USGIF
geoint
2021
SYMPOSIUM
Discovery and Connections
October 5-8
St. Louis, Missouri

Conducting GEOINT Intelligence at Scale!

Dr. William Kramer, Dr. Greg Bauer, and Dr. Aaron Saxton
University of Illinois

http://nfi.illinois.edu/GEOINT

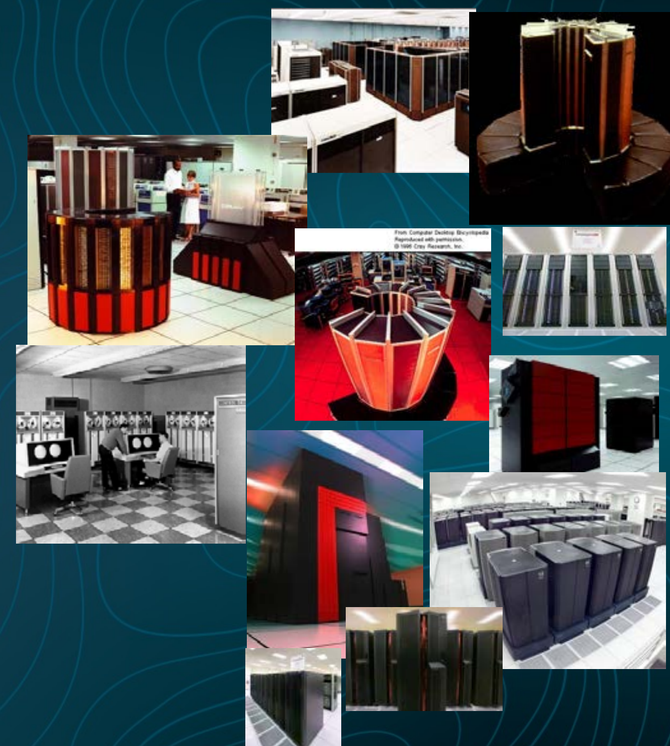
USGIF geoint SYMPOSIUM 2021

Outline

1. About GEOINT at Scale – Bill Kramer (20)
2. HPC and GEOINT Use Cases – Greg Bauer (15)
3. Machine Learning at Scale – Aaron Saxton (15)

Various Definitions of a Supercomputer

- A large and very fast computer
  - http://www.merriam-webster.com/dictionary/supercomputer
- A supercomputer is a computer that performs at or near the currently highest operational rate for computers. A supercomputer is typically used for scientific and engineering applications that must handle very large databases or do a great amount of computation (or both).
  - http://whatis.techtarget.com/definition/supercomputer
- A supercomputer is a computer at the frontline of contemporary processing capacity – which can happen at trillions of floating point operations per second.
  - http://en.wikipedia.org/wiki/Supercomputer
- "big, dumb and simple" – attributed to S Cray by a colleague
  - "dumb and simple" could be said of RISC processors created in the 1980's
- "Anyone can build a fast CPU. The trick is to build a fast system." – attributed to S Cray on the Cray Inc. web site
- ……

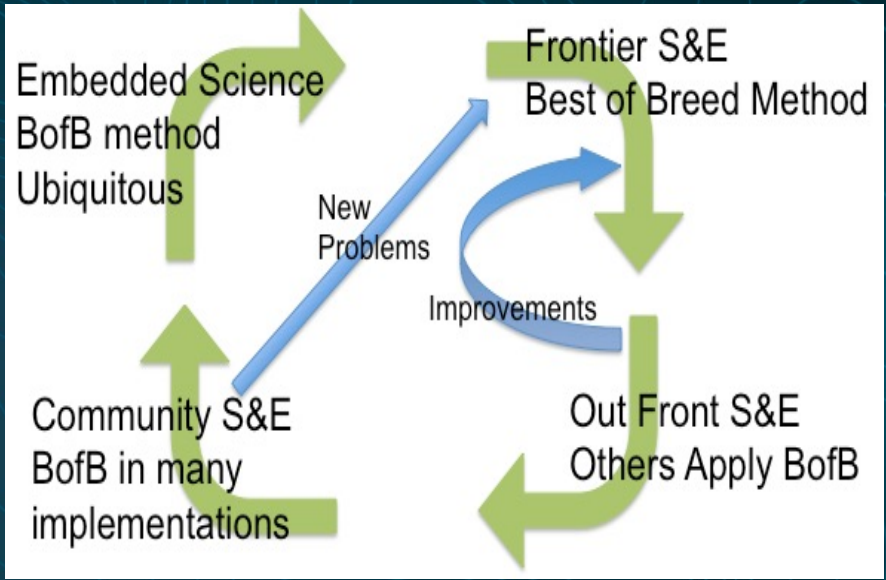


USGIF geoint 2021 SYMPOSIUM

2018

Enbleing Frontier Science and Engineering

- It often takes tremendous computing power to develop new ways to solve the most challenging problems
- Very specialized approaches are needed
- Improving the algorithms (methods of solving problems) decrease the time it takes to solve a problem at least as much as new hardware.
- What is done on a high-end systems typically becomes common practices a decade later on other systems, and is used for many standard things within another decade
- Leadership mission is to make teams addressing Frontier Science highly effective and productive as they solve some of the world's most challenging problems.

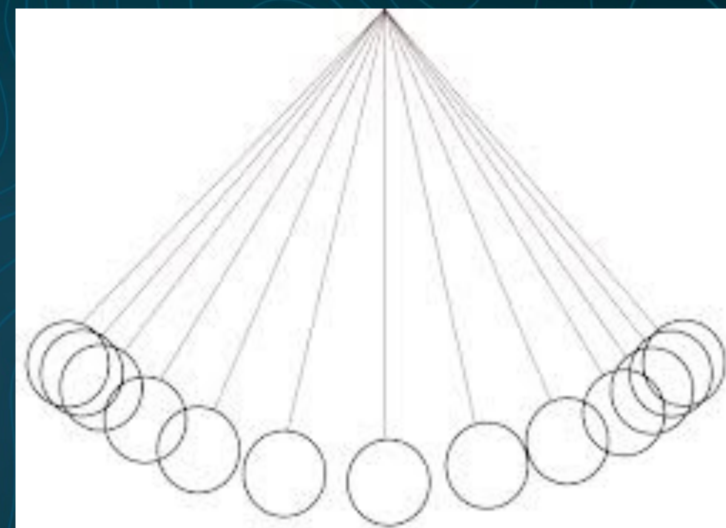


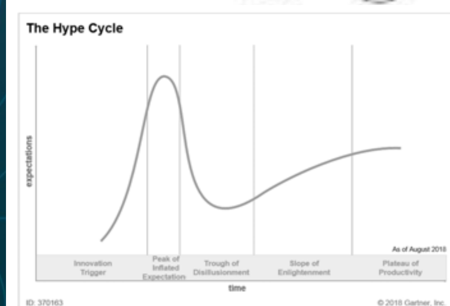
How HPC Increases Productivity

- Helps improve time to insights
  - If your problem is taking a really long time to get a result, or if you do not attempt to do something because you know it would take too long, HPC may be your solution
- Helps increase the fidelity of application (resolution, timesteps, number of particles, amount of data)
- Makes creating new methods feasible in a tractable amount of time
  - Ironfist, Adaptive Mesh Refinement (AMR), ML/AI.
- Provides people a robust/performant/balanced infrastructure
- Cost-effective for large amounts of computation and data and/or for new methods grand challenge method developments
- Deadline based production

“It’s like déjà vu all over again” – Yogi Berra

- “Cloud” – means many things
  - Technologies,
  - Business models
  - Software Methods
- The pendulum swings back and forth
  - Individual in-house systems vs outsourced systems
    - E.g. Tymshare,
  - Generations – single processors, multiple processors in SMPs, “vaxination of computing”, attack of killer micros, distributed systems with low and high latency interconnections, ….
  - Capability vs high throughput
  - SW models – proprietary SW, Unix based, Linux based, cloud featured
- Sometimes early use is specialized and then becomes common
  - Array processors <-> GPUs,
  - Specialized attached processors <→ FPGAs



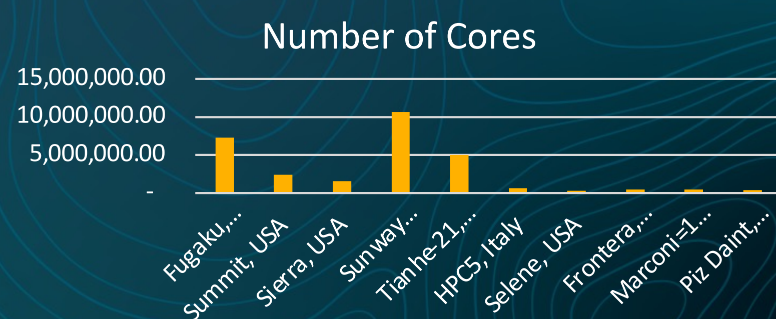


Source: Gartner (August 2018)

Processors/Accelerator Evolution

- Until mid 1990s individual CPUs got faster. Some systems had multiple CPUs starting from the mid 1970s
- From mid 1990s to mid 2000s, the number of cores per node increased. Custom CPUs were replaced by commodity CPUs
- Starting in mid 2000's hybrid (accelerated) systems became more common but proved difficult to be the "rising tide" for all applications
- Non-hybrid systems are still very common and productive

10 highest performing listed
System for the Nov 2020 HPC Linpack Benchmark



USGIF geoint SYMPOSIUM 2021

The Characteristics of a Supercomputer

- A system that can bring to bear the entire capability on one problem for Frontier and/or Best of Breed Research and Engineering and/or for critical time to solution
  - A system that is very efficient at parallel computing
  - Supports a variety of methods and investigations
  - Includes hardware and software in an integrated manner
- Features
  - Large amount of computational power
  - Very large amount of I/O and data capability
  - Very high bandwidth and low latency for memory and interconnect
  - Interconnect that allows all components to be applied to a single problem or many problems
- Non attributes of supercomputers/HPC
  - Only special things can use a supercomputer
  - Expensive
  - Difficult to use

The Characteristics of a Supercomputer
(cont)

- Balanced to support multiple needs
  - A system whose memory can feed all the CPUs in a node and all the nodes in the system
  - As system where I/O, storage and networking are never the bottlenecks
- Consistent performance
- A system that can support a wide gamut of research fields and computing styles
  - Being able do a number of different problems
    - E.g. MDgrape is not a supercomputer
  - Does not have to be most efficient/cost efficient at each of them
    - To quote a National Academy study, one "can run a capacity problems on a capability system, but cannot run a capability problem on a capacity system"
      - So, there is no added cost to run small scale/single scale problems on large scale system if the prioritization is correct.
      - https://www.nap.edu/catalog/21886/future-directions-for-nsf-advanced-computing-infrastructure-to-support-us-science-and-engineering-in-2017-2020
- Application software that is accessible for automated optimization

Characteristics for Future @Scale Work

- Dramatically increase fidelity in models and simulations to improve insights and address new problems.
  - Increasing use of multi-scale and multi-physics. These are needed to accurately explore simulated phenomena.
  - Increasing resolution.
  - Increasing complexity.
  - Increased number of “ensemble” trials.
- Longer simulated time periods
  - often required to accurately simulate the system of interest
  - simulations of larger systems often require longer periods of time to stabilize
- Increased number of problems to address
  - The first 100 million all-atom simulations were completed in 2013. By 2020 there will be tens to hundreds of teams doing hundreds to thousands of 100 million atom simulations

Characteristics for Future @Scale

- Changing workflow methods
  - Deadline driven analysis for experimental and observational data
  - visualization to interpret and understand the simulation and analysis results
  - Malleable/elastic resource management for application load balancing and resiliency.
  - Automation through workflows to support repeatability of computational/analytical solutions.
  - Use of data model programming methods,
- Increased integration with data sources and increased use of simulation data products.
  - data from multiple experiments and observations
  - Observation data assimilation
  - Track-1 systems enables them to produce community data sets that are then useful for others
- Changing algorithmic methods
  - Substantially improve their algorithmic methods to reach new research goals over the next five to ten year
  - Not just to address new computer architectures
  - Also to improve the time to solution independent of hardware changes and to develop the algorithms needed for multi-physics and multi-scale simulations.
  - Use of adaptive gridding and malleable/elastic resource management
  - Applications load balancing and resiliency will expand. Improving load balancing is critical to overcoming both Amdahl's law limits and the increasing variation in system component performance
  - Need resources to re-engineer, test and validate

Converged HPC Facilities and Systems

- HPC Systems are not designed to to support @Scale computation, “Big Data” and AI/ML
- Changing workflow methods
  - Deadline driven analysis for experimental and observational data
  - Visualization to interpret and understand the simulation and analysis results
  - Malleable/elastic resource management for application load balancing and resiliency.
  - Automation through workflows to support repeatability of computational/analytical solutions.
  - Use of data model programming methods,
- Increased integration with data sources and increased use of simulation data products.
  - Data from multiple experiments and observations
  - Observation data assimilation
  - Track-1 systems enables them to produce community data sets that are then useful for others
- Changing algorithmic methods
  - Substantially improve their algorithmic methods to reach new research goals over the next five to ten year
  - Not just to address new computer architectures
  - Also to improve the time to solution independent of hardware changes and to develop the algorithms needed for multi-physics and multi-scale simulations.
  - Use of adaptive gridding and malleable/elastic resource management
  - Applications load balancing and resiliency will expand. Improving load balancing is critical to overcoming both Amdahl’s law limits and the increasing variation in system component performance
  - Need resources to re-engineer, test and validate

Convergence – how does convergence integrate with new technologies

- Physically, the aggregate hardware requirements for highly parallel modeling and simulation workloads and workloads that are small and/or non-parallel but high intensity are similar.
- To provide a complete, productive (e.g. minimize overall time to insight) system
  - Many investigations require a mixture of workflows
- Increasingly, teams are coming to HPC systems for a combination of capabilities
  - Examples - I/O infrastructure, node and processor types and software availability.
- Examples
  - Arctic and Antarctic Digital Elevation Mapping – acquire and transfer millions of satellite images, process with single node runs for 24 to 36 hours per image pair, deposit results into a public repository
    - Jobs and data availability controlled remotely
    - Using over 300,000,000 Core*hours per year
  - Other examples - Genomic Workflows, Earthquake Analysis, HENP workflows, Astronomy workflows, DL workflows, …

USGIF geoint SYMPOSIUM 2021

Limited Cost Comparison of AWS vs BW

- General comparisons are challenging
  - Performance differs on hardware and types of use
  - Cloud prices change and different providers have different business models
  - Comparison of services is not equivalent
    - Data Movement (ingest, egress, internal amounts of I/O, …)
    - Large Storage Capacity
    - Support services
- The analysis is a specific snap-shot in time and a limited set of systems
- Consistent with other broader studies (e.g. DOE Magellan Project Report - https://www.osti.gov/servlets/purl/1076794)

Example Cloud Cost Analysis (Public Pricing)

- Use the DEM generation application *setsm* with 2-meter resolution as performance “standard candle”.
- Comparison of dual-socket 32 core AMD 6276 Interlagos node to dual-socket 256 core AMD 7742 Rome node showed ~4x run time improvement for the Rome node.
- Using 22,638 Blue Waters CPU XE nodes (~725,000 cores) is the equivalent of 5,600 cloud provider c5a.24xlarge (96 VCPU) nodes for *setsm*.
- “dedicated-class” with 1-year upfront pricing discount with cloud computing and storage costs $106,612,800
  - Limited Services – e.g. no shared parallel filesystem
  - No ingress or egress charges included in cost
  - No support services
    - Cloud provider Support (Business level) $2,892,198 for the year.
- One year for Blue Waters ~$17,000,000
  - Includes all data movement, ~30PB of storage, 4,228 XK GPU nodes, expert support and assistance, training, etc

Blue Waters was ~6.3 times or more cost effective for just the equivalent computational capability http://nfi.illinois.edu/GEOINT

HPC in GEOINT

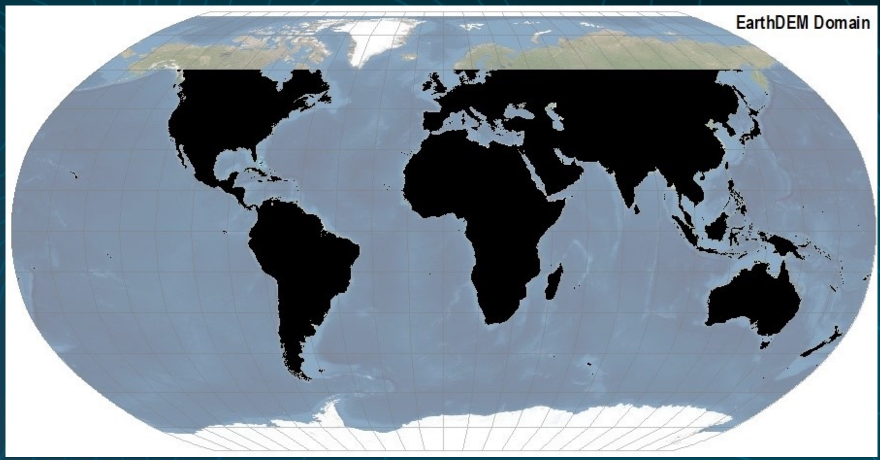
Each of the following use cases highlights GEOINT use requirements and how those requirements were addressed in an HPC environment.

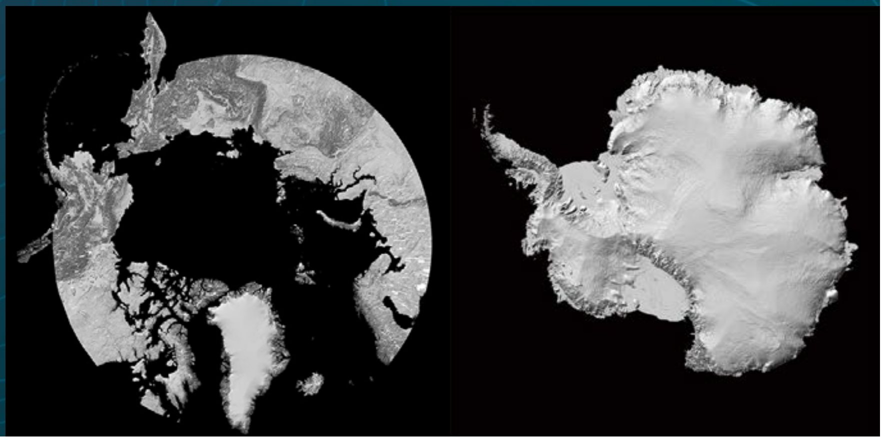
1. DEM generation – large scale image processing
2. IRONFIST – flood inundation as a graph problem
3. Earth Gravity Model – numerical inversion
4. NASA Tree Counting – large scale ML image processing

USGIF geoint 2021 SYMPOSIUM

Digital Elevation Models

- **DEM collaboration with UMN PGC (Paul Morin), OSU (Ian Howat) and NGA (Cathleen Williamson)**
- **Generation of 2-meter surface resolution DEMs from 50 cm commercial satellite stereoscopic stripe imagery using** Surface Extraction from TIN-based Searchspace Minimization (**SETSM) code.**
- **Historical DEMs for change detection.**
- **Polar regions completed as ArcticDEM and REMA.**
- **New EarthDEM in August https://www.pgc.umn.edu/data/earthdem**





HPC Challenges

- **Bundling and tracking jobs 100,000s of jobs**
  - **Use of the** *swift* **workflow software**
- **Petabytes of data transfer**
  - **Performant routes.**
  - **Capable GlobusOnline transfer hosts.**
- **Code Considerations**
  - **Performance Optimizations**
  - **Memory footprint**
  - **CI/CD with git and jenkins**

USGIF geoint SYMPOSIUM 2021

Benefits to GEOINT from HPC

- Routinely used 640,000 cores (20,000) compute nodes.
- Able to process  of imagery in a weekend at 2 meter.
- Sustained disk space footprint of 2 PB as new imagery came in and processed imagery was transfered.

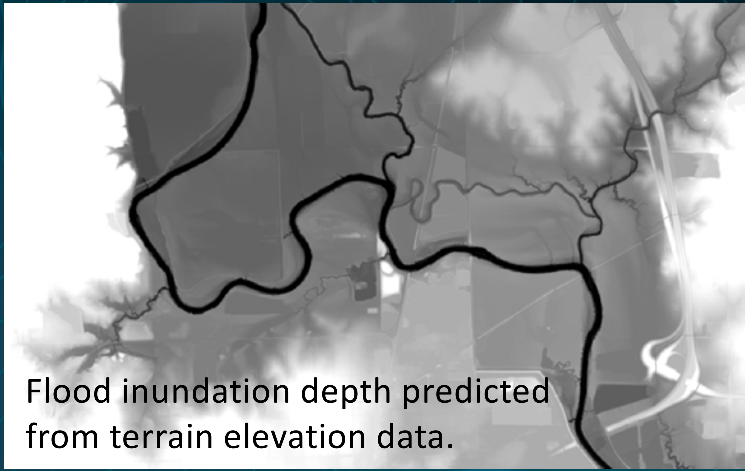
Fields near Petropavlovsk-Kamchatsky

DEMs made from DigitalGlobe/Maxar imagery

Padjelanta National Park, Sweden

IRONFIST Flood Inundation Mapping

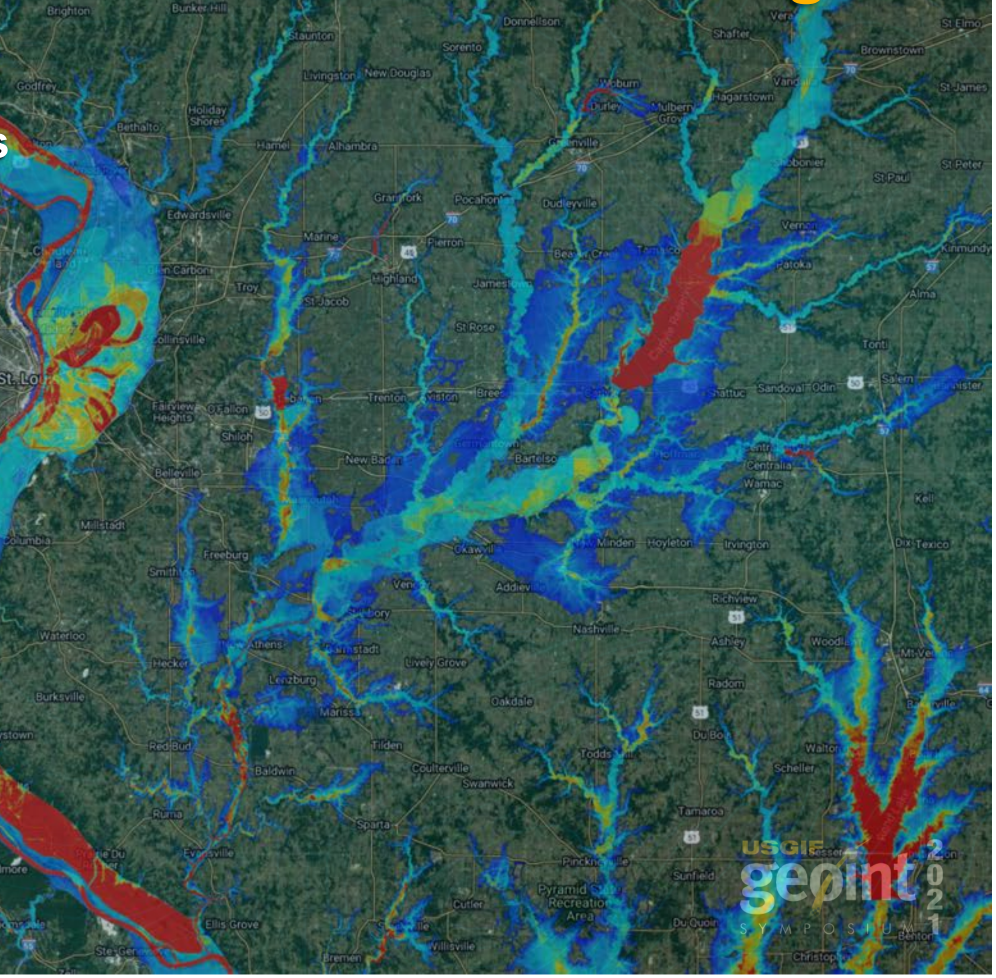
- FIST - Flood Inundation Surface Topology
- Collaboration with NGA (Kevin Dobbs)
  - Began at Nov 2018 NCSA Workshop
  - OTA award Sep 2020 – Jan 2022
- Multi-dimensional effort by NFI
  - Compute and storage for CONUS DEMs
  - Method implementation and refinement
  - Algorithm optimization and parallelization
  - Code development, evaluation, and testing
  - Workflow prototyping for production deployment



Flood inundation depth predicted from terrain elevation data.

CONUS Hydro-conditioning and Inundation Modeling

- Modified TauDEM
  - Limit memory usage at large rank counts
  - Enhanced scalability of algorithms
  - Parallel raster output in VRT format
  - Tiled raster output for random access
- Inundation model code
  - MPI parallelization
  - Tile decomposition
  - Dynamic scheduling
- Leaflet web-based visualization
  - Fully MPI-parallel gdal2tiles.py
  - Parallel tar of output directory tree
  - Tiles served from VM local filesystem



Modest HPC use reduces wait time, enhances interaction

- Hydro-conditioning
  - 4 hours on 4,320 cores
- Inundation modeling
  - 20 minutes on 512 cores
- Web-map tile generation
  - 35 minutes on 2,048 cores
- Web-map tile server
  - 2 cores, 8 GB RAM, 1 TB HDD
  - Managed by campus admins

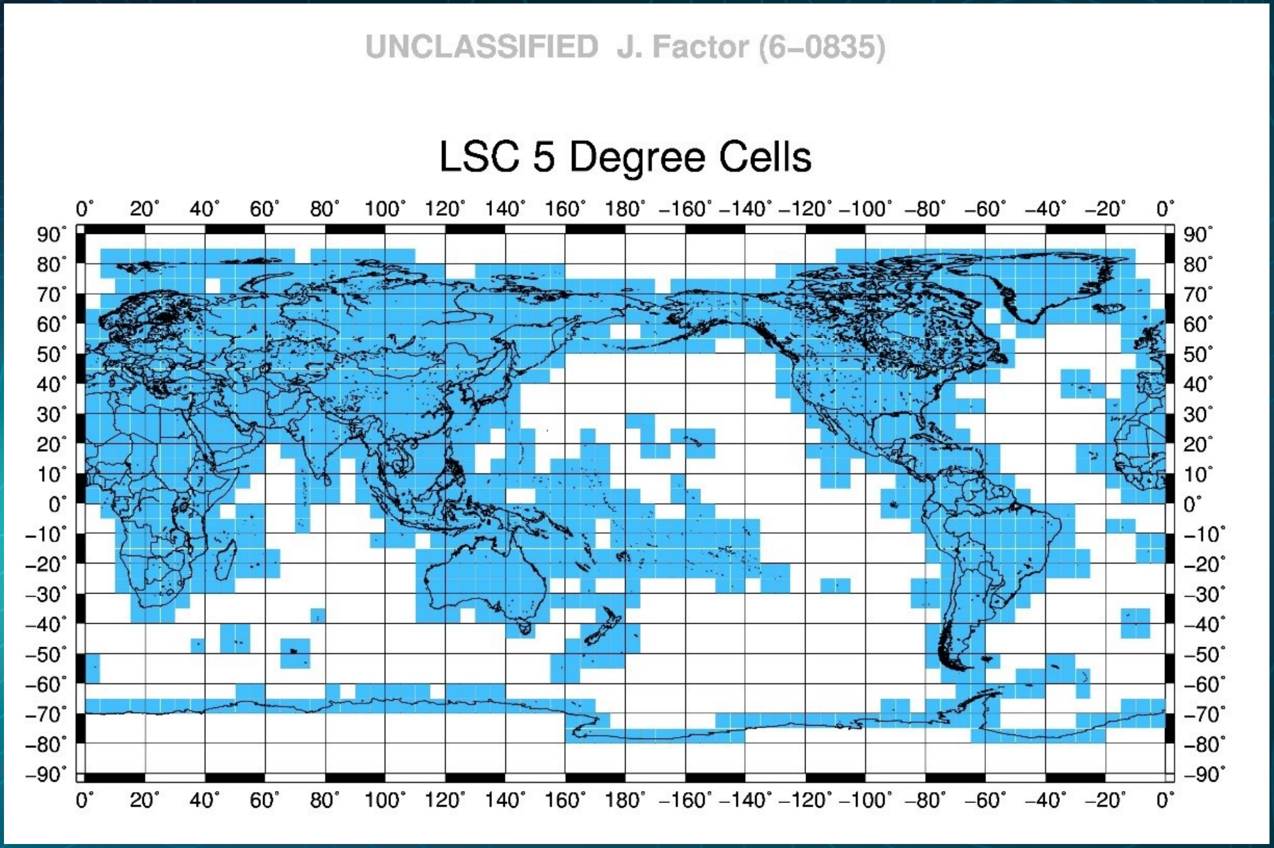
- Interactive solutions used:
  - Jupyter notebooks
    - Standard, supported solution
  - User-space HTTP server
    - View tiles on parallel filesystem
  - VM-hosted web server
    - Better small file performance
    - Publicly available
  - Custom client/server web app
    - GUI-based interactive modeling

USGIF geoint SYMPOSIUM 2021

NGA Earth Gravity Model (EGM)

- NGA Geomatics Group at NGA West
  - Initial meeting at Illinois Summer 2018
  - Goal: Reduce time of EMG (2020) model generation and evaluation cycle
- HPC Areas
  - Modernize legacy workflow.
    - Used MPI based work queue.
    - Allowed us to address workload imbalance.
  - Code performance analysis
    - Compiler study – open source and commercial, flags
    - Code optimization
    - Solver replacement with optimized libraries.
  - Scale out to modest number of nodes (function of LSC cell count)

NGA Earth Gravity Model

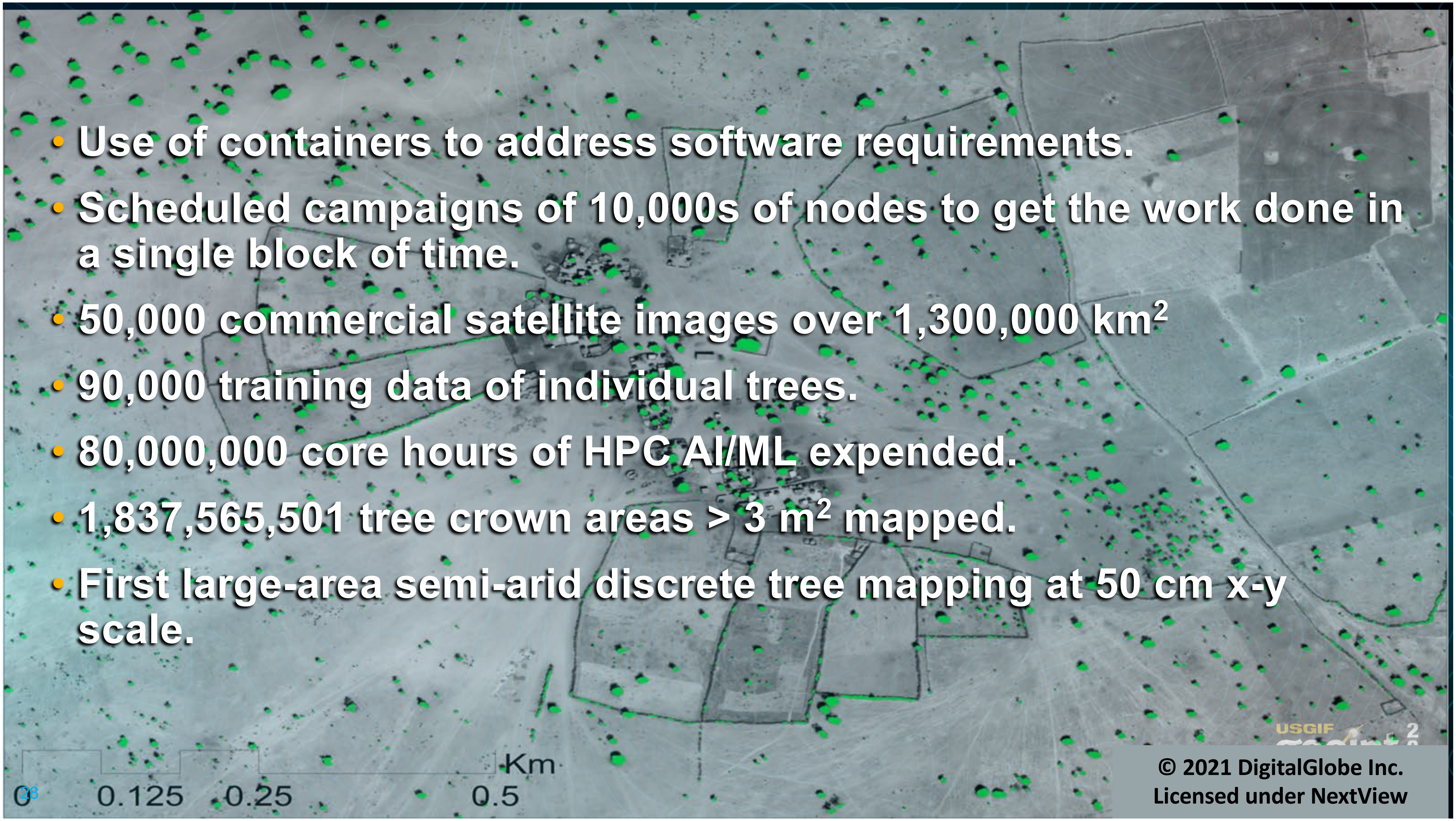


- Up to ~10x speed-up from code optimizations, solver replacements.
- Time reduced from a week to less than 6 hours when scaled to 4,800 cores (160 nodes).

NASA Tree Enumeration by Satellite

- **NASA led project with collaborators from PGC, Illinois, …**
- **Goal**
  - **Develop a framework for accurate and timely determination of biomass to understand land Carbon sink.**
  - **Use Saharan region for proof of concept.**
- **Approach**
  - **Use commercial satellite data at 50 cm**
  - **Detect tree crowns when trees are green & ground is brown**
  - **No overlap—no multiplicative counting**
  - **Manually assembled 90,000 individual trees training data**
  - **AI/ML to identify individual trees**
  - **Use HPC systems for large scale inferencing**

- Use of containers to address software requirements.
- Scheduled campaigns of 10,000s of nodes to get the work done in a single block of time.
- 50,000 commercial satellite images over 1,300,000 km^2
- 90,000 training data of individual trees.
- 80,000,000 core hours of HPC AI/ML expended.
- 1,837,565,501 tree crown areas > 3 m^2 mapped.
- First large-area semi-arid discrete tree mapping at 50 cm x-y scale.

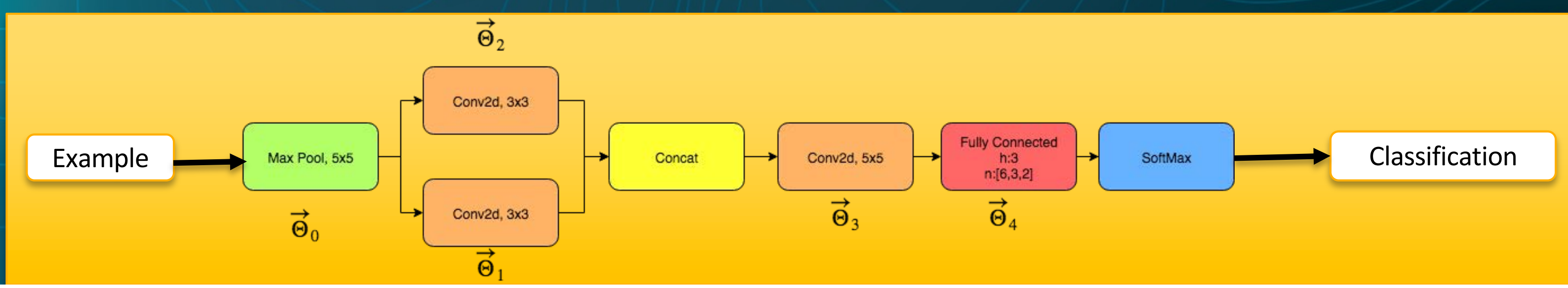


http://nfi.illinois.edu/GEOINT

USGIF geoint SYMPOSIUM 2021

Machine Learning At Scale

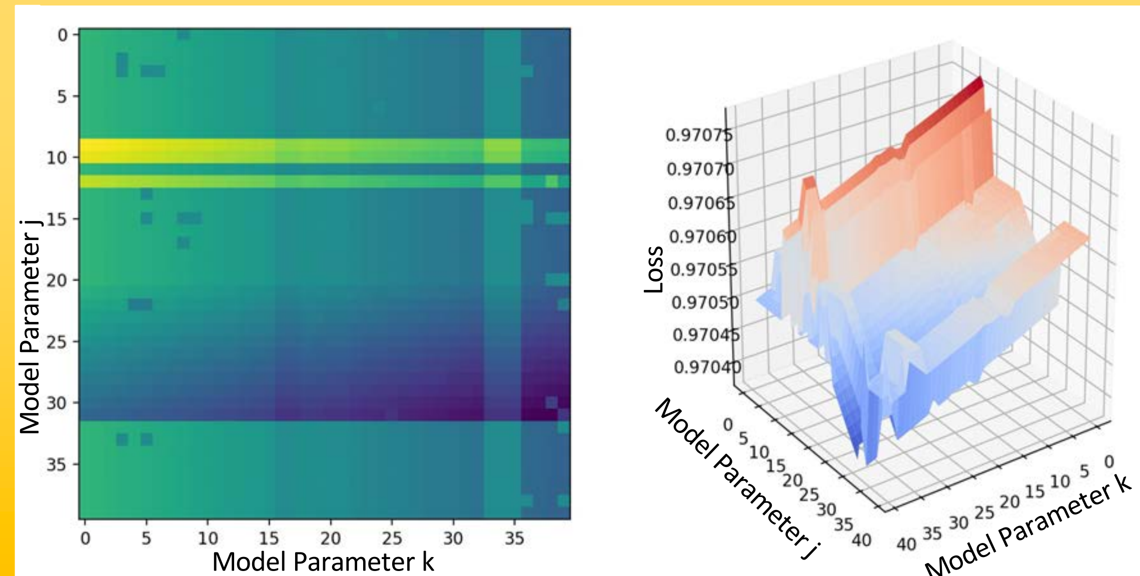
- Training vs. Inference
  - Inference on single example
    - Serial Computation
    - No inter-process communication
    - Scales: Embarrassingly Parallel
  - Training with Stochastic Gradient Decent (SGD)
    - Mini-Batch for parallelization
    - Iterative process



USGIF geoint SYMPOSIUM 2021

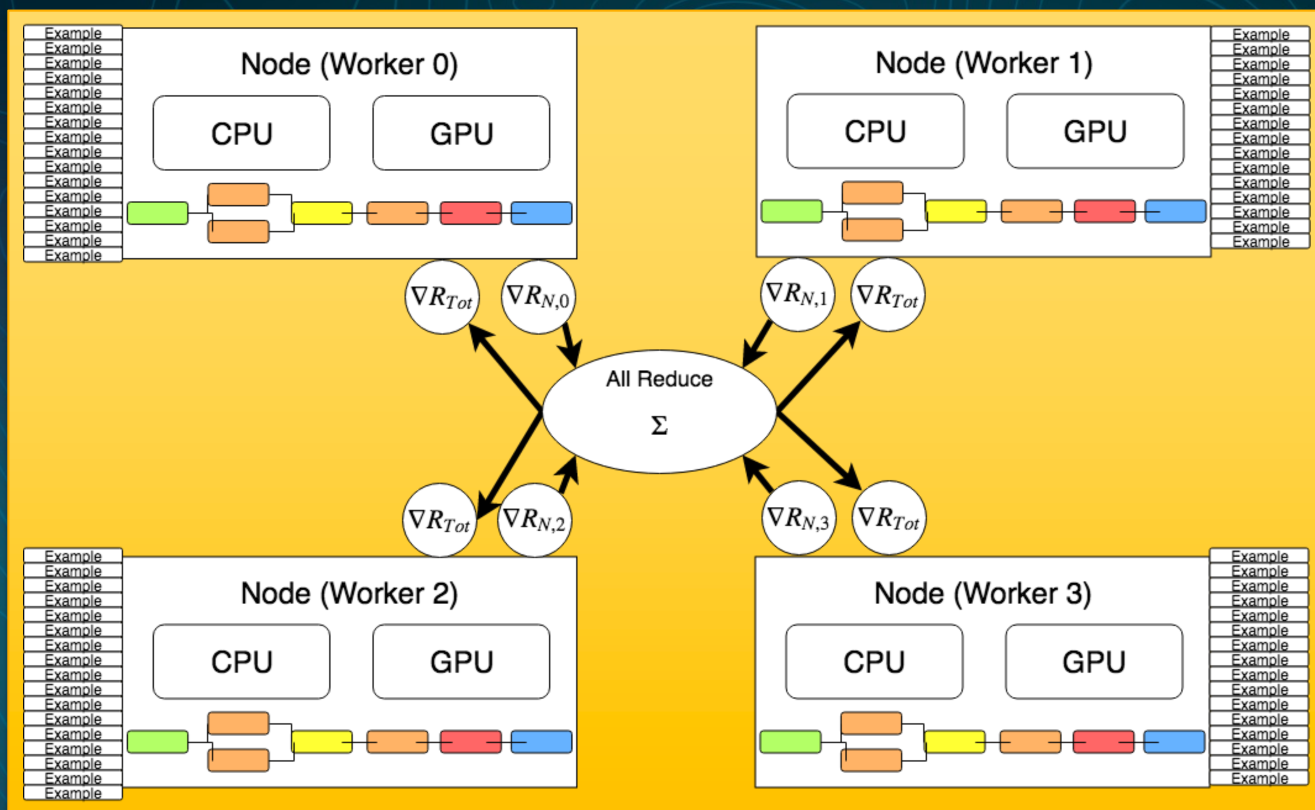
Machine Learning At Scale: Training

- Loss
  - $R = \sum_{m \in B} l(\Theta, m)$
  - B: Batch of Examples
- Sum in loss function is where we exploit parallelism
  - $R = \sum_{m \in B_1} l(\Theta, m) + \cdots + \sum_{B_k} l(\Theta, m)$
  - $B_1, B_2, \ldots$ partitions B
    - Called "Mini Batches"
- $\Theta_{N+1} = \Theta_N + \gamma \cdot \nabla_\Theta R$
  - γ: Learning rate
  - Θ_N: Model parameters at training step N



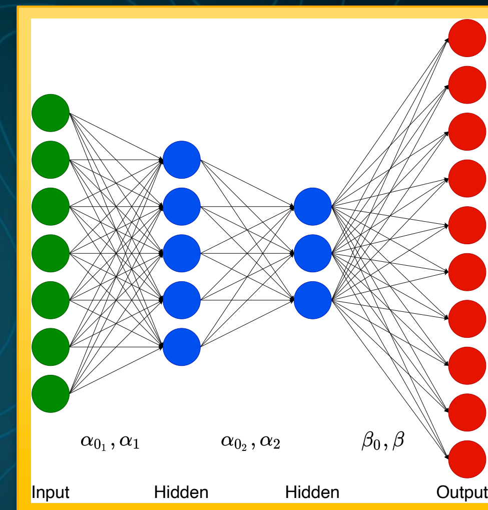
USGIF geoint 2021 SYMPOSIUM

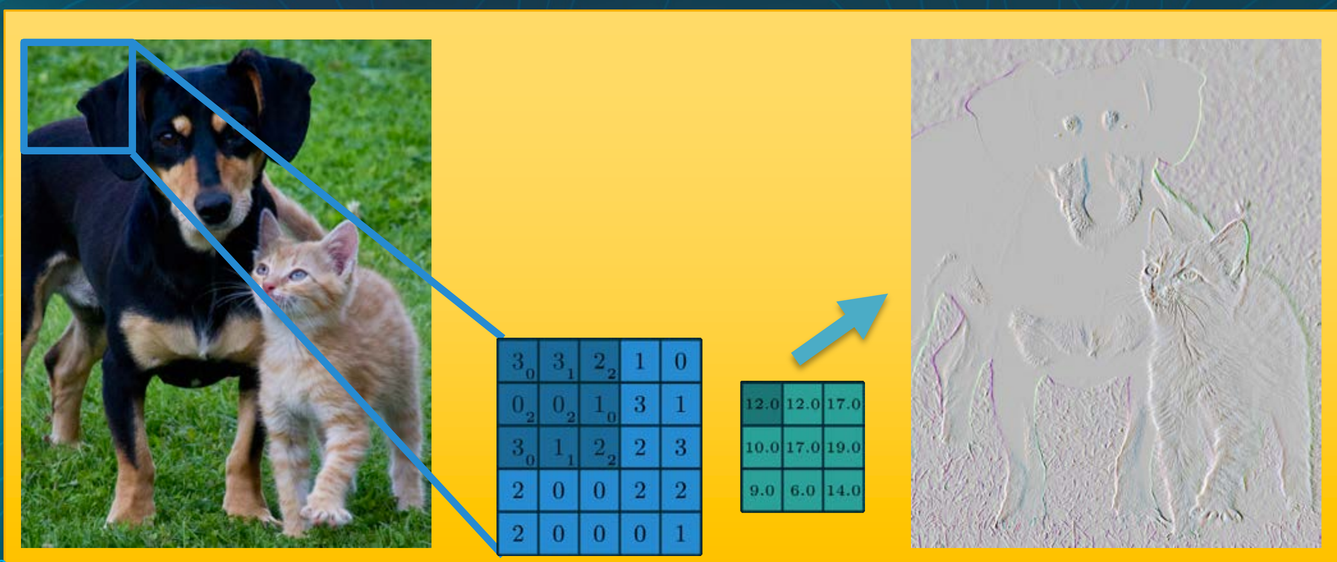
Machine Learning At Scale: Training



Machine Learning At Scale: Training

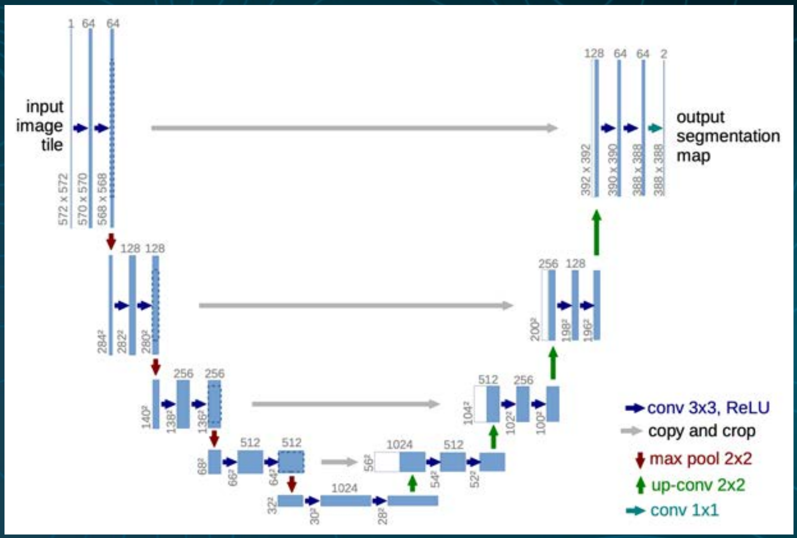
- Common ML Layers
  - Neural Networks
  - Convolutions

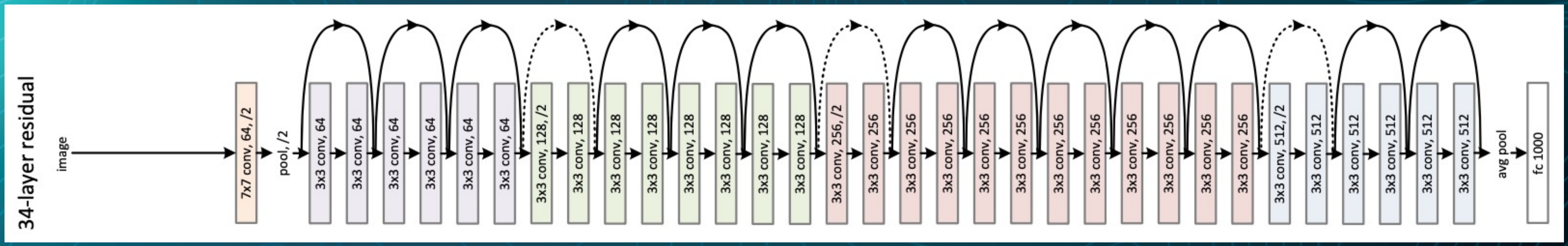




Machine Learning At Scale: Training

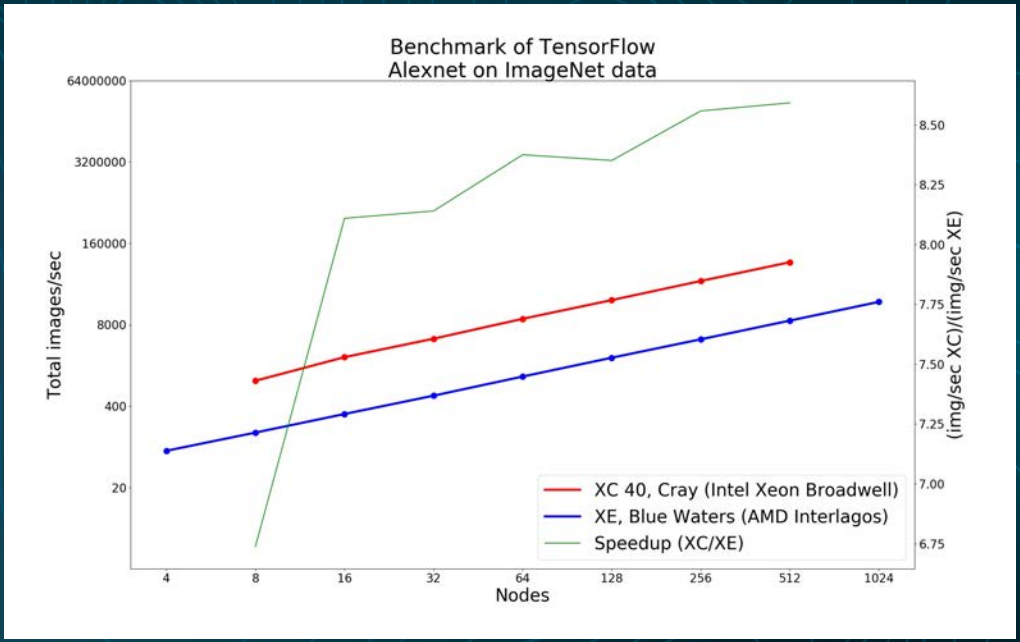
- With these layers you can build
  - ResNet
  - U-Net





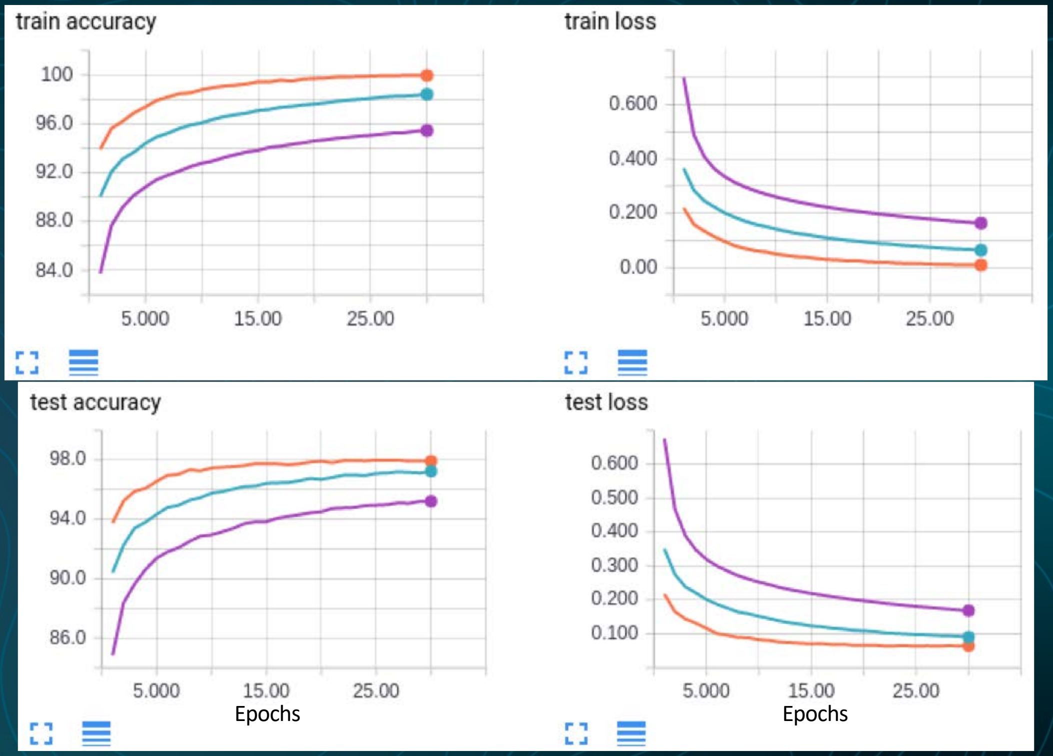
Machine Learning At Scale: Training

- Gemini vs. Aries interconnect
- Indeed!
  - Linear scaling w.r.t. example/sec
  - Different interconnect, but similar scaling



Machine Learning At Scale: Training

- Example of hyperparameter tuning
- Training on MNIST (handwriting) dataset
- Model is a NN with 2 FC layers
- Orange: Batchsize 64
  Blue: Batchsize 256
  Purple: Batchsize 1024

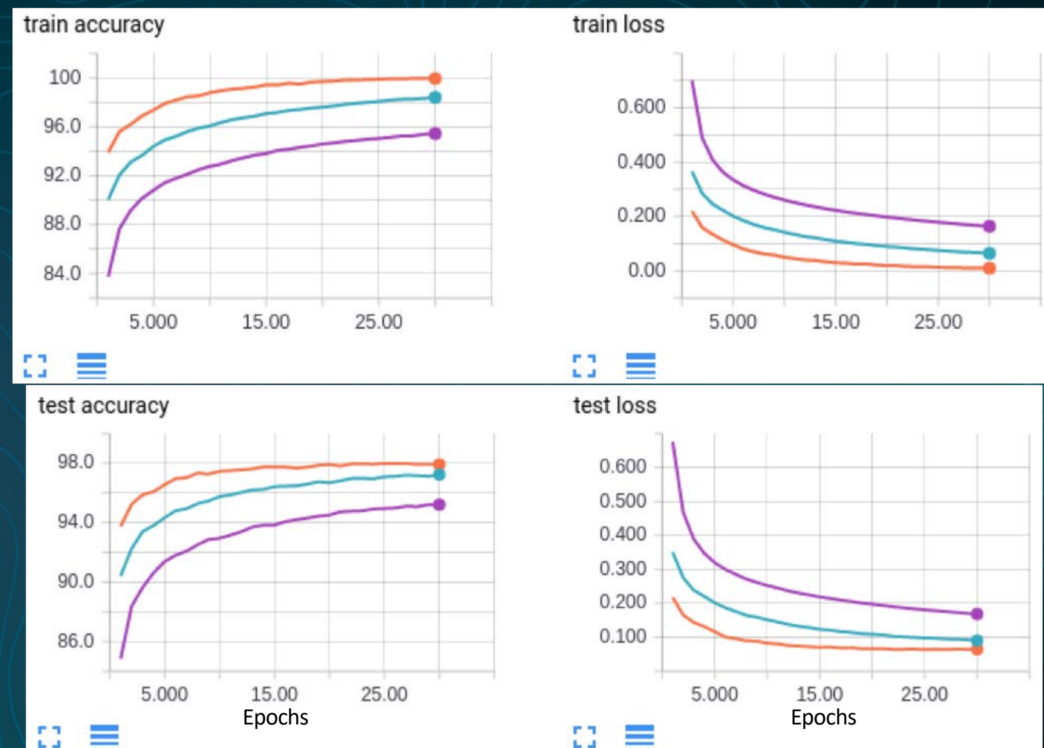


https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e

USGIF geoint SYMPOSIUM 2021

Machine Learning At Scale: Training

- Not all is lost!
- 2017, "Extremely Large Minibatch SGD: Training ResNet-50 on ImageNet in 15 Minutes" T. Akiba et. Al.
  - Mini Batch of 32k
- Orange: Batchsize 64
  Blue: Batchsize 256
  Purple: Batchsize 1024



https://medium.com/mini-distill/effect-of-batch-size-on-training-dynamics-21c14f7a716e

USGIF geoint 2021 SYMPOSIUM

http://nfi.illinois.edu/GEOINT

Do you need to scale-up a project?

- HPC can have a significant benefit on human and project productivity and an accelerated time to discovery for a broad range of geospatial intelligence challenges.
- Strategies for achieving these benefits:
  - Access live and recorded training sessions
  - Consult with experts in the field
  - Improve application codes and workflows
  - Apply for access to HPC resources
  - Partner with organizations to enhance geospatial intelligence

For More Information

- **Contacts**
  - **Bill Kramer -** wtkramer@illinois.edu
  - **Greg Bauer -** gbauer@illinois.edu
  - **Aaron Saxton -** saxton@illinois.edu
- **Blue Waters Project** - https://bluewaters.ncsa.illinois.edu
- **Illinois New Frontiers Initiative** – https://nfi.illinois.edu

We will be in exhibit booth #1739

USGIF geoint 2021 SYMPOSIUM