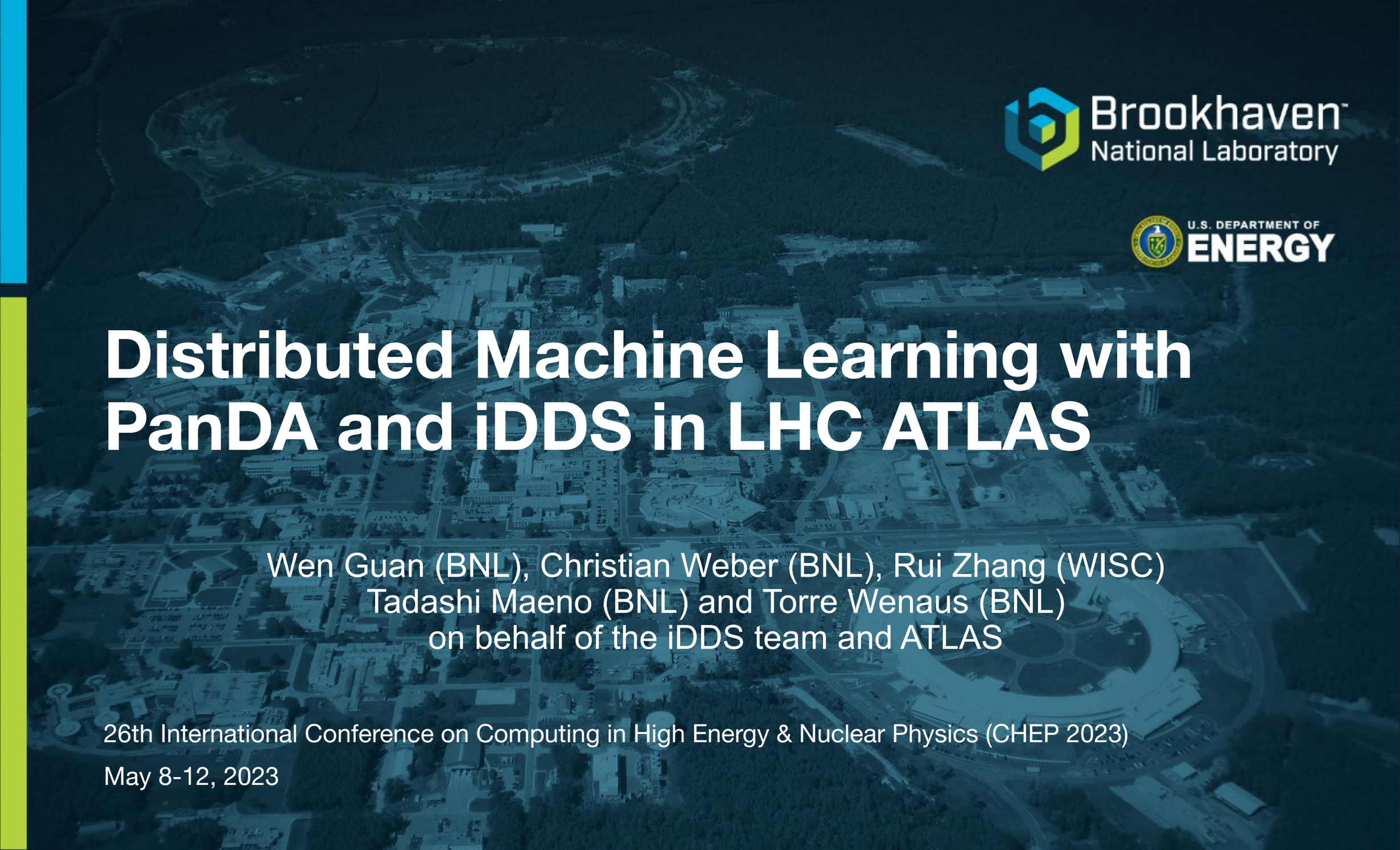


Distributed Machine Learning with PanDA and iDDS in LHC ATLAS

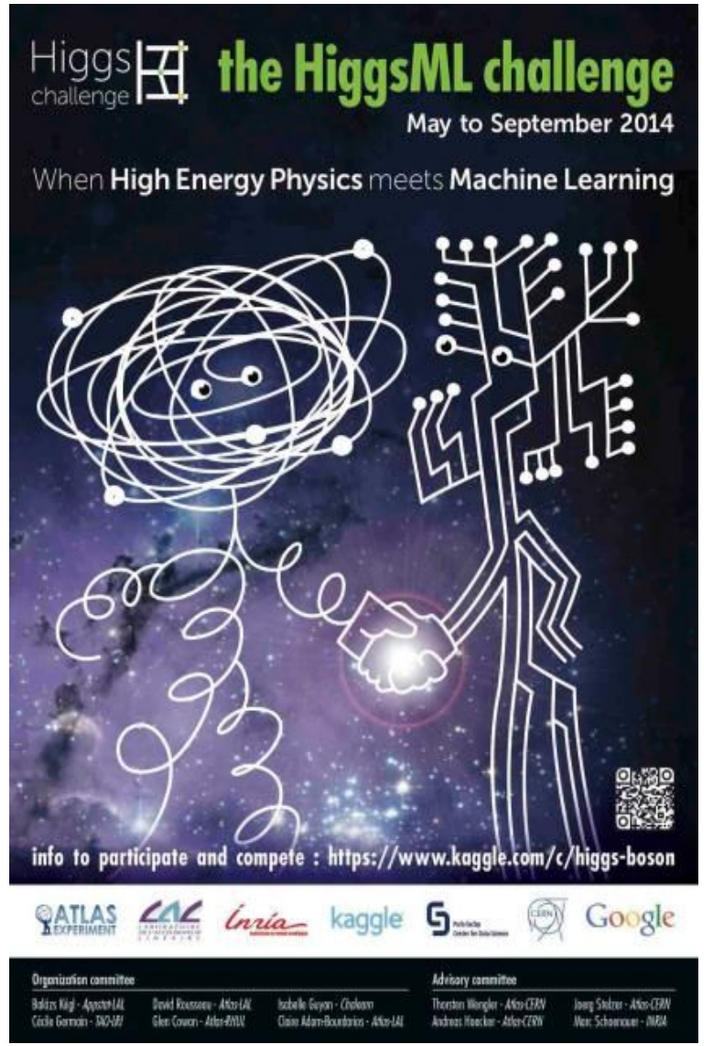
Wen Guan (BNL), Christian Weber (BNL), Rui Zhang (WISC)
Tadashi Maeno (BNL) and Torre Wenaus (BNL)
on behalf of the iDDS team and ATLAS

26th International Conference on Computing in High Energy & Nuclear Physics (CHEP 2023)

May 8-12, 2023

Distributed Machine Learning (ML) for ATLAS

- Motivations of Distributed Machine Learning
  - Access to multiple remote large scale resources
    - GPUs, HPCs, Clouds
  - Meeting large scale AI/ML requirements
    - Speed up the training and optimization by orders of magnitude
    - Make more complex and resource intensive AI/ML applications accessible
  - Support complex AI/ML workflows
    - A single complex workflow can present requirements for which a single resource is not optimal
- Goals of Distributed Machine Learning
  - Transparently schedule workload to distributed resources, scalable to various huge resource requirements
  - Orchestrate between tasks in a workflow, to automate the workflow
    - In traditional analyses, manual operations are required to analyze the results to submit new tasks when a task finishes



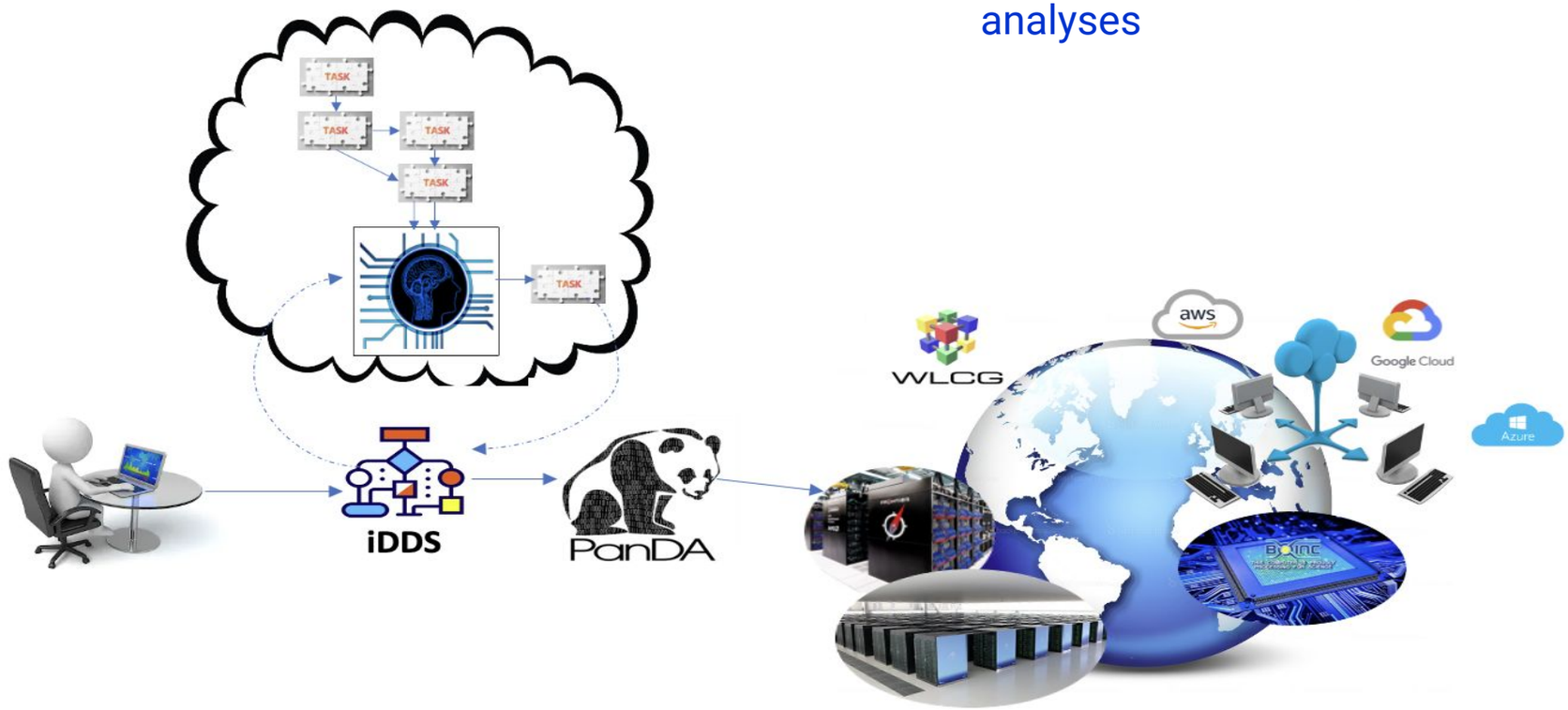
ATLAS' Higgs ML Challenge

Distributed ML with PanDA and iDDS in ATLAS

- PanDA as an engine for large scale AI/ML
  - PanDA is powerful to schedule jobs to distributed heterogeneous resources
  - Large scale
  - Transparent to users for different computing resources
  - Smart workload routing

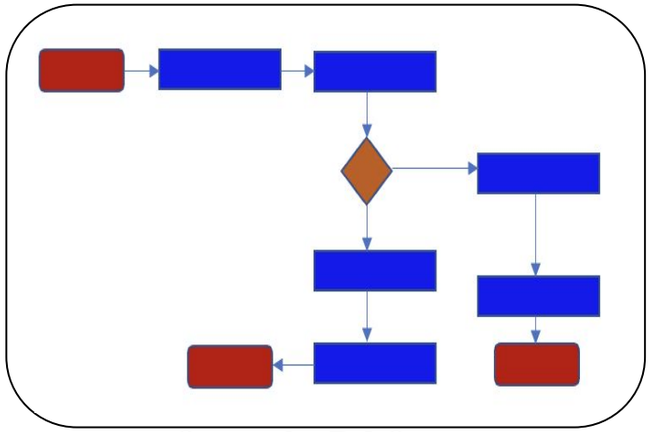
CHEP2023 Talk: T. Maeno, et al. Utilizing Distributed Heterogeneous Computing with PanDA in ATLAS

- iDDS (intelligent Data Delivery Service) orchestrates the workflow for automation
  - Generic workflow orchestration
    - Directed Acyclic Graph (DAG) management.
    - Condition workflow and Loop workflow management
  - Complex workflow orchestration
    - Collect results from previous tasks
    - Analyze the results with user predefined jobs
    - Generate new tasks/jobs based on the analyses

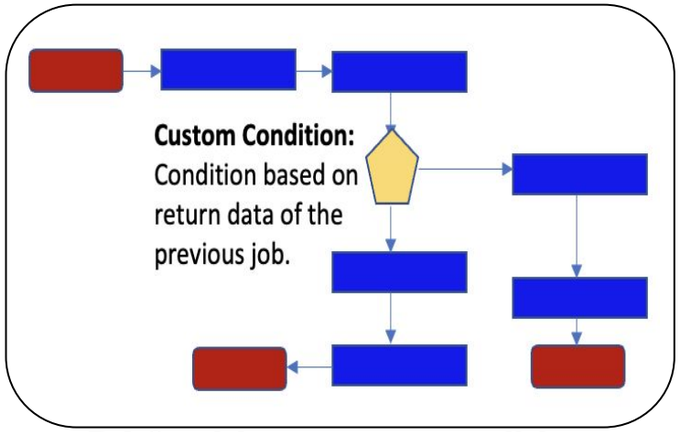


Workflow Orchestration

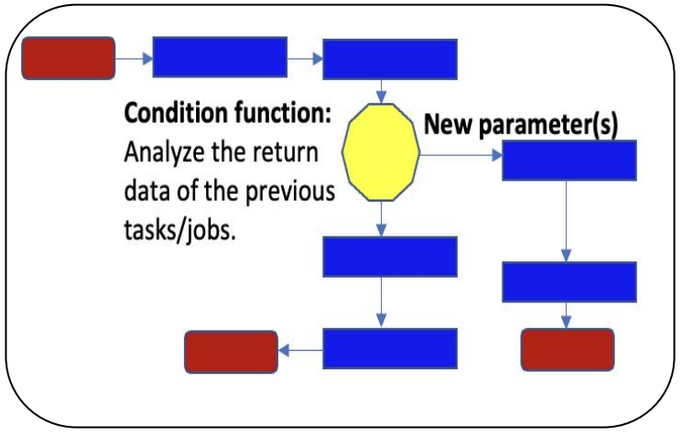
- iDDS orchestrates the workflow, to automate the task/job chain in the workflow
  - Generic workflow orchestration (DAG, Loop workflow, Condition workflow)
    - Mainly based on the status of the previous task/job, e.g. whether it's finished or failed
  - Complex workflow orchestration
    - Not only based on the status of the previous task/job, but also depends on the return data from the previous task/job ---> Custom condition
    - Even complicated cases, e.g. we may need an external job to analyze the return data from the previous tasks/jobs, to generate new conditions with/without new parameter(s) for new jobs ----> Condition function



Generic DAG



DAG with custom condition



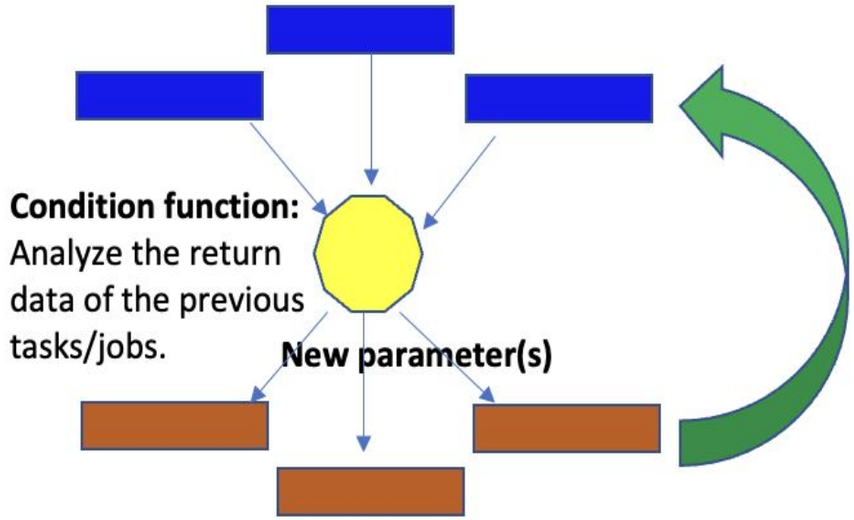
DAG with condition function

Complex Workflow Orchestration

- iDDS complex workflow orchestration requires two points
  - Return data from the previous tasks/jobs
  - Condition function can be customized
- Return data of previous tasks/jobs
  - iDDS provides APIs for a job to report the return data to iDDS, an example format is a pair of {parameter: result}
  - The condition function can also use some data management tools, such as Rucio, to download the files produced by previous jobs
- Condition function
  - An interface to be able to execute external jobs to analyze the return data from the previous jobs, to generate new conditions, and also with new parameters for next jobs
  - The external job can be some predefined methods, such as some predefined Bayesian methods
  - The external job can also be user-defined methods
  - The external job normally will be executed in an iDDS internal cluster pool (to optimize the turnaround time). It can also be scheduled to other resources

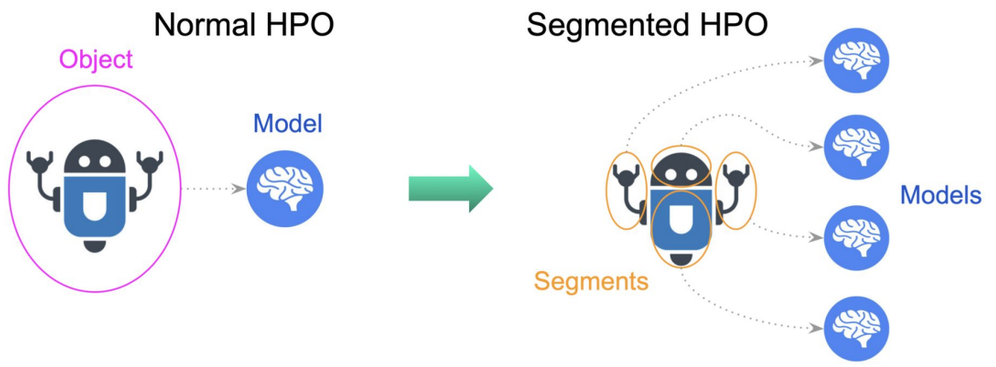
Workflow orchestration for Distributed ML

- Iterative regression structure for Distributed ML with condition function
  - Condition function
    - Multiple inputs from distributed jobs
    - Multiple new parameter outputs for a bunch of new jobs
    - Threshold to trigger the condition function, e.g. when a job finishes or 50% jobs finish
  - Iterative loop supports
- By customizing the condition function, different use cases can be supported
  - E.g. with the Bayesian Optimization method as a condition function to optimize hyperparameters for a ML task
- Distributed ML use cases in ATLAS
  - Distributed HyperParameter Optimization (HPO)
  - Monte Carlo Toy based Confidence Limits
  - Active Learning

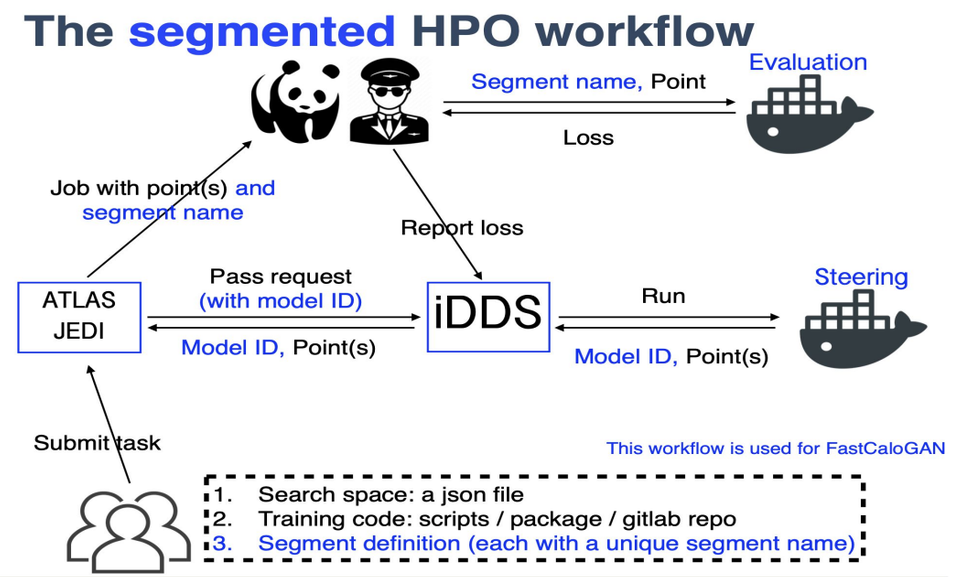


Distributed HyperParameter Optimization (HPO)

- Provide a full-automated platform for HPO on top of distributed heterogeneous computing resources
  - Hyperparameters are generated centrally in iDDS
  - PanDA schedules ML training jobs to distributed heterogeneous GPUs to evaluate the performance of the hyperparameter
  - iDDS orchestrates to collects the results and search new hyperparameters based on the previous results
- Applied for ATLAS FastCaloGAN
  - The HPO service is in production for FastCaloGAN, part of the production ATLAS fast simulation AtlFast3
  - With hyperparameters to tune various models targeting different particles and η slices
  - Distributed GPUs, HPCs, commercial cloud
  - Ref: FastCaloGAN, AML workshop, IML, ATLAS S&C week
- Used in ATLAS, however not specific to ATLAS



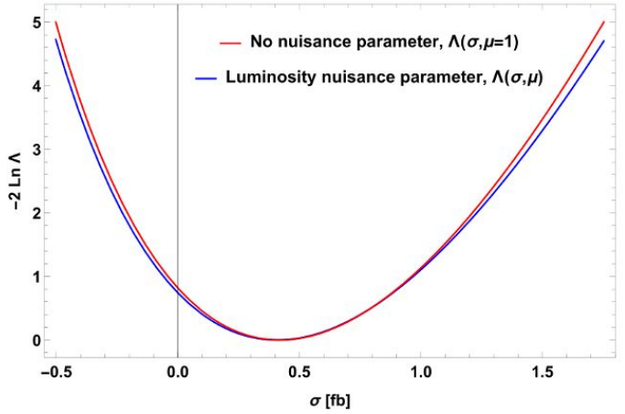
R. Zhang 5th ATLAS Machine Learning Workshop

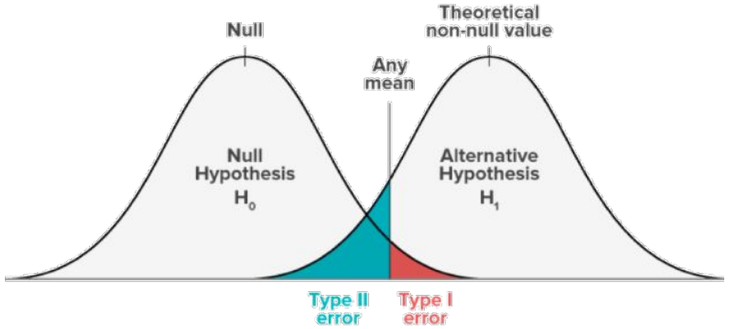


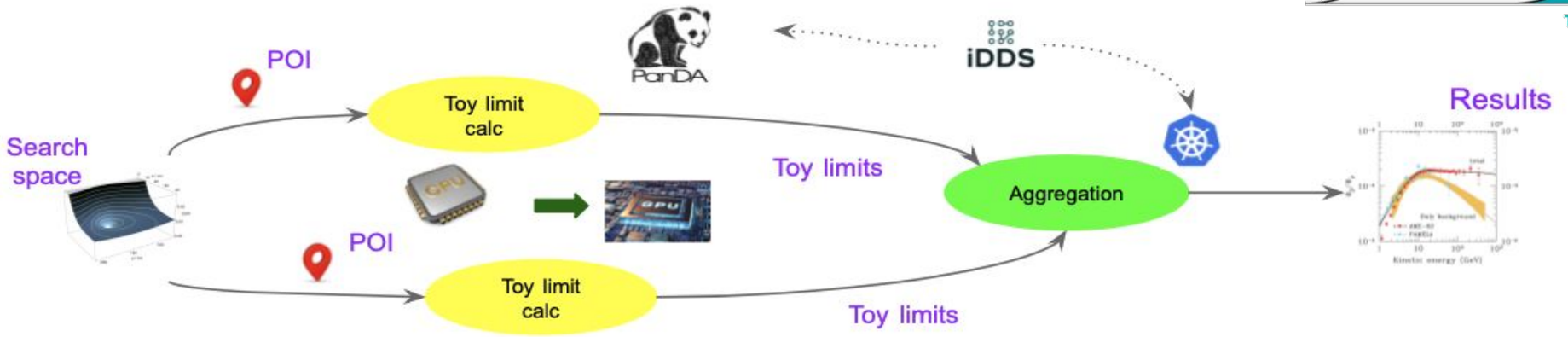
R. Zhang FastCaloSim+DnnCaloSim

Monte Carlo Toy based Confidence Limits

- Confidence Limits in Analyses
  - Exclude some ranges of relevant phase space for future processing
  - Show that obtained results are meaningfully different from what could have obtained by chance
- An Monte Carlo (MC) Toy based confidence limits workflow requires multiple steps of grid scans, where the current step depends on the previous steps
- Automate the workflow of Toy limits calculation and aggregation
  - Point of Interest (POI) generation based on the search space and results aggregation to generate new POIs in iDDS
  - Distributed Toy limits calculation to distributed resources with PanDA

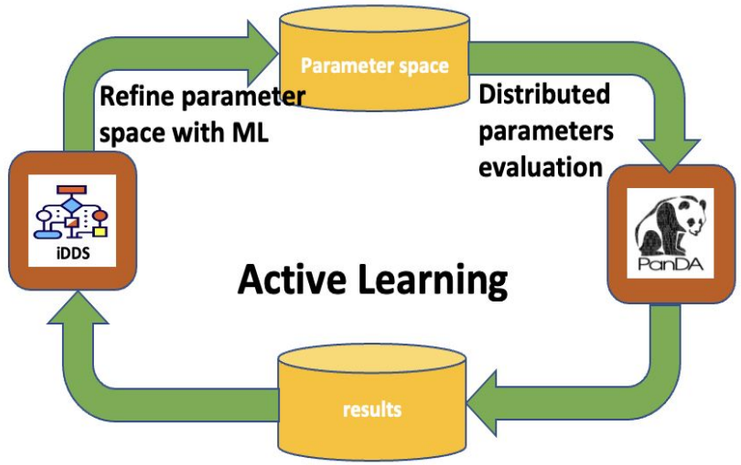


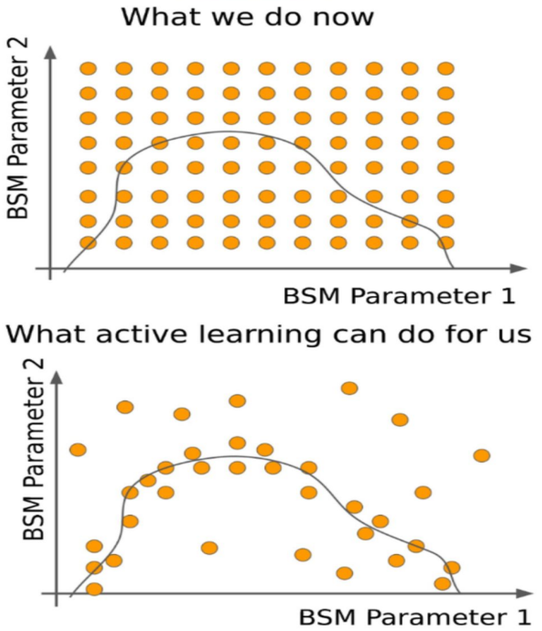




Active Learning

- An iterative ML assisted technique to boost the parameter search in New Physics search space
  - The Active Learning technique we are applying was developed by Kyle Cranmer et al, “Active Learning for Excursion Set Estimation”, ACAT 2019
  - Redefine the parameter space for the next iteration based on the previous results with ML, more efficient than a single-step processing
  - Optimize the parameter space points for evaluation to maximise the information gain from each evaluation
  - Distributed computing resources for parameter evaluation
- Automate the multi-steps processing chain with PanDA and iDDS for ATLAS
  - Integrated REANA (Reusable Analyses) with PanDA for learning processing
  - iDDS orchestrates the workflow to trigger new tasks/jobs based on the previous results

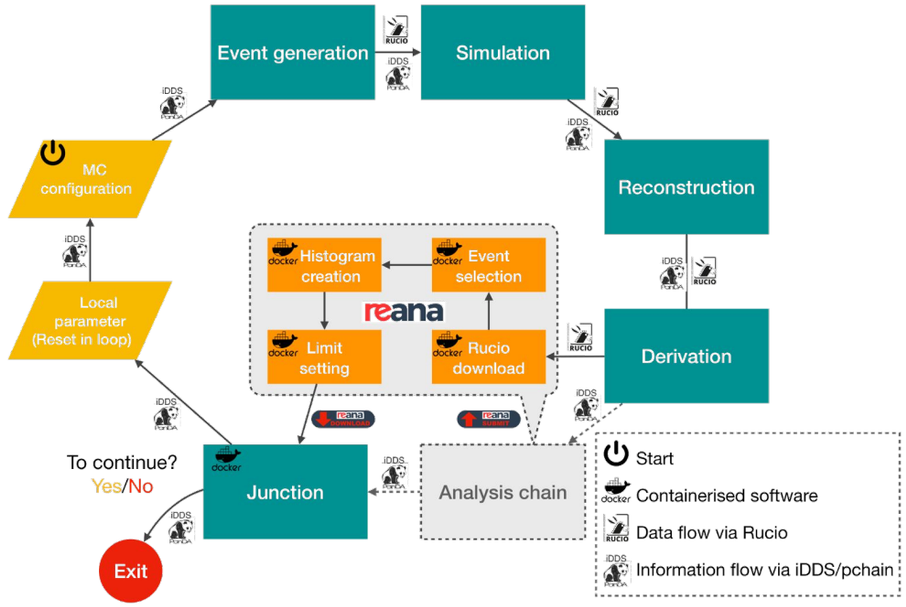


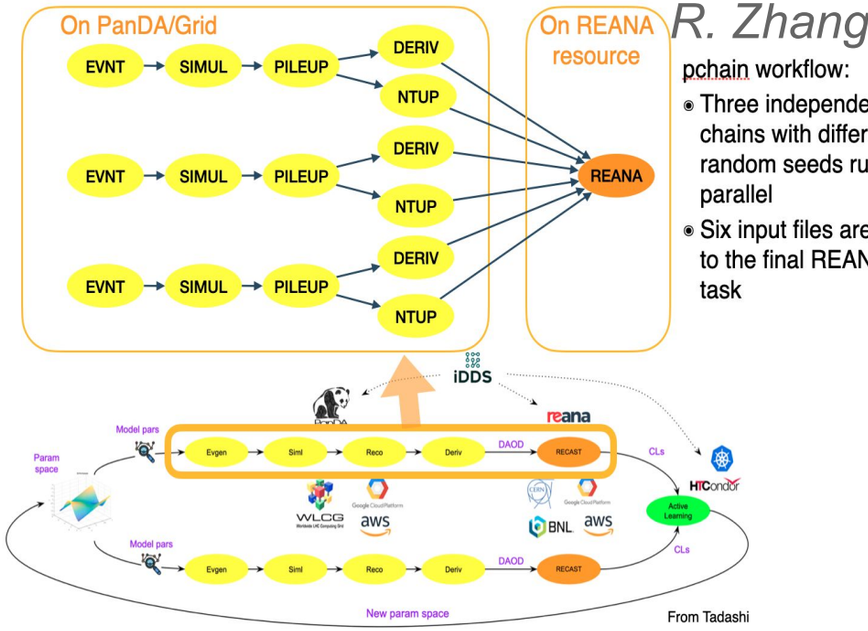


Active Learning via iterative regression on a limit surface

Active Learning for ATLAS

- Applied the Active Learning service in the H $\rightarrow ZZ_d \rightarrow 4\ell$ dark sector analysis
  - Avoids a complex interpolation scheme, costly in development and validation
  - Apply Bayesian Optimization to refine the parameter space
  - Greater efficiency, scalability, automation enables a wider parameter search (instead of 1D, 2D or even 4D on large scale resources) and improved physics result
  - Has demonstrated active learning driven re-analysis for dark sector analysis
  - ATLAS PUB NOTE in progress





R. Zhang

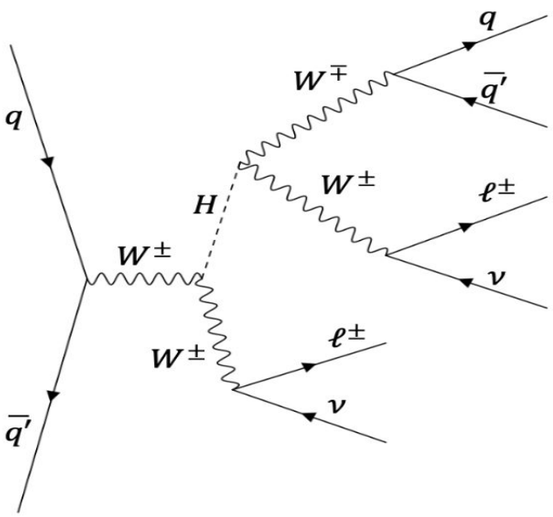
pchain workflow:

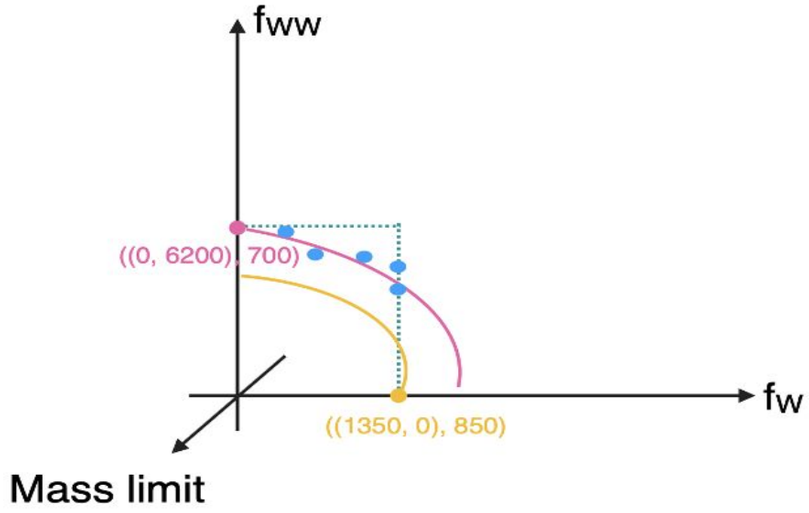
- Three independent chains with different random seeds run in parallel
- Six input files are fed to the final REANA task

CHEP2023 Talk: C. Waber, et al. An Active Learning application in a dark matter search with ATLAS PanDA and iDDS

Active Learning for ATLAS

- Applying for generic Heavy Higgs → WW search
  - It's important to understand the characteristics of the mH-fw-fww space
  - It's too expensive to sample the 3D space points with the traditional way
  - Active Learning may be possible
    - Draw a contour in fw-fww plain for a given mH
    - Sample the 3D space points
  - Under R&D at the moment





Conclusion and future plans

- Distributed Machine Learning workflow with PanDA and iDDS
  - PanDA as an engine for large scale AI/ML. It utilizes distributed heterogeneous computing resources to support user workflows
  - iDDS orchestrates the workflow. It automates the chain in the workflow
  - An integrated service for Distributed Machine Learning
  - Different use cases applied in ATLAS, however the work is not specific to ATLAS
- Future plans
  - Generalize the services as a contribution to the HEP AI/ML ecosystems
  - Apply new technologies, such as Platform-as-a-Service (PaaS) and Function-as-a-Service (Faas), to enhance the user ability and experience
  - Improve user interfaces with Jupyter.