I/O optimizations is non-trival to predict
- Non-contiguous I/O supports data-sieving optimization
  - Transforms non-sequential I/O to large contiguous I/O
  - Tunable with MPI hints: enabled/disabled, buffer size
  - Benefit depends on system AND application



- Data sieving is difficult to parameterize
  - What should be recommended from a data center's perspective?

Measured Data

University of Reading

- Simple single threaded benchmark, vary access granularity and hole size
- Captured on DKRZ porting system for Mistral
- Vary Lustre stripe settings
  - 128 KiB or 2 MiB
  - 1 stripe or 2 stripes
- Vary data sieving
  - Off or On (4 MiB)
- Vary block and hole size (similar to before)
- 408 different configurations (up to 10 repeats each)
  - Mean arithmetic performance is 245 MiB/s
  - Mean can serve as baseline “model”

System-Wide Defaults

University of Reading

- Comparing a default choice with the best choice
- All default choices achieve 50-70% arithmethic mean performance
- Picking the best default default for stripe count/size: 2 servers, 128 KiB
  - 70% arithmetic mean performance
  - 16% harmonic mean performance ⇒ some bad choices result in very slow performance

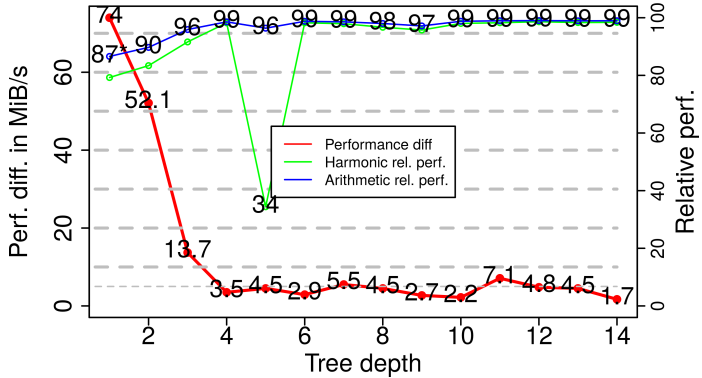
Default Choice			Best	Worst	Arithmethic Mean			Harmonic Mean	
Servers	Stripe	Sieving	Freq.	Freq.	Rel.	Abs.	Loss	Rel.	Abs.
1	128 K	Off	20	35	58.4%	200.1	102.1	9.0%	0.09
1	2 MiB	Off	45	39	60.7%	261.5	103.7	9.0%	0.09
2	**128 K**	**Off**	87	76	**69.8%**	209.5	92.7	8.8%	0.09
2	2 MiB	Off	81	14	72.1%	284.2	81.1	8.9%	0.09
1	128 K	On	79	37	64.1%	245.6	56.7	15.2%	0.16
1	2 MiB	On	11	75	59.4%	259.2	106.1	14.4%	0.15
2	**128 K**	**On**	80	58	**68.7%**	239.6	62.6	**16.2%**	0.17
2	2 MiB	On	5	74	62.9%	258.0	107.3	14.9%	0.16

Performance achieved with any default choice

Applying Machine Learning

University of Reading

- Building a classification tree with different depths
- Even small trees are much better than any default
- A tree of depth 4 is nearly optimal; avoids slow cases



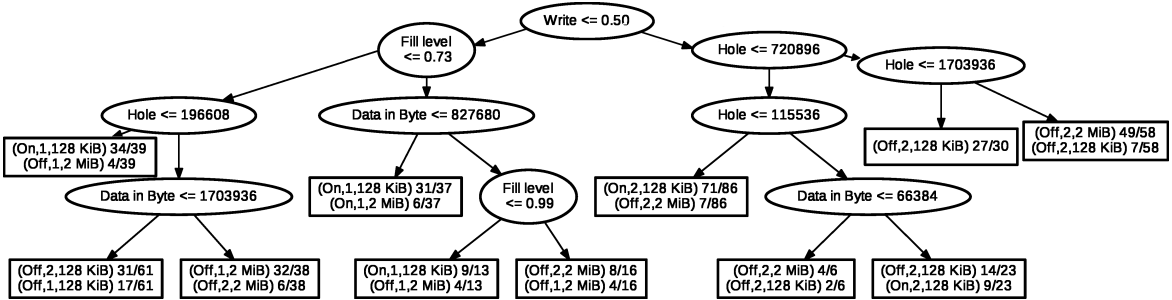
Perf. difference between learned and best choices, by maximum tree depth, for DKRZ's porting system

Decision Tree & Rules

Extraction of knowledge from a tree

University of Reading

- For writes: Always use two servers; For holes below 128 KiB ⇒ turn DS on, else off
- For reads: Holes below 200 KiB ⇒ turn DS on
- Typically only one parameter changes between most frequent best choices



Decision tree with height 4. In the leaf nodes, the settings (Data sieving, server number, stripe size) and number of instances for the two most frequent best choices

Outline

University of Reading

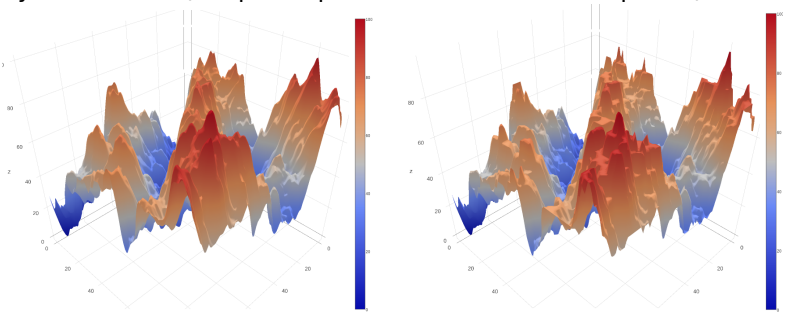
Compression Research: Involvement

University of Reading

- Development of algorithms for lossless compression
  - MAFISC: suite of preconditioners for HDF5, aims to pack data optimally
    Reduced climate/weather data by additional 10-20%, simple filters are sufficient
- Cost-benefit analysis: e.g., for long-term storage MAFISC pays of
- Analysis of compression characteristics for earth-science related data sets
  - Lossless LZMA yields best ratio but is very slow, LZ4fast outperforms BLOSC
  - Lossy: GRIB+JPEG2000 vs. MAFSISC and proprietary software
- **Development of the Scientific Compression Library (SCIL)**
  - Separates concern of data accuracy and choice of algorithms
  - Users specify necessary accuracy and performance parameters
  - Metacompression library makes the choice of algorithms
  - Supports also new algorithms
  - Ongoing: standardization of useful compression quantities
- A method for system-wide determination of data characteristics
  - Method has been integrated into a script suite to scan data centers

Example for Data Compression

Synthetic data (Simplex, options 206, 2D: 100x100 points)



Right image uses lossy compression (Sigbits 3bits, ratio 11.3:1)

Supported Quantities in SCIL

University of Reading

Accuracy quantities:

absolute tolerance: compressed can become true value ± absolute tolerance

relative tolerance: percentage the compressed value can deviate from true value

relative error finest tolerance: value defining the absolute tolerable error for relative compression for values around 0

significant digits: number of significant decimal digits

significant bits: number of significant decimals in bits

Performance quantities:

compression speed: in MiB or GiB, or relative to network or storage speed

decompression speed: in MiB or GiB, or relative to network or storage speed

Supplementary quantities:

fill value: a value that scientists use to mark special data point

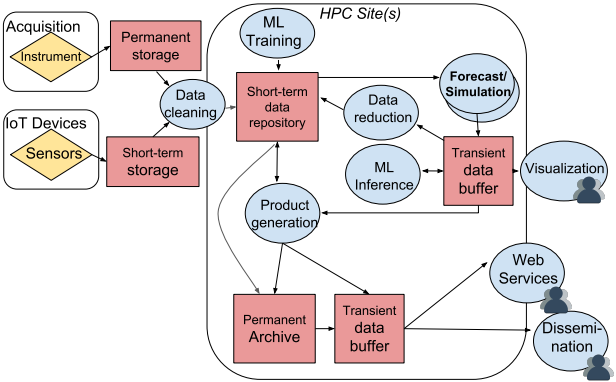
Outline

Summary

University of Reading

- Data management at scale is challenging for users/domain scientists/experts
- Parallel I/O is complex
  - System complexity and heterogeneity increases significantly
  - $\Rightarrow$ Expected and measured performance is difficult to assess
  - HPC users (scientists) and data centers need methods and tools
- Tools, statistics and machine learning help with key aspects:
  - Diagnosing causes and identify anomalies
  - Predicting performance
  - Prescribing best practices
- I work towards intelligent systems to increase insight and ease the burden for users
  - Novel interfaces are needed to unleash the full potential of system resources
- We need smarter data handling not only in climate/weather

Smarter Climate/Weather Workflows in 2020+



- IoT (and mobile devices)
  - Additional data provider
  - Improves short-term weather prediction
- Machine learning support
  - Localize known patterns
  - Interactive use: Visual analytics
- Data reduction
  - Output is triggered by ML events
  - Compress data of ensembles