Hewlett Packard
Enterprise

Zetta-what and Zetta-how: What comes next after Exascale?

Larry Kaplan
Senior Distinguished Technologist

ModSim 2025
August 15th, 2025

What is Zettascale and what comes next?

- Traditional interpretation: 10^{21} "things" – i.e., operations per sec, bytes of memory or storage, etc.
- Focus for this talk is on computational performance/operations
  - Data movement also important, especially for energy, not discussed here
- In HPC, previous assumptions about these operations no longer hold
  - No more pervasive 64-bit floating-point operations, or even floating-point operations at all in some cases
- AI/ML, and the enormous market for it, use different operations
  - Vendor momentum far greater in this space given the relative market size
- Other potential technologies may also be brought to bear
- Measurement is no longer straightforward
- **Software will play an increasing role**
  - and not just in application implementation

Created via gemini.google.com

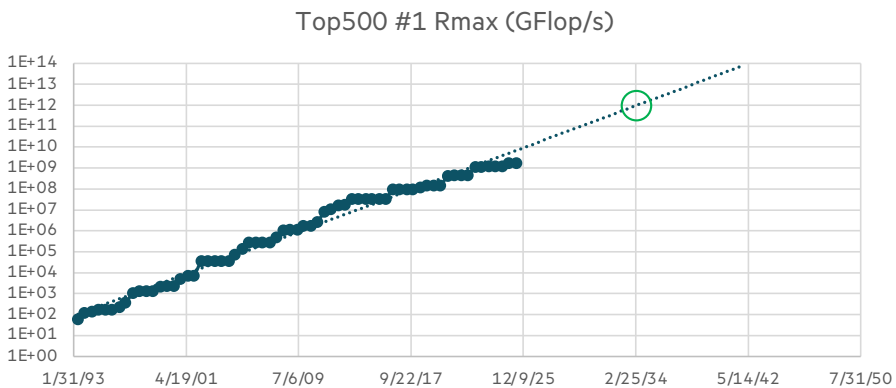
We face a hybrid future – what are the trends and challenges?

64-bit Floating-Point Trends

Status and progression

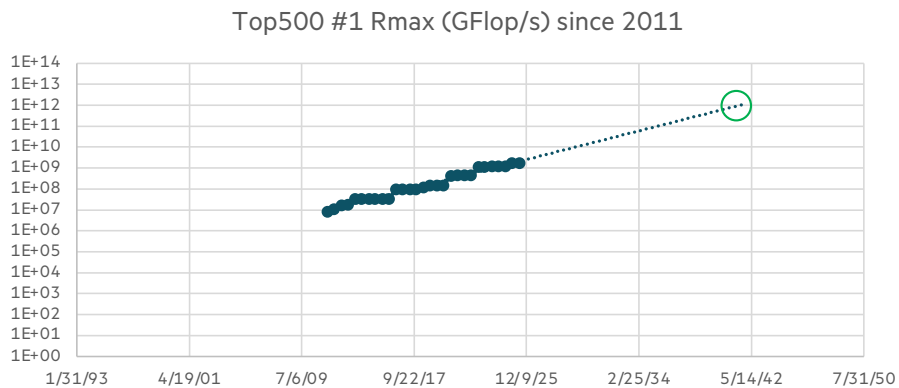
Where are we today?

- Current standard is 64-bit floating-point operations per second
  - Provides a consistent basis for comparisons
- Several 64-bit exaflop systems now exist
  - Some may not be publicly advertised



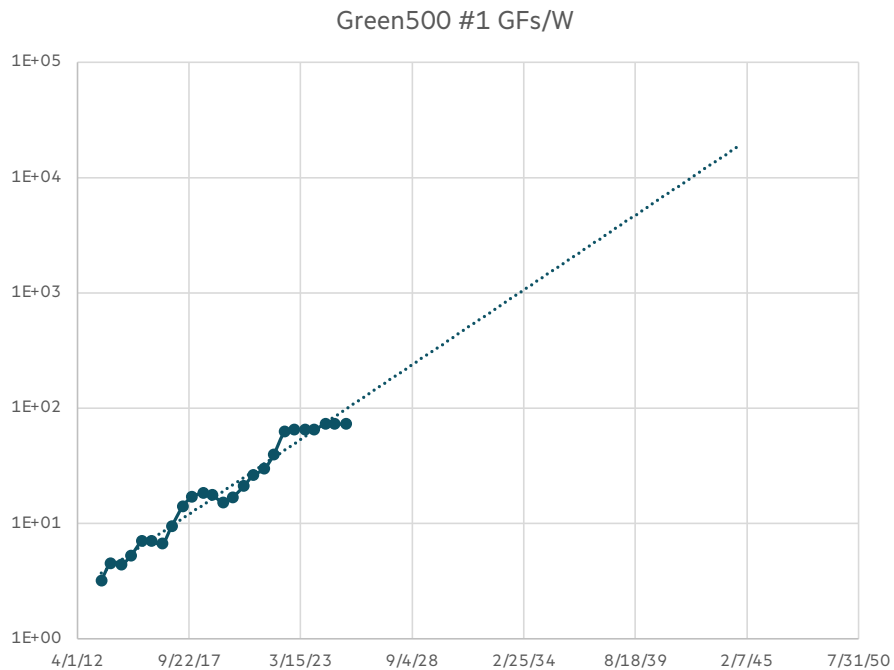
Where might we be going?

- Looking at trends is getting harder given ”end” of Dennard Scaling and other factors
- Will there be discontinuities that push beyond the current trend?
  - Process, transistor design, packaging
- Or will the trends continue to flatten?



64-bit Floating-Point Trends
Challenges

- HPC is no longer driving the industry
- Industry momentum is moving away from 64-bits
  - It may not disappear, but further improvements appear to be slowing
- Do we really need 64-bit floating-point math for everything?
  - Giving up on it pervasively increases application complexity, increases software burden
- Reliability also a potential issue
  - FIT rates associated with # of sockets not transistors
  - Socket count depends on flops/watt and watts/socket
  - Finer-grained silicon geometries also introduce issues
  - Resilient SW solutions important at scale



Mixed and Reduced Precision

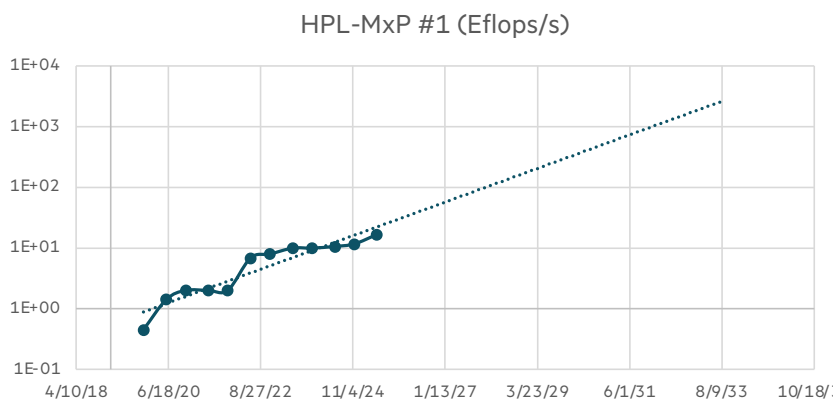
Status and progression

Where are we today?

- For many science domains, full 64-bit precision is not required
  - Greatly depends on the dynamic range of the physics equations and input data sets
- Good mixed-precision results for specific algorithms starting to be published
- Lower data volume can help reduce data movement concerns
- Must note that 64-bit floating point is already an approximation!
  - Discretization will always lose information, the question is how important is the information that's lost

Where might we be going?

- HPL MxP provides some guidance on potential performance trends
  - Current trend shows "effective" Zettaflops by 2032
- Expect increases in lower precision operation performance to continue
  - Moreso than for 64-bit flops
  - Improvements from ISA, transistor design, process technology
- Other algorithms being studied and implemented – "Ozaki Splitting" shows promise



Mixed and Reduced Precision
Challenges

- Each science domain and algorithm needs precision sensitivity analysis
  - Some science domains require 64-bits (and more)
  - Several issues to consider such as range, de-norms, rounding, stability
- How do we count these operations for Zettascale?
  - Just count them?
  - Figure out the number of 64-bit operations they replace?
    - HPL-MxP strategy
- How can we really compare the different precisions when some domains require more than others?

Created via Canva.com

AI Surrogate Component Models

Status and progression

Where are we today?

- Surrogate component models provide an efficient way to replace difficult or computationally complex computations with training and inference
  - Orders of magnitude less computation required for inference than for direct simulation
- Example: fog prediction can be very expensive to simulate, good results possible via AI/CNNs
  - ~200 Mops (AI) vs ~500 Gops (simulation) for 50 km^2 study at 1 km resolution
- Other cases aren't less expensive but are more accurate with a surrogate (upcoming slide)

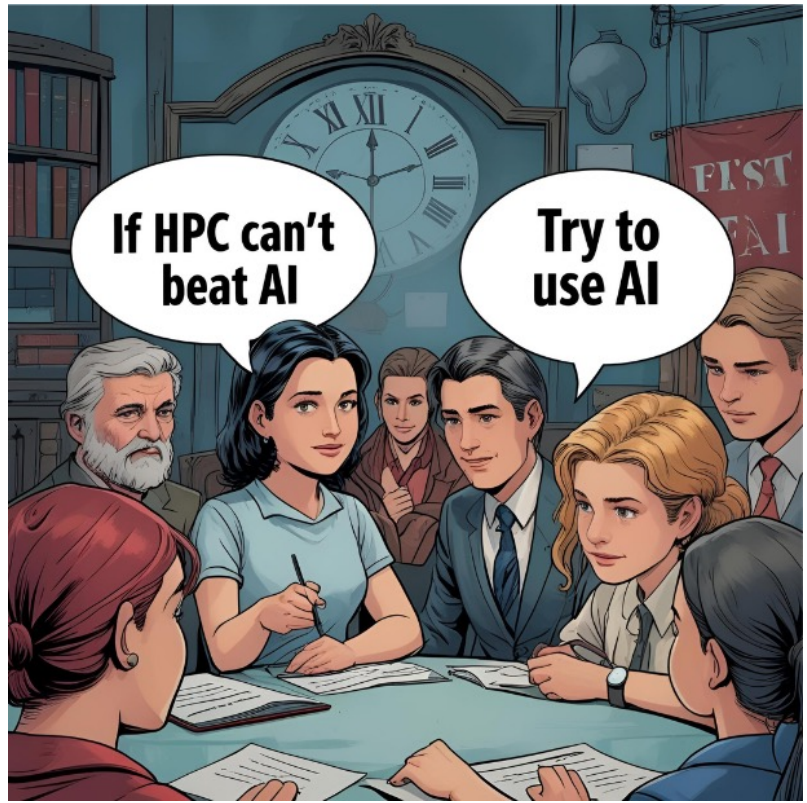
Where might we be going?

- Training itself has already reached Zettascale by operation counts
  - Inference likely can as well though operation counts are highly dependent on AI model
- LLMs are not the only form of AI and in fact are often not the best choice for surrogates
- Trend looks aggressive, but questions remain on maintaining the rate of improvement
  - Trends driven more by architecture than silicon – e.g., ISA, low precision
  - Is there a point of diminishing returns for training?

AI Surrogate Component Models
Challenges

- Where does the training data come from?
  - Real world in some cases, high-precision simulations in others
  - How often is re-training required?
  - How many times is a given training used?
- Integration of surrogate components into simulations
  - How tightly coupled must they be?

- How do we count operations for Zettascale?
  - Just count the reduced precision operations used for inference?
  - Figure out the number of 64-bit operations the surrogate replaced?
  - Issues of training versus inference – what should we count?
    – What about retraining?

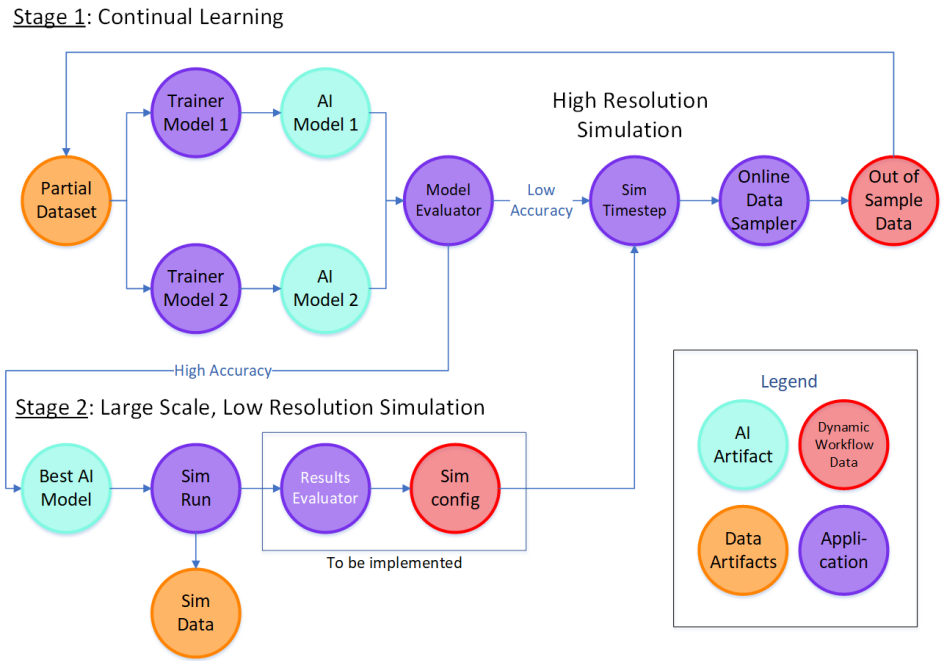


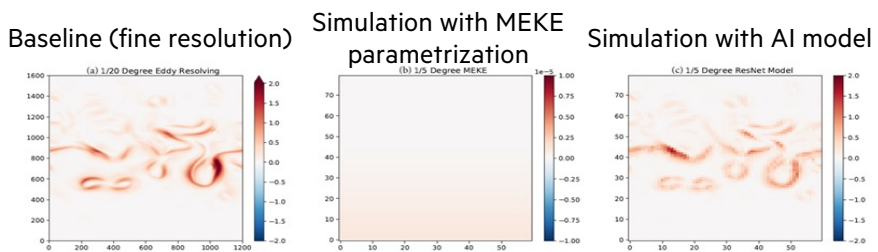
Created via Canva.com

Convergence of HPC and AI: Acceleration of Climate Modeling Example

- AI model of eddy quantities (turbulence) accelerates CFD simulation ≥10x
- Model periodically retrained for out-of-distribution data
- Data tracking by HPE Common Metadata Framework (CMF)
- Workflow orchestration and coordination by HPE SmartSim
  - Large scale, coarse resolution simulation with AI inference
  - Small scale, fine resolution simulations providing training data
  - AI model training

- https://github.com/hewlettPackard/cmf
- https://github.com/CrayLabs/SmartSim
- "Framework for tracking metadata, lineage and model provenance in hybrid simulation-AI HPC exascale workflows", M. Foltin et al., ACM ICPS 2025 (submitted)





Full AI Replacement of Simulation

Where are we today?

- Full AI models have shown success
- ECMWF AIFS, NOAA HRRR-Cast, and Google WeatherNext are examples for weather forecasting
- Example: atmospheric model for weather simulation, whole earth, 30-km resolution, 137 levels, 6-hour prediction
  - ~8-80 Tflops (64-bit) for simulation
  - ~1.5 Gflops (lower precision) for FourCastNet

Where might we go?

- Investigation using sparse observations directly for inference
  - Avoid pre-processing burden
- Performance progression similar to surrogate models

What are the challenges?

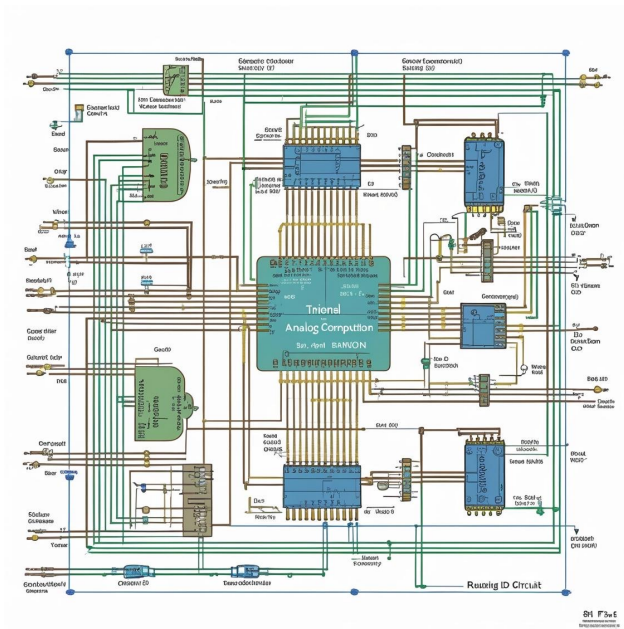
- Training (and re-training) potentially harder in some cases given the need for broader sets of input data than in individual models
- Moving further from the underlying science/physics, does that matter?
  - See previous comment about discretization
  - How to understand the error bounds?
- Accounting for performance has the same issues as with surrogate models

Other Technologies

- Quantum – big potential gains, but not broadly applicable across the algorithm and application space
  - When it becomes more practical, can address several important areas
  - Energy efficiency depends on technology
  - Integration still being investigated, especially at fine granularities

- Analog – potentially fast and accurate
  - Can completely avoid the discretization/approximation problem
  - Circuits are built to represent functions
    - Circuit design is often non-trivial and not the same as “programming”
  - Consideration of energy is different since there aren’t discrete “operations”
    - Circuits tend to operate continuously at a given power draw
    - Maybe be able to equate to replaced operation counts
  - Integration with digital computation still a challenge

- Others mentioned this week
  - Superconducting, neuromorphic, simulation on AI devices



Created via Canva.com

Radical Evaluations

- "Operation" count metrics don't reflect what is really being accomplished with computation
- So what would? Insights? What are those and how would we measure them?

- Insights are domain specific
  - A weather forecast
    – or specific part of one, e.g., component such as precipitation
  - Understanding a stellar explosion
  - Designing an airplane engine
    – or a specific part, e.g., turbine, compressor

- Insights have applicable ranges and accuracies
  - Weather forecasts
    – For a certain region at a specific granularity
    – Deterministic (short-term) versus probabilistic (long-term)
    – Provides different accuracies depending on lead time
  - Different stellar bodies and events have different characteristics

- Hard to normalize but ultimately isn't this what we are trying to achieve?
  - Then can consider insights per time and/or energy

Created via Canva.com

Software Challenges for Technology Integration

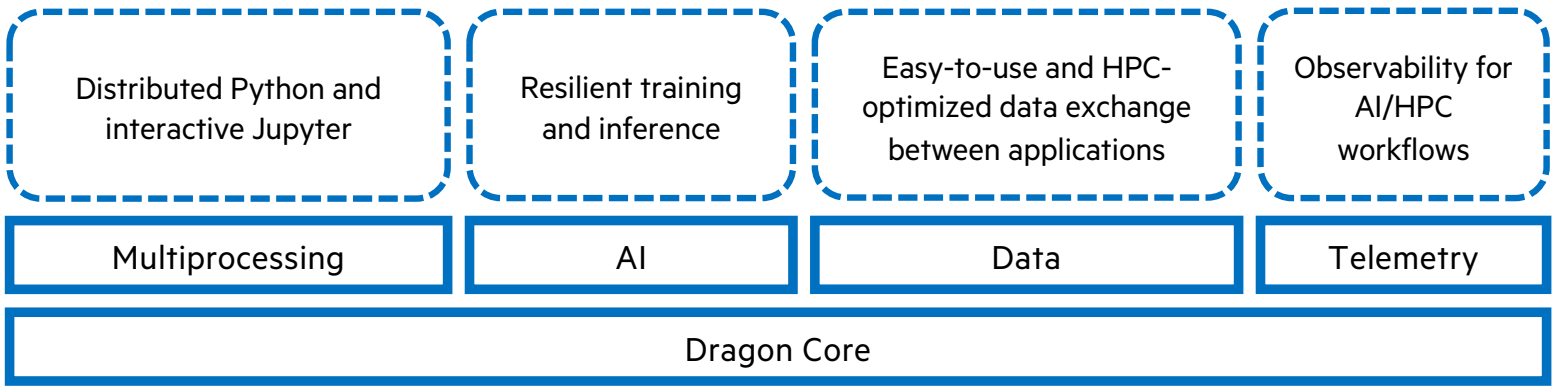
- How do we “glue together” simulation with AI or other technologies?
  - The tighter the coupling, the bigger the challenge
  - Python clearly in common use in the AI space, seems useful for some forms of coupling
- Many domain-specific workflow management engines exist but don’t necessarily help tight coupling

- Several federation and programmatic frameworks for HPC and AI exist in the community
  - NERSC Superfacility, ORNL Intersect, CSCS FireCrest
  - Each represents **necessary** APIs but none seem complete or **sufficient**
- Can we create a framework to support a union of these APIs?
  - Have vendors and “service” providers provision one fully-featured API definition
  - Provide a portable substrate across systems and vendors
- Then support mapping any of the existing APIs to the new one
- Users can choose to use their own APIs (and potentially extend them) or call the new API directly

- Overall requirement to better define and coordinate workflows on heterogeneous resources

Runtime for AI and HPC

Dragon is a composable distributed runtime that enables users to create sophisticated, scalable, resilient, and high-performance AI/HPC applications, workflows, and services through standard Python interfaces

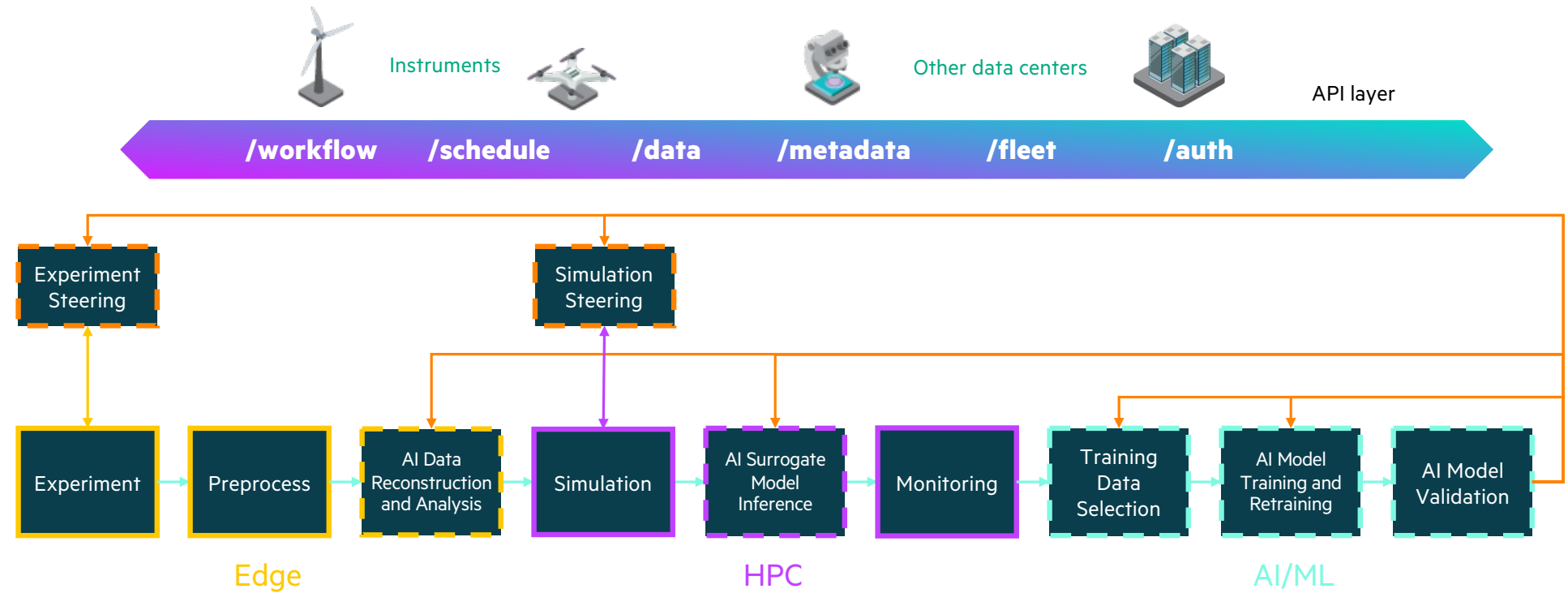


- 2 – 100X faster data processing than Ray
- Scalable to over 1000 nodes
- Multi-system features offer a hybrid experience, spanning from laptop to supercomputers
- Open-source or HPE-optimized packages
- Well-documented with numerous cookbook examples and easy setup

developer.hpe.com/platform/dragonhpc/home
github.com/DragonHPC/dragon

Common Federation Framework: Workflow Deployment SDK

Enables federated hybrid workflows on data from Edge-to-Supercomputer-to-Cloud



Conclusion

- Zettascale is here already in some limited forms
- Traditional 64-bit flop Zettascale is at least 15 years away, barring technological discontinuities
- Widespread “Zettascale” with a combination of techniques is much closer
- But accurate quantification and measurement needs more consideration

Software will play an ever-increasing role

- Traditionally:
  - Mapping scientific formulas to numerical algorithms
  - Compilers to convert user readable code for algorithm implementations to machine code
  - Advanced runtimes to support algorithm implementations
- Going forward:
  - Mixed and reduced precision along with careful studies of applicability
  - Integration of AI surrogates and esoteric technologies
    - Resilient runtimes and workflow/federation APIs
    - Common and domain specific software to drive these APIs
  - Software defined workflows and resources with easy reconfiguration
  - Pervasive use of AI assistants in coding and analysis of implementations

Thank you