ESTF2020

Towards a Scalable Parallel Bayesian Observing System Simulation Framework

Presenter and PI: Derek J. Posselt[1]

Team Members: Brian Wilson[1], Rachel Storer[2], Noppasin Niamsuwan[1], George Duffy[1], Matt Lebsock[1], Berlin Chen[1], Benyang Tang[1], Simone Tanelli[1]

Program: AIST-18

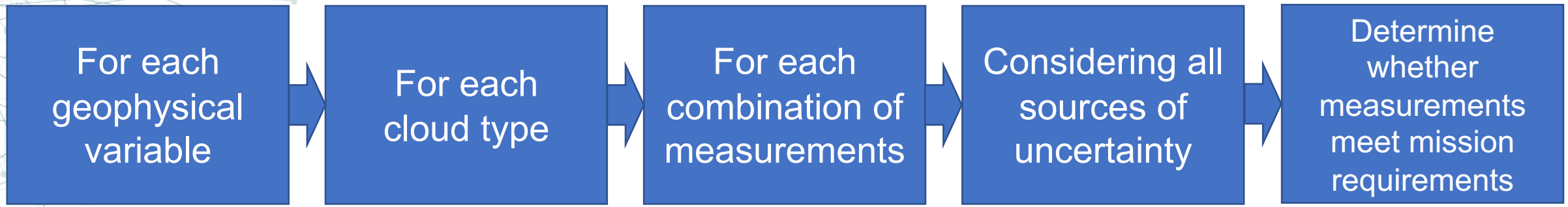
[1]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA
[2]University of California, Los Angeles, CA

Scientific Challenge

- Clouds and precipitation are central to climate and weather
- After decades of space-borne measurements, *key processes are still missing*
- Goal: design a new observing system (e.g. ACCP)
  - Address specific science objectives
  - Consider the vast range of possible measurements
  - Rigorously quantify uncertainties

Technical Challenge

- The design trade-space is *large* and clouds are *diverse*
- The dimensionality of the mission design problem is *immense*
  - Multiple different geophysical scenarios (different cloud types)
  - Diversity of measurement types (active, passive, single-point, distributed)
  - Multiple sources of uncertainty (instrument noise, forward models)
- **Computational challenge: identify suitable candidates**



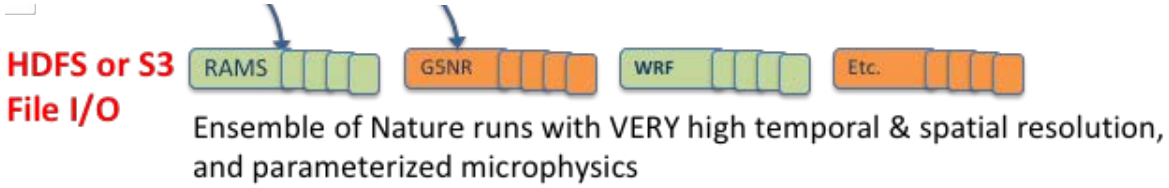
Solution:
Simulators + Bayesian Retrievals + Parallel Map-Reduce

- **Solution: Combine measurement simulators and Bayesian retrieval with a Parallel Map-Reduce framework ➔ Pervasive parallel computing**
- Library of nature runs – simulations of atmospheric phenomena
- Containers for Pluggable measurement simulators
- Bayesian retrieval algorithms for quantitative uncertainty estimates
- Flexible knowledge database
  - Search for and group experiment outputs by tags & run metadata
  - Fast ensemble statistics, comparisons, drill-down
- Map-Reduce framework and cluster/GPU computing to:
  - Generate large database of simulations – geophysical variable (retrieval) pairs
  - Compute analytics to determine whether measurements satisfy mission requirements

Parallel OSSE System

- Library of nature runs
- Containers for pluggable measurement simulators
- Bayesian retrieval algorithms to quantify information gain
- Flexible knowledge database: tags and metadata to search and group experiment outputs
- Map-Reduce framework and cluster/GPU computing to:
  - Generate large database of simulations – geophysical variable (retrieval) pairs
  - Compute analytics to determine whether measurements satisfy mission requirements

Architecture of the Parallel OSSE Framework



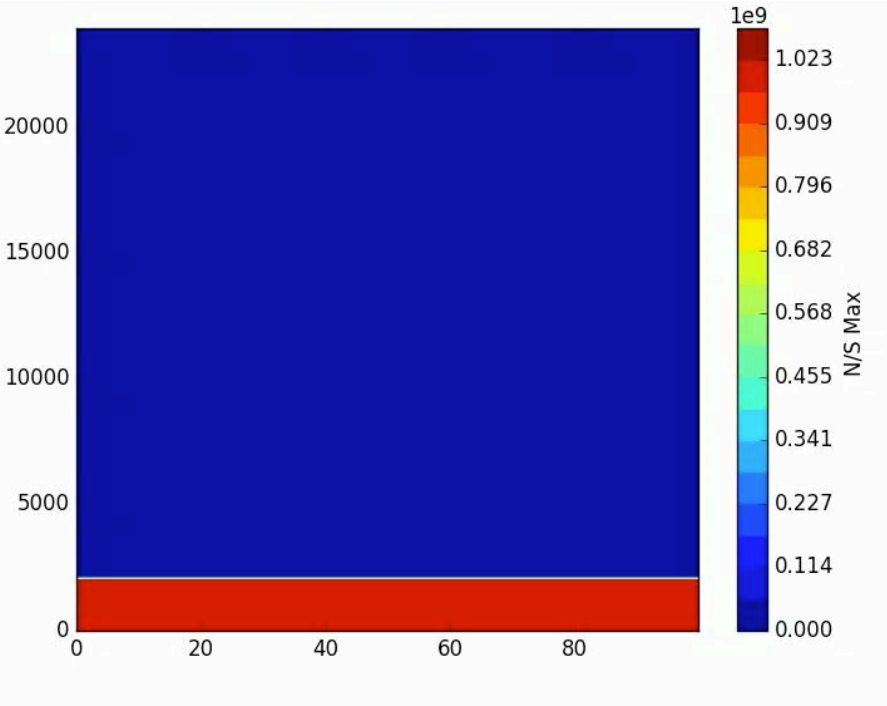
Deploy using on-premise hardware cluster AND at Amazon.

Details of OSSE Infrastructure

- **Knowledge Base**
  - ElasticSearch JSON doc. database, on-premise or at Amazon
  - Kibana dashboard to keep track of experiments in-flight or completed
  - Traceability for all experiment runs by configuration
  - Metadata: user tags, timestamps, code versions, input configs.
  - Sub-docs. for details of ensemble, instrument, retrieval parameters
- **Workflows and Notebooks**
  - Exploratory statistics, visualizations, and code development in Jupyter Notebooks
  - Production workflows in version-controlled Python scripts
- **Pluggable Codes**
  - Binary executables called from Python (“wrapped”): QuickBeam, H&B
  - Thin Python clients calling into Docker container: NEOS3
- **Deployment on parallel backends**
  - Codes written once using PARMAP Python library
  - Deploy by changing config. “string” to Spark/Dask cluster or Lambdas

Example: Radar Observations of Convection

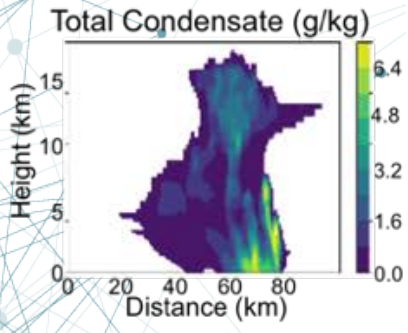
- Nature run consists of library of highly realistic simulations of convection
- Simulate radar observations
- Implement a Bayesian retrieval
- Two use cases:
  - Effect of changes in ice microphysics assumptions
  - Variability in radar design parameters (footprint, vertical sampling)

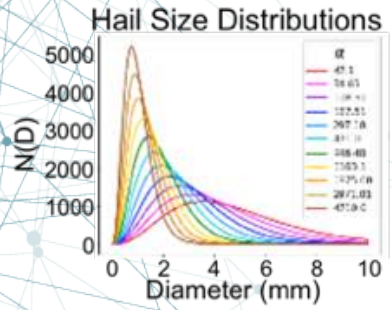


Cross-section through convection showing transport of pollution from the boundary layer in convection

Example: Radar Observations of Convection

User Input





- Produce a large ensemble of radar profiles with quantified uncertainty
- Analyze statistics to assess measurements vs requirements

Summary and Next Steps

- Progress:
  - Project began Jan 2020
  - Have already constructed a skeleton system, and are testing on relevant cloud retrieval problems (low-cloud objective for ACCP)
    - Jupyter hub
    - Python-based workflows
    - JSON configuration documents and Elasticsearch database
    - Bayesian (optimal estimation) radar-based retrieval of cloud properties
- Application to future missions:
  - ACCP low clouds, deep convection, and snowfall
  - Assessment of potential future distributed architectures (ESTO NOS) – convection and snowfall

Back Up Slides

Further details on knowledge base

- Progress
  - Deployed Elasticsearch (search/analytic back end) and Kibana (application front end) on on-premise cluster
  - Developed lightweight Python handler for ingesting experiment metadata into Elasticsearch
  - Implemented Python wrappers for radar simulator (QuickBeam) and cloud water path retrieval, ingesting metadata (such as microphysics param.) into Elasticsearch during runtime
- Next step:
  - Extend Python handler: add querying functionality
  - Explore possibility of transitioning to AWS Elasticsearch Service

Further details on end-to-end retrieval experiment for shallow clouds

- Components implemented:
  - Python handler for running forward executable (QuickBeam)
  - Python code for optimal estimation retrieval of cloud water path
  - Wrappers (also in Python) allowing 1. and 2. to be fully configurable in JSON; the wrappers upload all relevant metadata (such as path of output h5) to Elasticsearch
  - MAP function transforming experiment trade space to a list of embarrassingly-parallel run configurations
  - PARMAP library providing parallelization capability
- Next step:
  - Analytics capability: query Elasticsearch, retrieve data, and generate statistics/visualizations
  - Finalize I/O format and put together a top-level function/object
  - Conduct experiments

Project Objective, Technology, & Science Goals

- **Objective**: Construct a software architecture capable of rapidly and thoroughly evaluating mission architectures
- **Technology**: Pluggable instrument simulators connected to Spark MAP-REDUCE analytics, Jupyter notebook workflows, and an ElasticSearch database
- **Science Goals**: Evaluate measurements of hydrometeors and dynamics in convection from diverse platforms