Azure Cloud Computing Practice in DSM2 Simulation Applications

CWEMF, Apr.6th, 2022

(Joey) Yu Zhou, Raymond Hoang, (Nicky) Prabhjot Sandhu
Modeling Support Office, CADWR

Outline

Why using cloud computing?

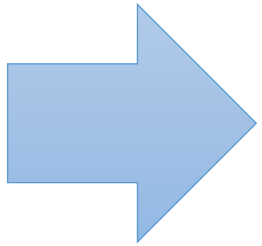
Microsoft Azure

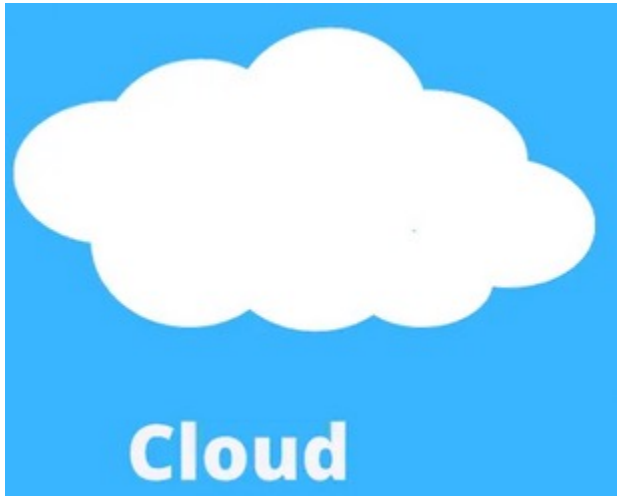
- Key concepts and access
- Resource and process management

Example Applications

- DSM2 batch simulation, calibration and postprocessing

Why using Cloud Computing



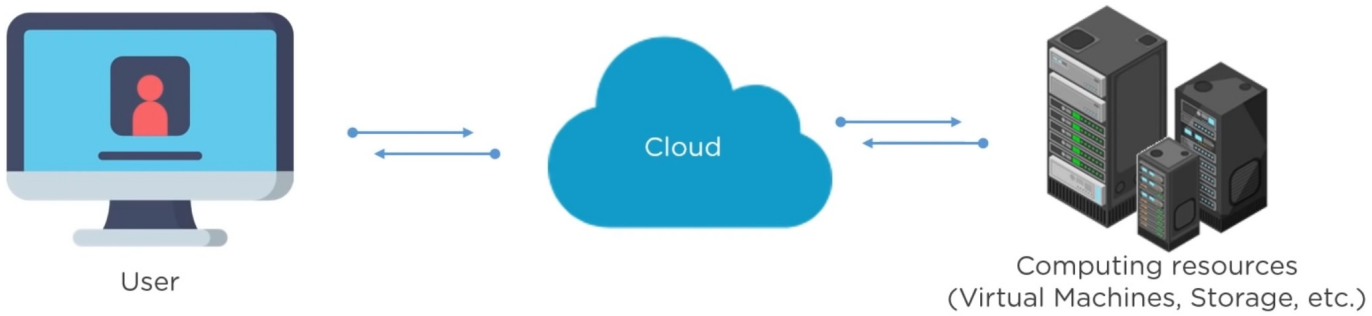


Challenges

- Heavy workload within short timeframe
- Initial cost on hardware; limited configuration; not fully utilized
- Maintenance/update/security

Cloud Computing

Cloud computing is a platform that provides access to computing resources over the internet



Microsoft Azure

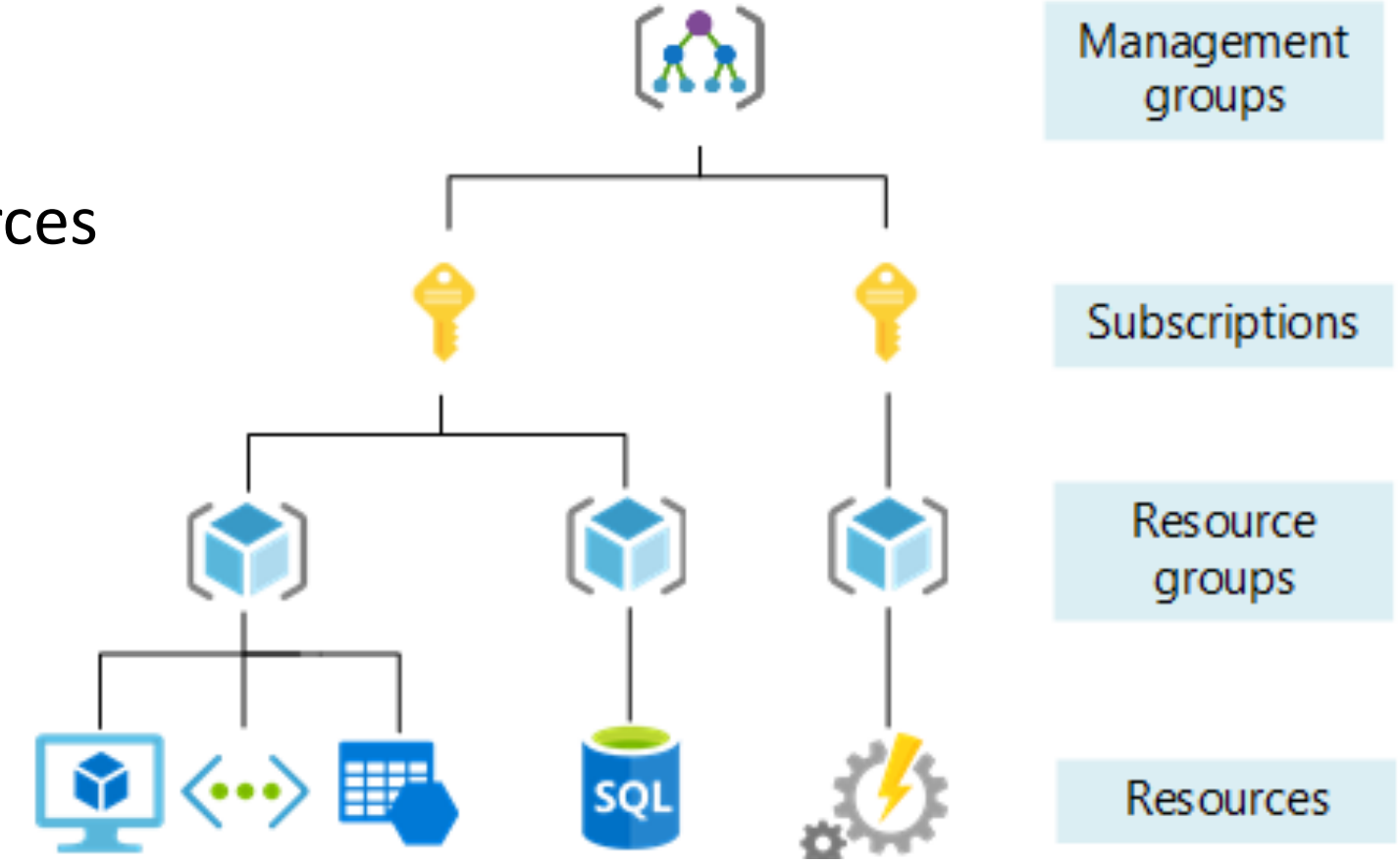
- Easy for Windows users to migrate
- More than 200 products and services

Benefits of Cloud Computing

- Speed
- Flexibility
- Security
- Cost
- Collaboration

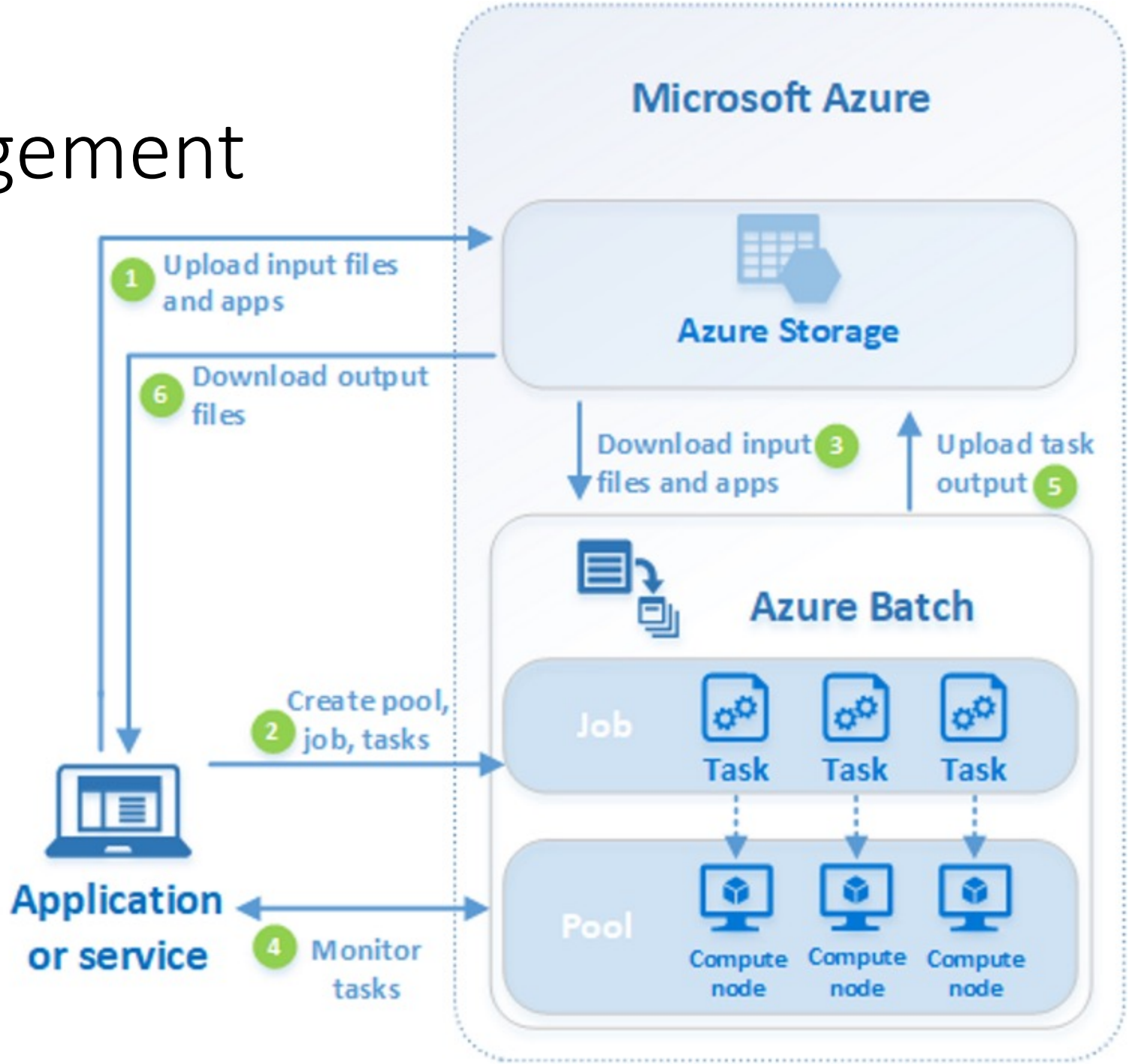
Resources Management

- Policy, and compliance
- Manage costs and the resources
- Logical containers like web apps, databases, and storage accounts
- Instances of services, like virtual machines, storage, or SQL databases



Process Management

- Schedule
- Monitor
- Manage
- Automate



Access to Azure

Web Portal

Windows APP

Mobile

Command line Interface (CLI)

API (C#, Java, Python, etc)

Azure Cloud Computing Examples

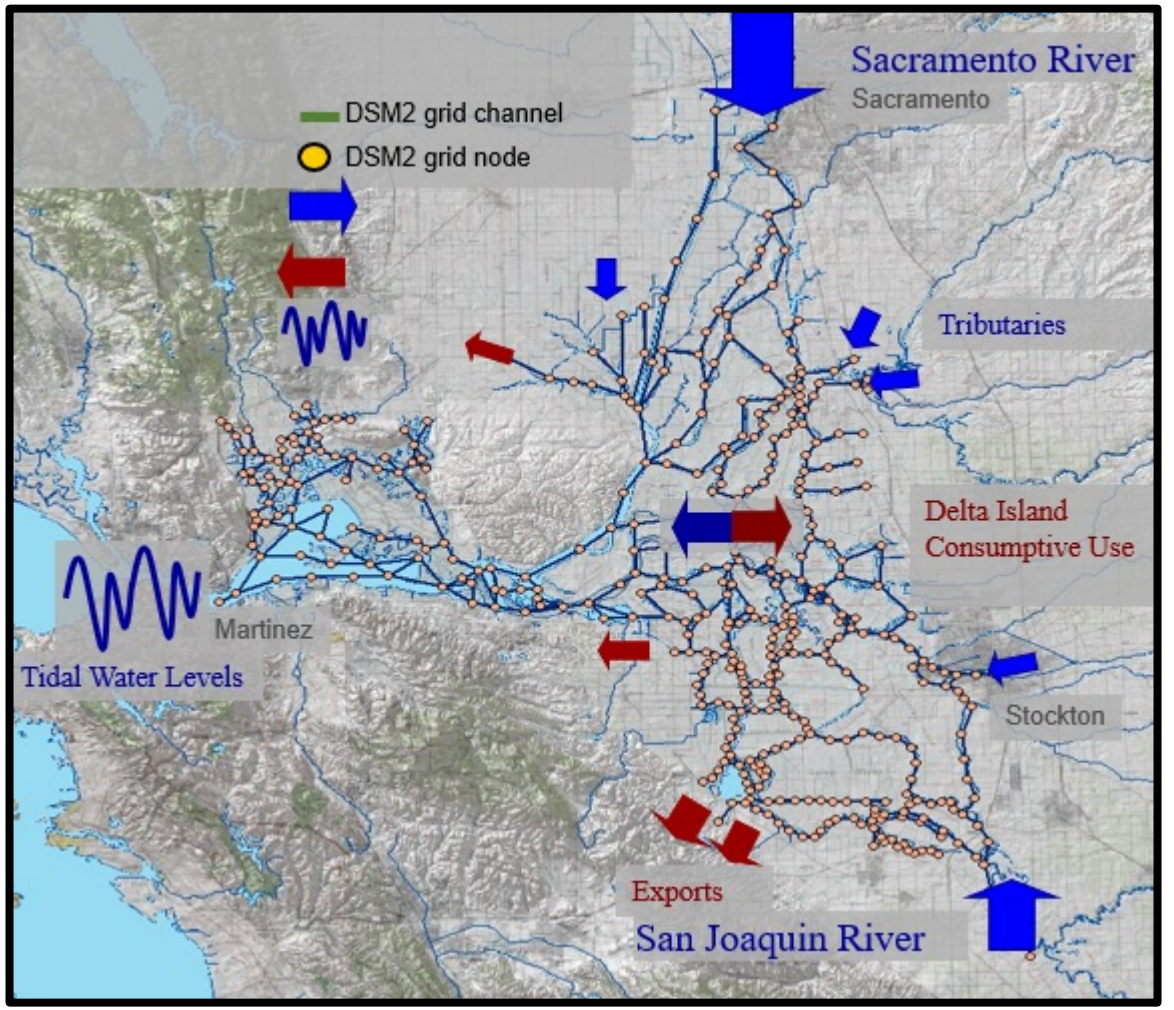
DSM2 Batch Simulation

DSM2 Calibration with PEST

Post-process

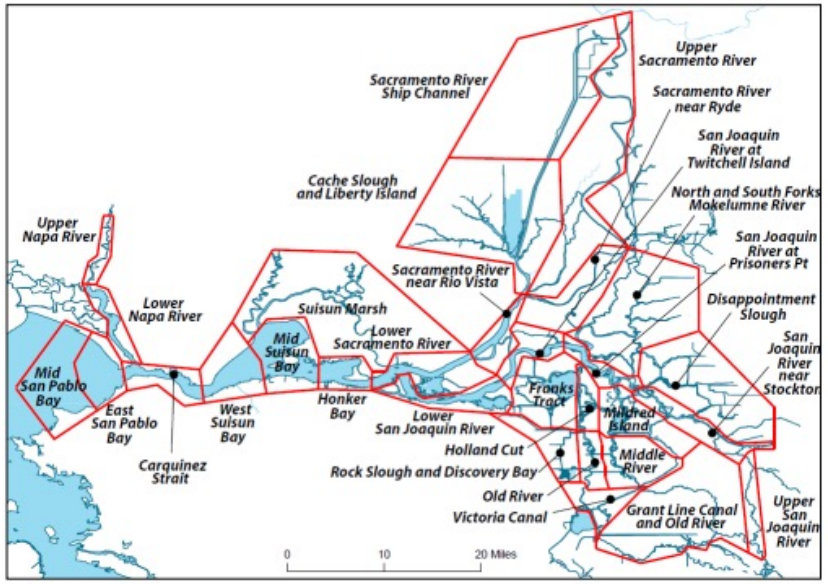
DSM2

- 1-dimensional flow and water quality model
- DSM2 has been applied to the Sacramento-San Joaquin Delta
- Used for planning studies, historical studies, real-time operations
- Sub-modules: hydro - qual/ptm/gtm



https://water.ca.gov/Library/Modeling-and-Analysis/Bay-Delta-Region-models-and-tools/Delta-Simulation-Model-II

DSM2 Residence Time Simulation



Residence Time

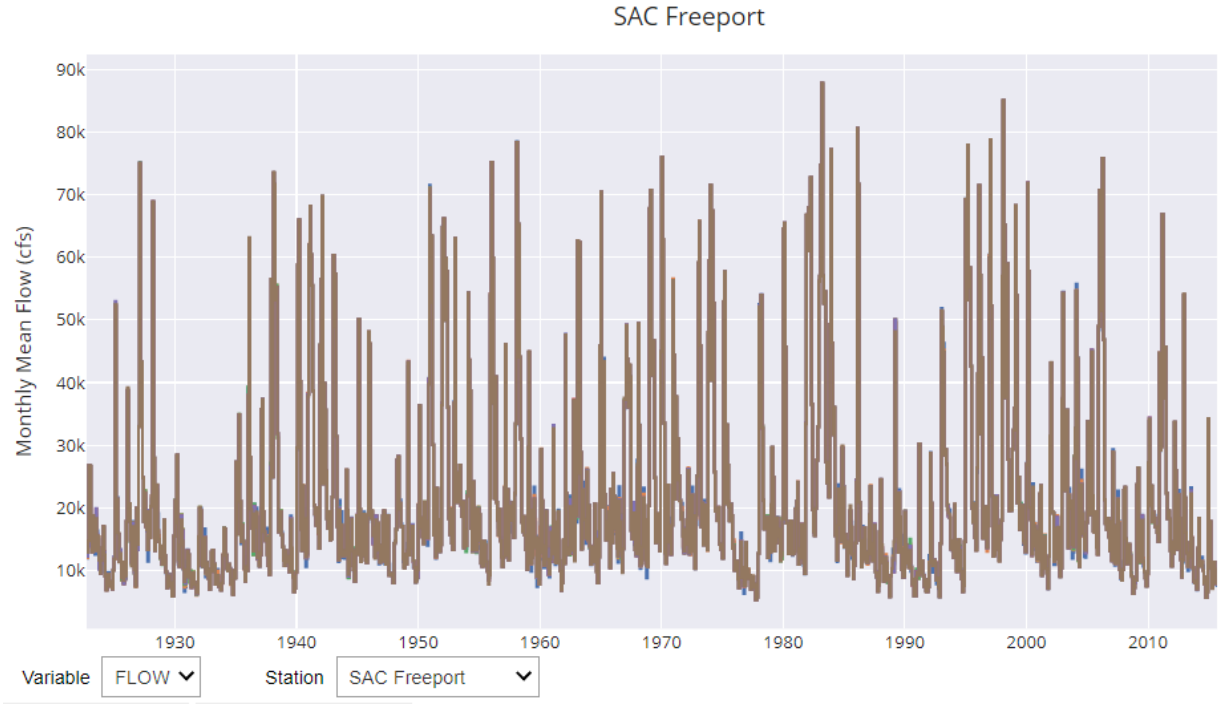
- Track and monitor water constituent concentration

Scenario conditions

- Multiple local regions
- Multiple time periods across the years

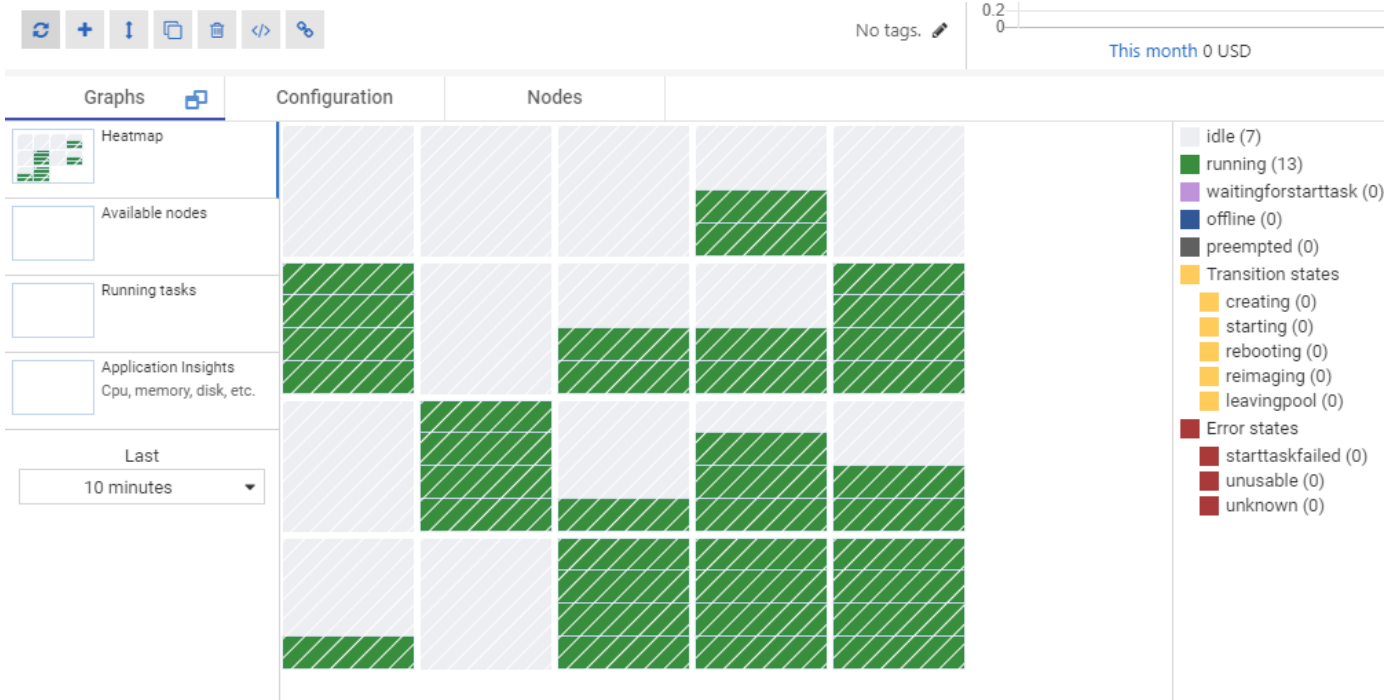
Two approaches

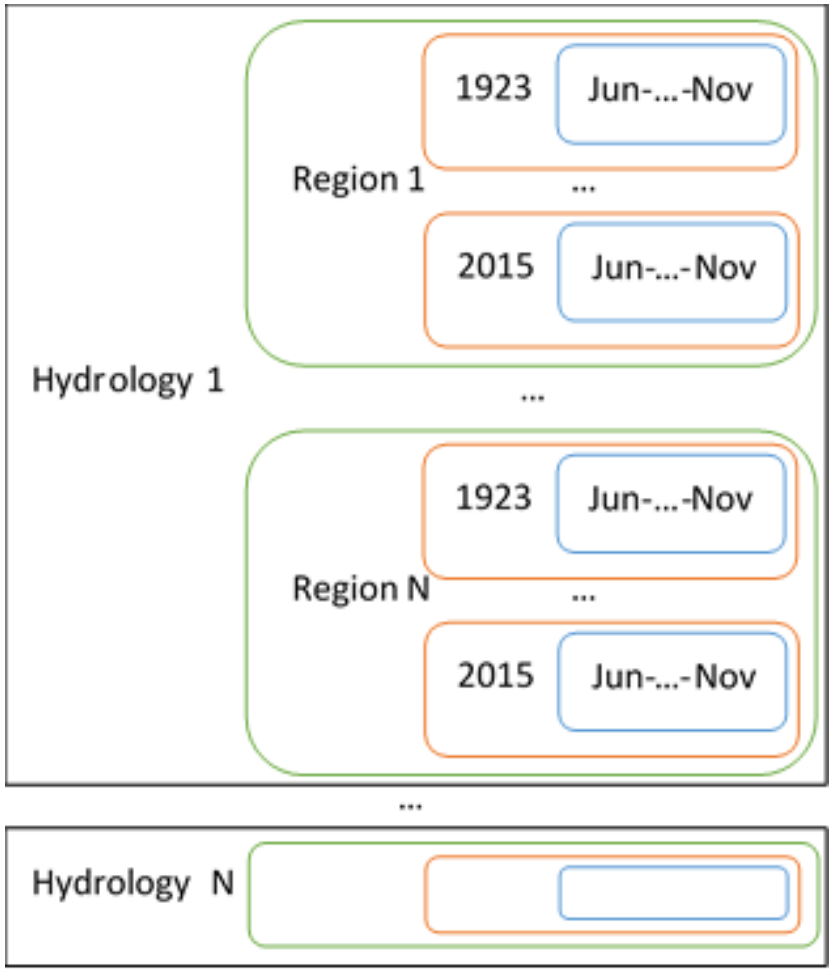
- Representative combo of region + time
- A comprehensive suite of scenarios



Batch Runs on Azure Cloud

- Virtual Machines with multiple CPU
- 20*4 scenarios running in parallel
- Azure takes 2 hours; cost $10





- 6700 scenarios covering 93 years, regions, ops
- Computation time: 168 hours

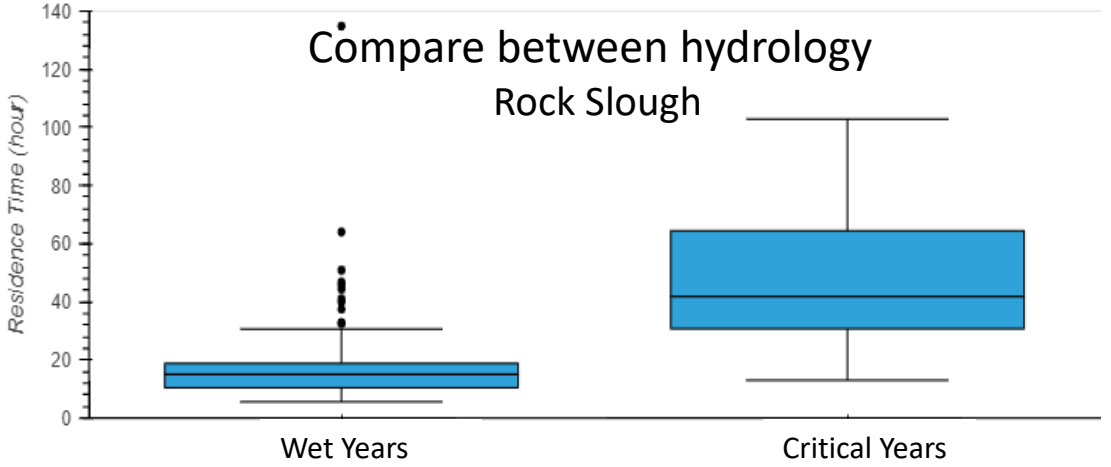
Post-process on Cloud

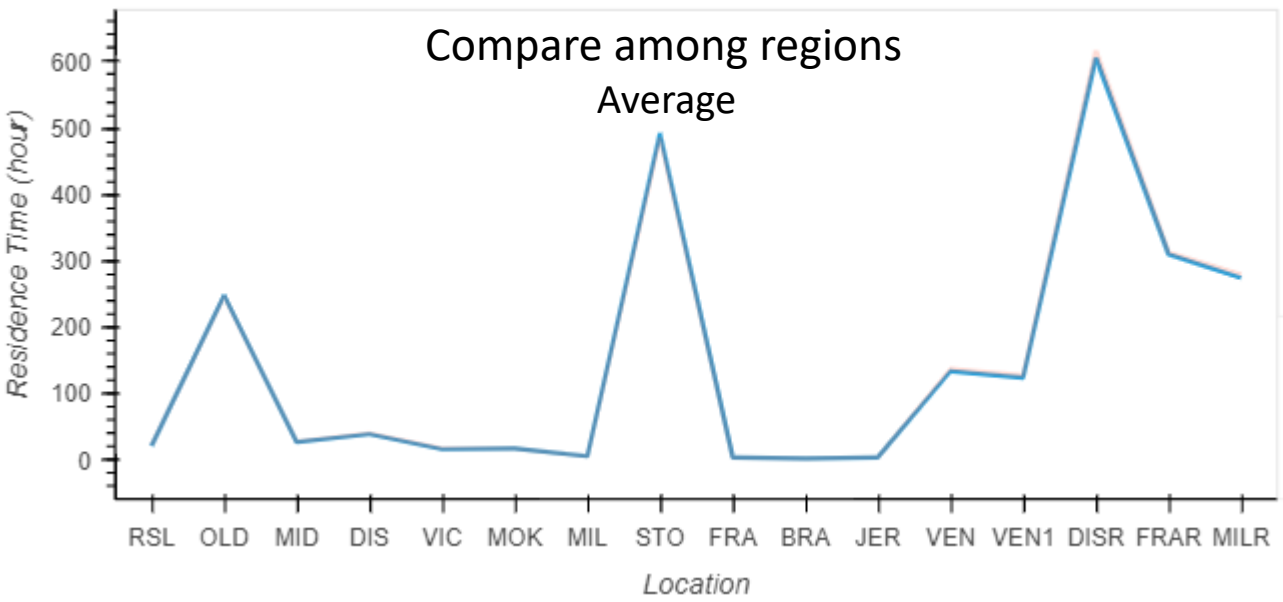
Save massive data on cloud storage

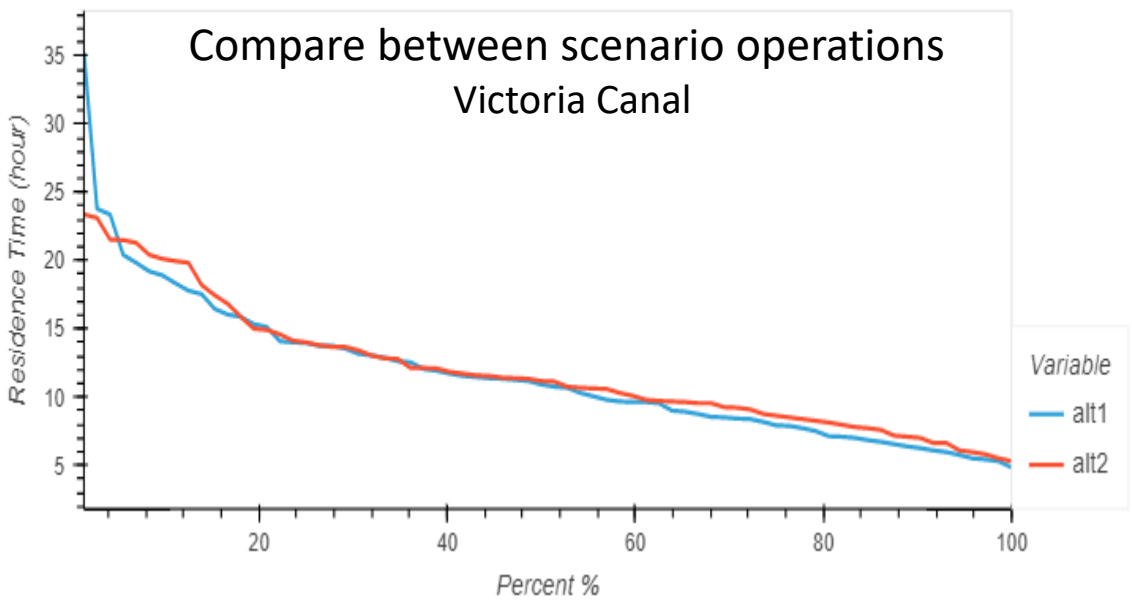
Efficiency of transfer inside vs outside cloud

Only download the required info to local

- Statistical summary
- Details for specific checking

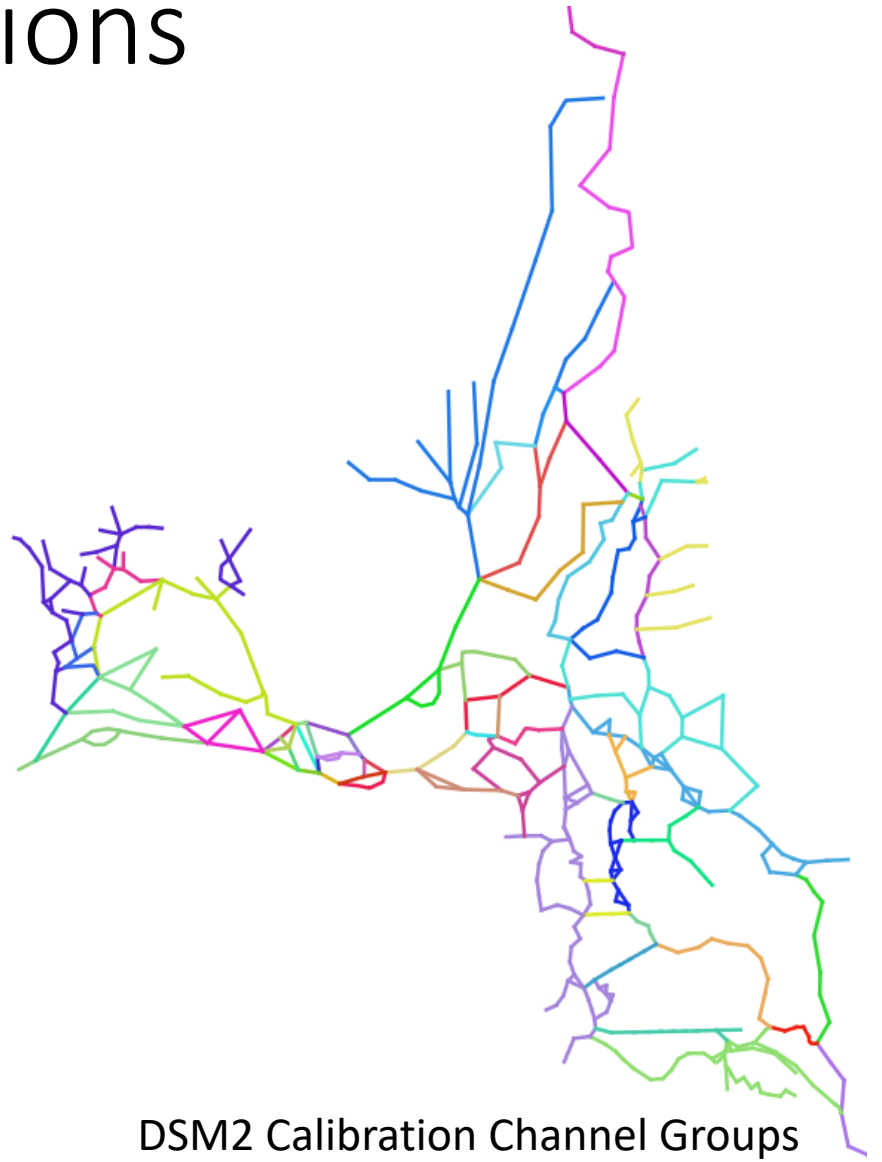






DSM2 Calibration Trial Assumptions

Calibration Period	WY 2011 – WY 2013
Calibration Parameters	Manning's n for DSM2 65 *channel groups,* where *e*ach group assumed to have the same Manning's n
Observation Locations	Historical flow and stage for available stations in the Delta
PEST Objective Function	Minimize differences between observed data and modeled outputs

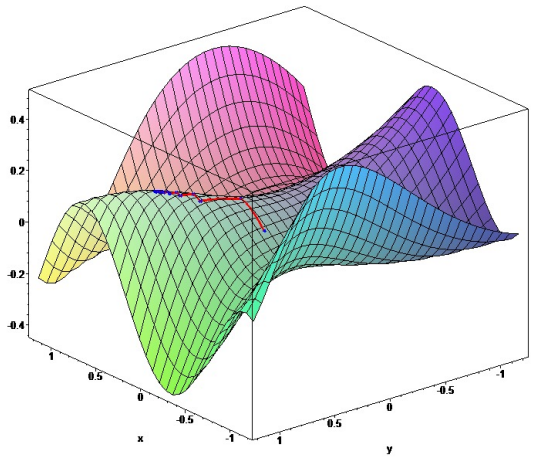


DSM2 Calibration Channel Groups

Parameter ESTimation with PEST and DSM2

- Uses gradient descent methodology
- Parameter values are adjusted based on the derivatives of the observations with respect to the parameters
- PEST iterates DSM2 to build a sensitivity matrix

Iterate DSM2 with incremental parameter adjustments

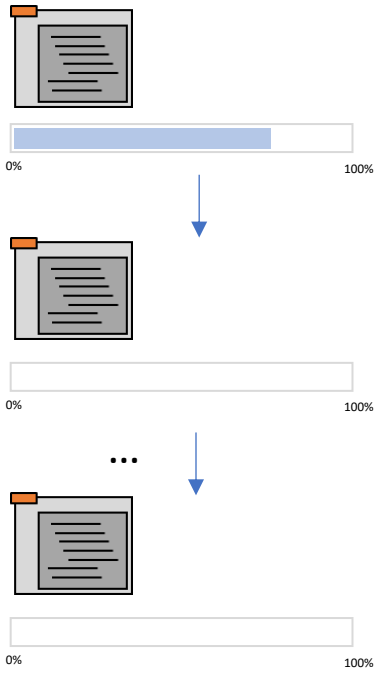


Gradient Descent Algorithm

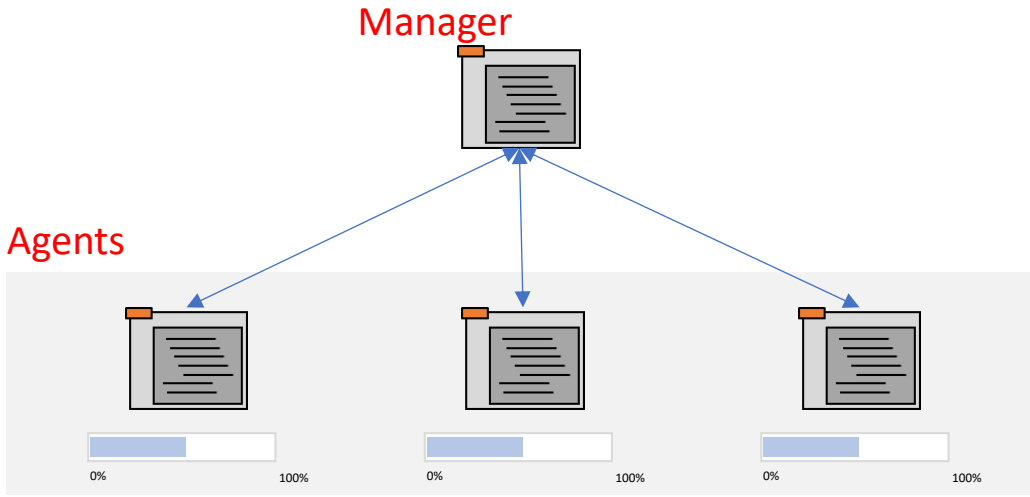
Parallel Processing with BeoPEST

BeoPEST is a parallel run manager. The longest part of the PEST run is running DSM2 to generate the Jacobian matrix – this part can be 100% parallelizable.

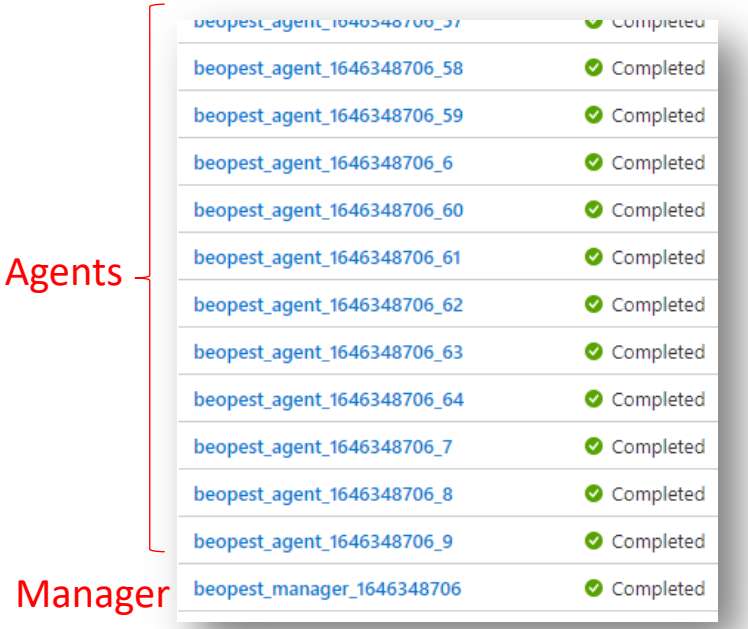
A. Serial Process using PEST



B. Parallel Process using BeoPEST

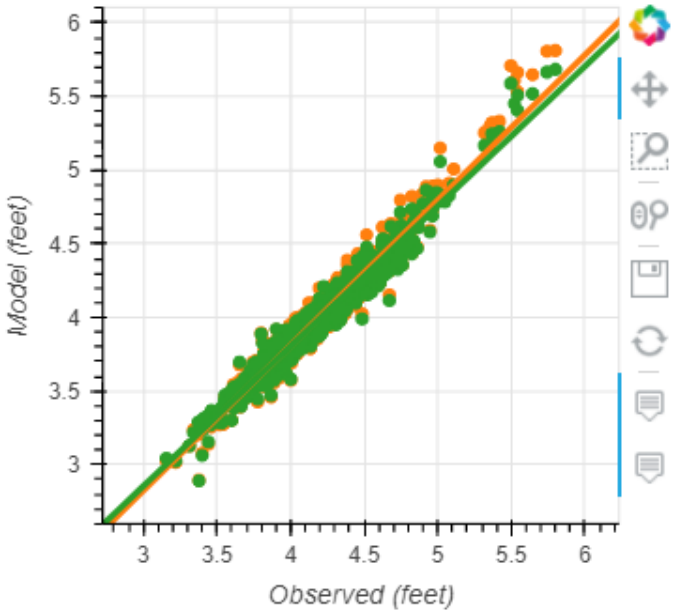


C. Parallel Process using BeoPEST with Azure



Post-Processing

#	Study	Equation	R Squared	Mean Error	RMSE	Amp Avg %Err	Avg Phase Err
0	pestcalib_v150	y=0.98x+96.30	0.99	-223.41	5.33E+02	6.50	10.76
1	v8.2.1	y=0.98x+192.8	0.99	-98.51	4.75E+02	1.64	-15.98




```
                                OPTIMISATION RESULTS

Covariance matrix and parameter confidence intervals cannot be determined:-
Some form or regularisation was implemented so these are not applicable.

Use the PREDUNC7 utility to obtain a full posterior covariance matrix.


Parameters ----->

Parameter       Estimated value
 1                4.500000E-02
 10               3.734990E-02
 11               2.684037E-02
 12               3.000125E-02
 13               2.676912E-02
 14               2.538615E-02
 15               4.500000E-02
 16               3.082014E-02
 17               2.916014E-02
 18               2.445712E-02
 2                1.000000E-02
 3                2.381983E-02
 4                1.000000E-02
 5                1.000000E-02
 6                4.500000E-02
 7                4.500000E-02
 8                1.000000E-02
 9                3.027443E-02

See file dsm2_820.sen for parameter sensitivities.


Observations ----->

Observation             Measured        Calculated      Residual        Weight      Group
                        value           value
bdl-nse-stage           1.00000         0.961106        3.889424E-02    1.000       obgnme
bdt-nse-flow            1.00000         0.936694        6.330615E-02    1.000       obgnme
dsj-nse-flow            1.00000         0.905384        9.461627E-02    1.000       obgnme
dsj-nse-stage           1.00000         0.957133        4.286743E-02    1.000       obgnme
fpt-nse-flow            1.00000         0.993778        6.221887E-03    1.000       obgnme
gct-nse-stage           1.00000         0.813726        0.186274        1.000       obgnme
```

Summary

Azure Cloud greatly facilitated our modeling works

- Computation, storage, post-process
- Sensitivity studies with large amount of independent scenarios
- Parallel process to assist intensive computation of auto-calibration
- Novel model test and technology deployment

Plan to move more suitable works to cloud

- Fast, scalability, cost-effective, easy-to-use, collaboration
- Supplement/replace/balance on-premises equipment

Going to publish sample scripts on Github, open source

Supplemental Slides

- Virtualization
- Azure pricing and cost management
- Access to Azure
- PTM batch run on Azure
- DSM2 sub-modules
- PEST fundamentals
- PEST-DSM2

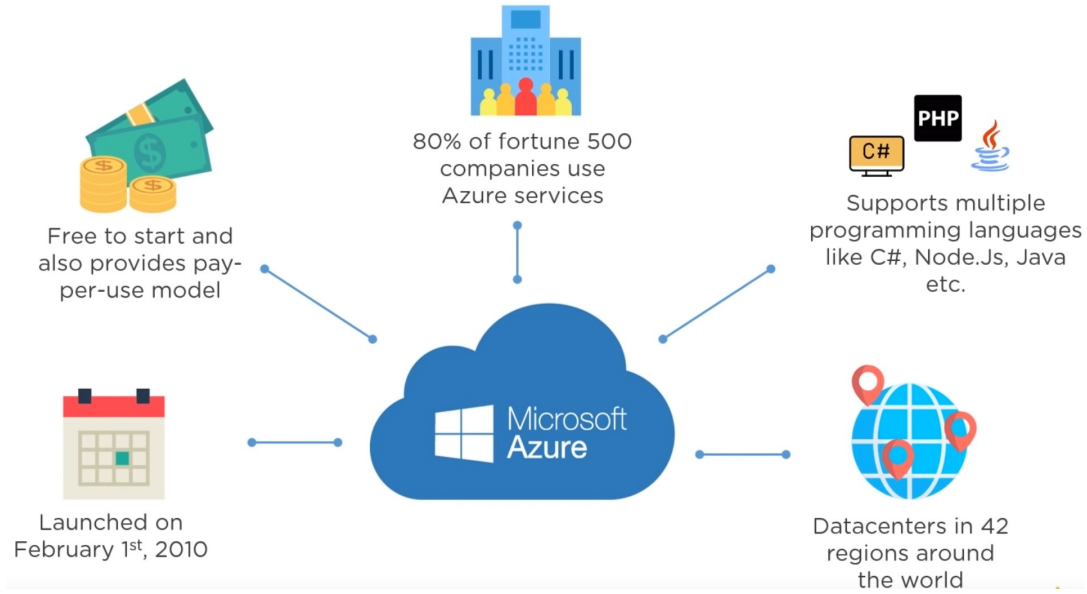
Facilitate DSM2 Compile/Test

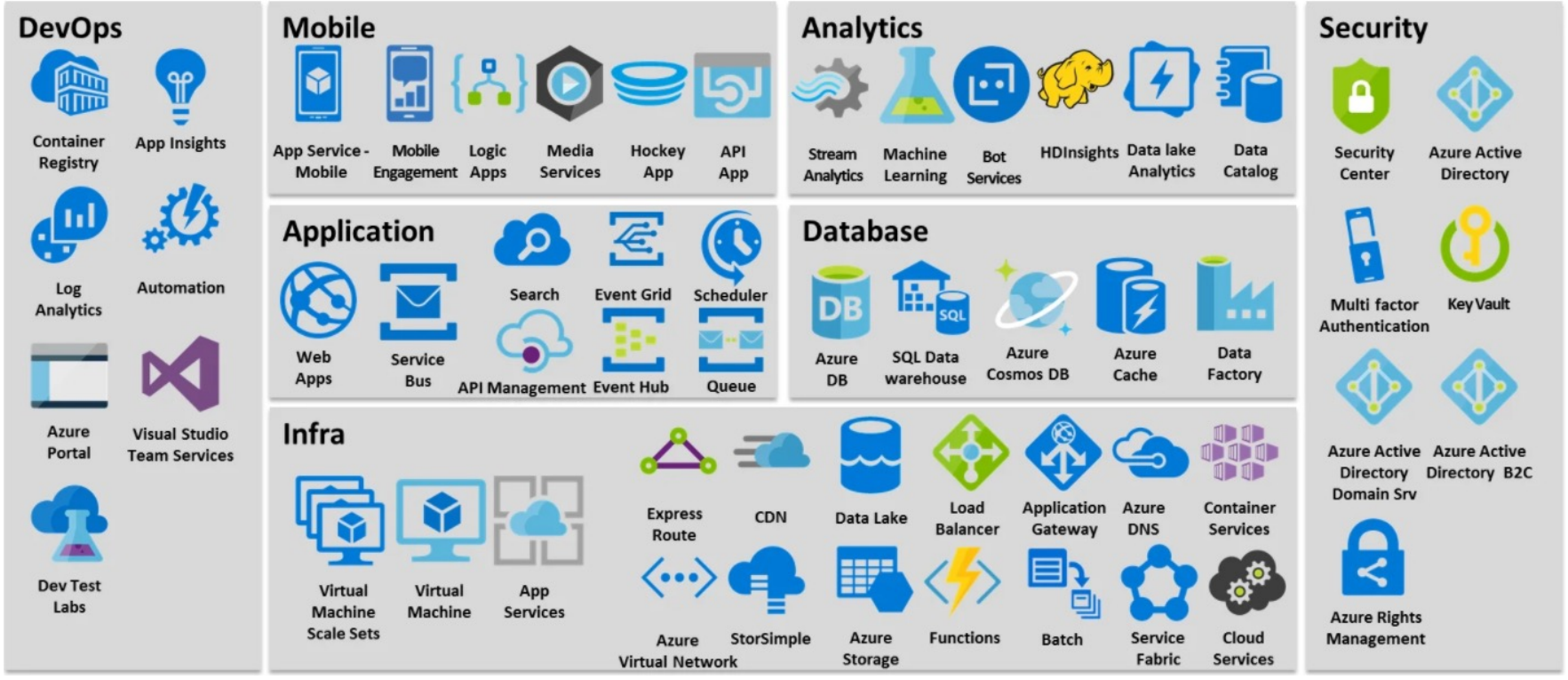
- Mixed programming languages: Fortran, Java, C, Python
- Supporting programs: Intel Compiler, Visual Studio, CMake, Cygwin, Java JDK, etc

- Versions update
- Various operation System
- Various hardware specification
- Coordination between multiple developers

Microsoft Azure

- Easy for Windows users to migrate
- More than 200 products and services





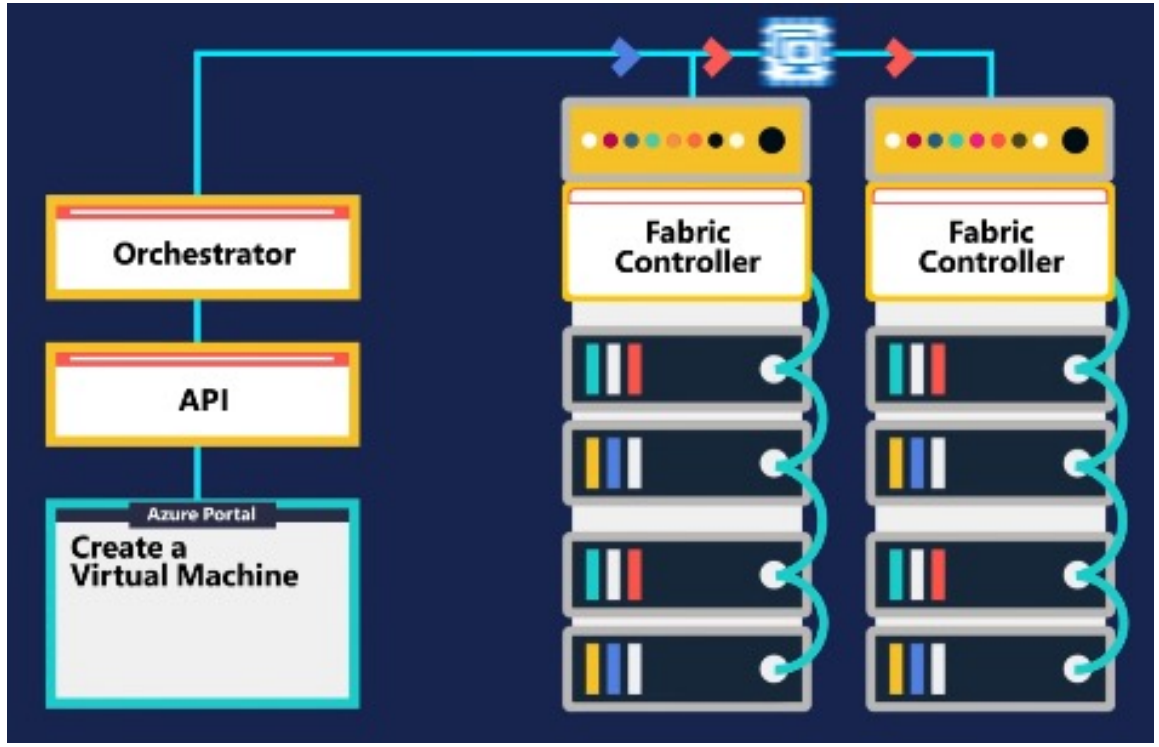
How Azure works

Virtualization

- separate tight coupling of CPU and its OS
- virtualized hardware to execute in software

Hypervisor

- emulating all the functions of a real computer in a VM
- can run multiple VM at the same time.
- apply on massive scale on data centers all over the world



Azure Pricing

Data storage prices pay-as-you-go

All prices are per GB per month.

	Premium	Hot	Cool	Archive
First 50 terabyte (TB) / month	**$0.15** per GB	**$0.018** per GB	**$0.01** per GB	**$0.00099** per GB
Next 450 TB / month	**$0.15** per GB	**$0.0173** per GB	**$0.01** per GB	**$0.00099** per GB
Over 500 TB / month	**$0.15** per GB	**$0.0166** per GB	**$0.01** per GB	**$0.00099** per GB

Virtual Machines

REGION: West US

OPERATING SYSTEM: Windows

TYPE: (OS Only)

TIER: Standard

CATEGORY: All

INSTANCE SERIES: All

INSTANCE: D2 v3: 2 vCPUs, 8 GB RAM, 50 GB Temporary storage, $0.209/hour

- D2s v5: 2 vCPUs, 8 GB RAM, 0 GB Temporary storage, $0.204/hour
- D4s v5: 4 vCPUs, 16 GB RAM, 0 GB Temporary storage, $0.408/hour
- D8s v5: 8 vCPUs, 32 GB RAM, 0 GB Temporary storage, $0.816/hour
- D16s v5: 16 vCPUs, 64 GB RAM, 0 GB Temporary storage, $1.632/hour
- D32s v5: 32 vCPUs, 128 GB RAM, 0 GB Temporary storage, $3.264/hour
- D48s v5: 48 vCPUs, 192 GB RAM, 0 GB Temporary storage, $4.896/hour
- D64s v5: 64 vCPUs, 256 GB RAM, 0 GB Temporary storage, $6.528/hour
- D96s v5: 96 vCPUs, 384 GB RAM, 0 GB Temporary storage, $9.792/hour
- D1 v2: 1 vCPUs, 3.5 GB RAM, 50 GB Temporary storage, $0.126/hour
- D2 v2: 2 vCPUs, 7 GB RAM, 100 GB Temporary storage, $0.252/hour
- D3 v2: 4 vCPUs, 14 GB RAM, 200 GB Temporary storage, $0.504/hour

Virtual machines: 1 × 730 Hours

Savings Options

Save up to 72% on pay-as-you-go prices with 1-year or 3-year Reserved Virtual Machine I
applications that require reserved capacity. Learn more about Reserved VM Instances pric

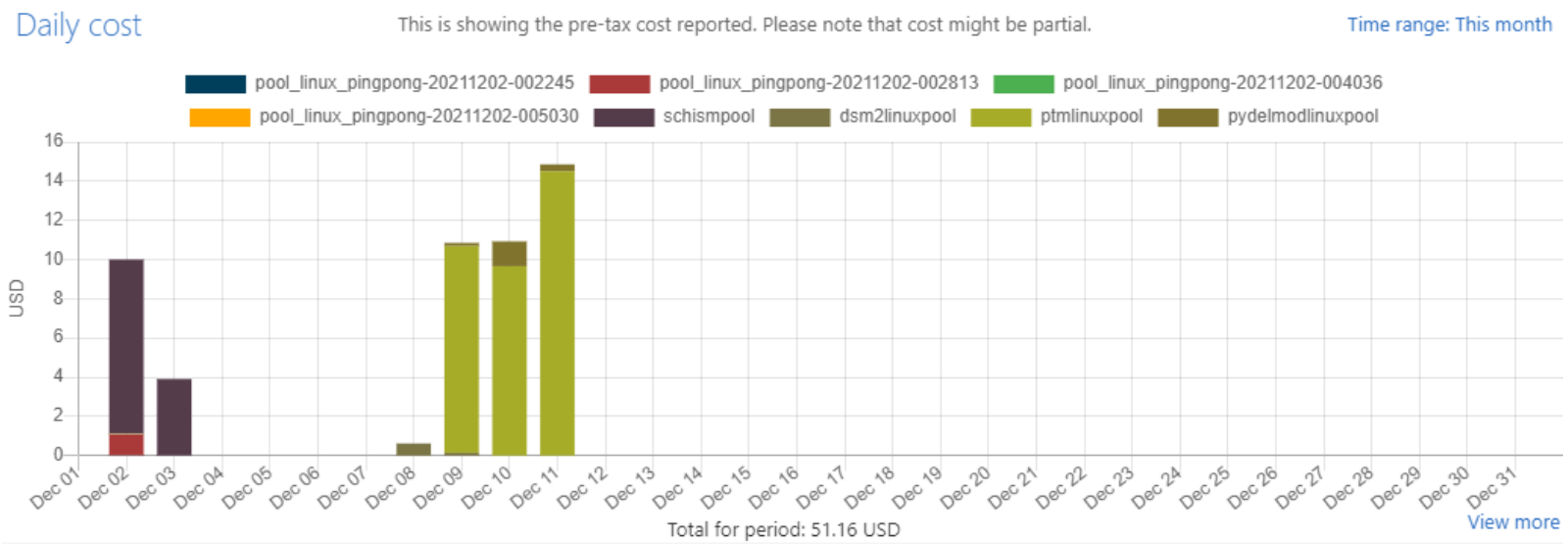
Compute (D2 v3)

- (•) Pay as you go
- ( ) 1 year reserved (~32% discount)
- ( ) 3 year reserved (~57% discount)

OS (Windows)

- (•) License included
- ( ) Azure Hybrid Benefit

Cost Management



Resources

Job status

dsm2jobs	active
neutrally_buoyant_particles_pa3k_2020	active
neutrally_buoyant_particles_pa6k_2020	active
pydelmodjob	active

View all jobs

Pool status

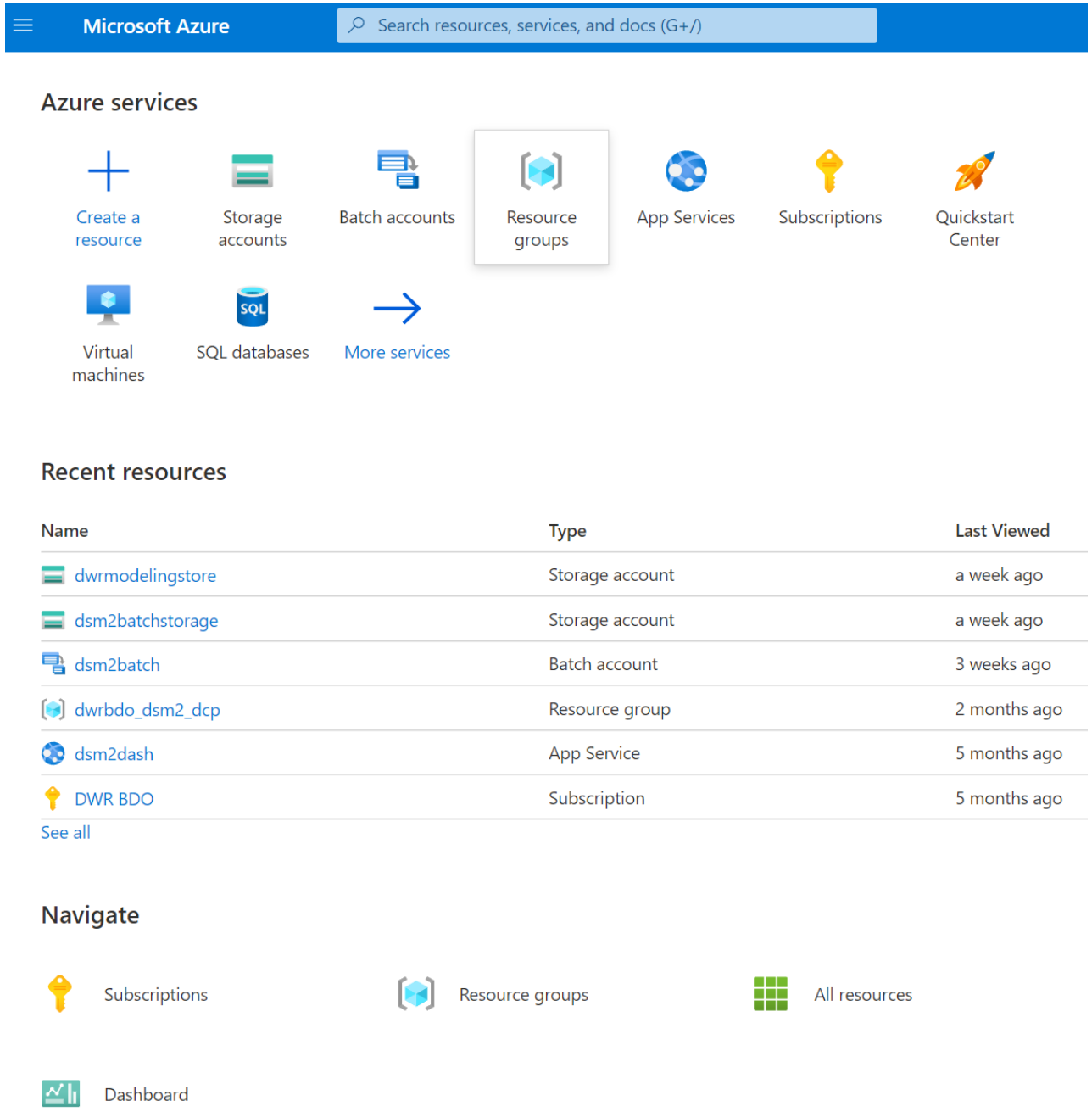
dsm2linuxpool	0
dsm2winpool	0
ptmlinuxpool	0
pydelmodlinuxpool	0

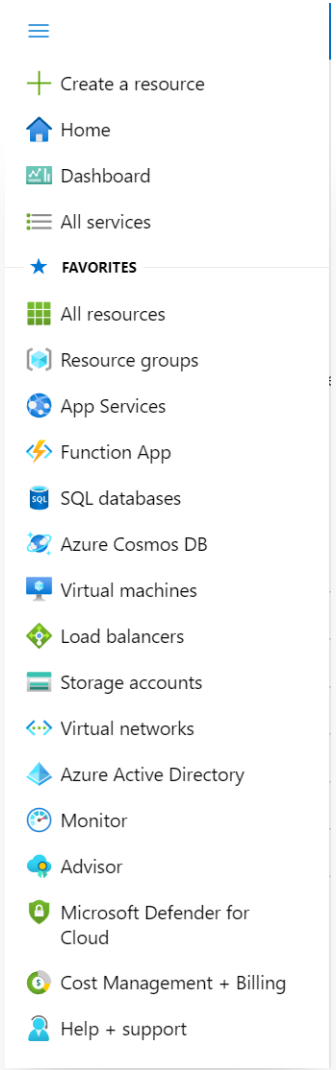
View all pools

App packages

- dsm2
- dsm2linux
- pest
- schism
- unzip
- vista

View all packages

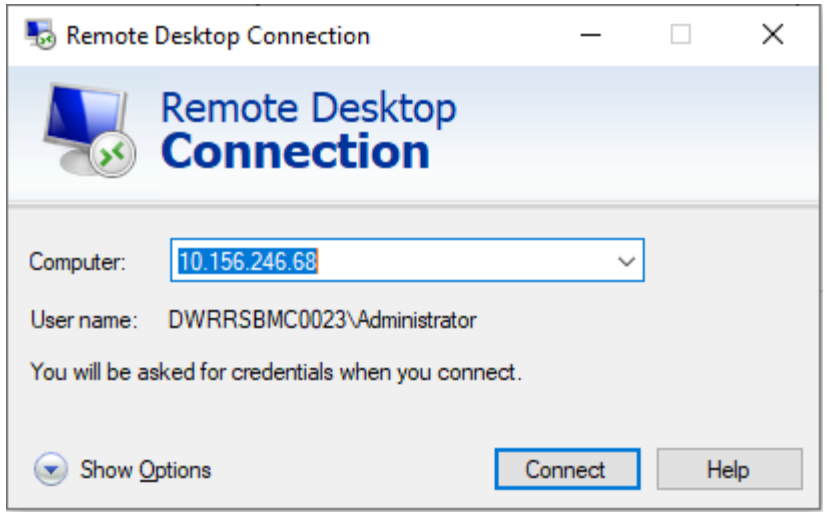


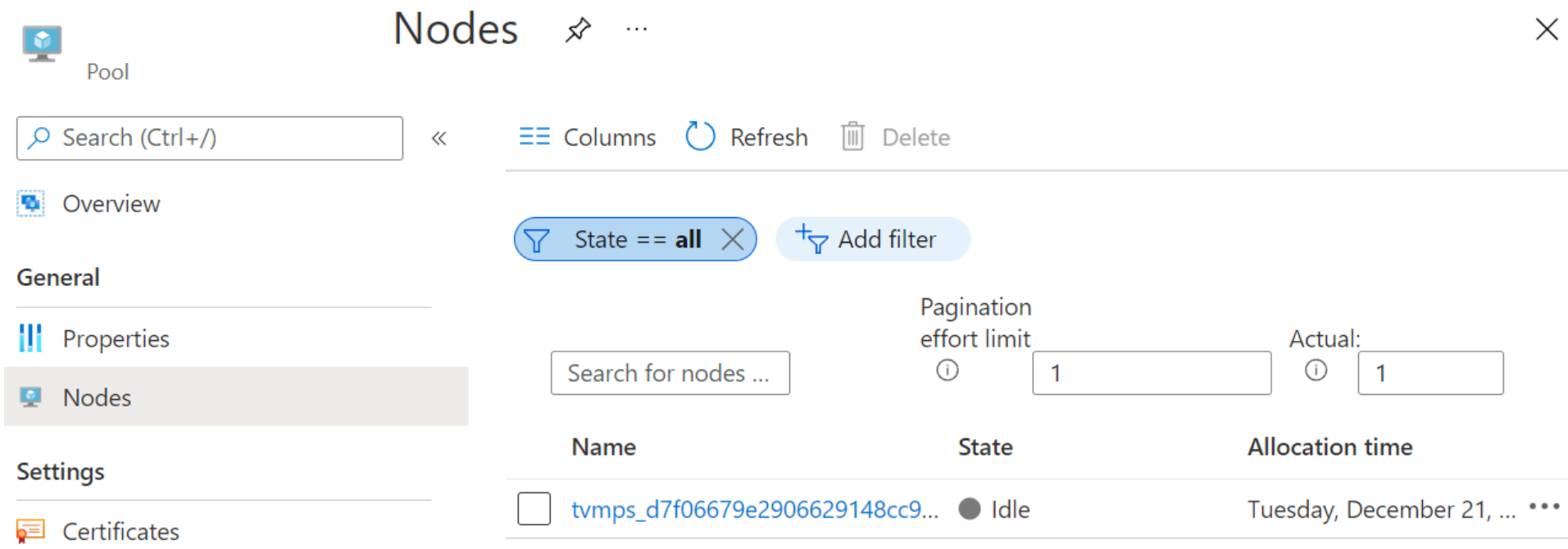


Azure Web Portal

Output, Error, Debug

- Standard output/error log
- Customized output/error log
- Connect to VM





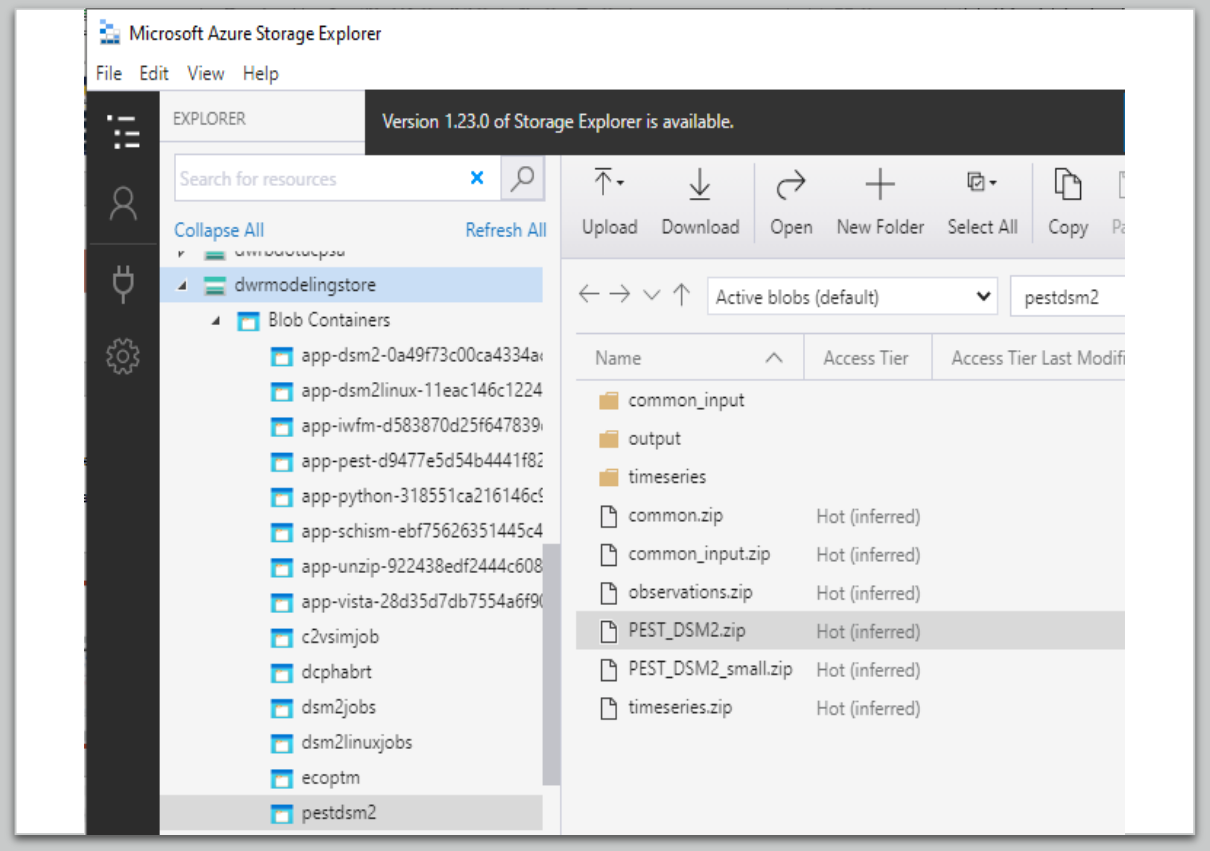
Azure Desktop Apps

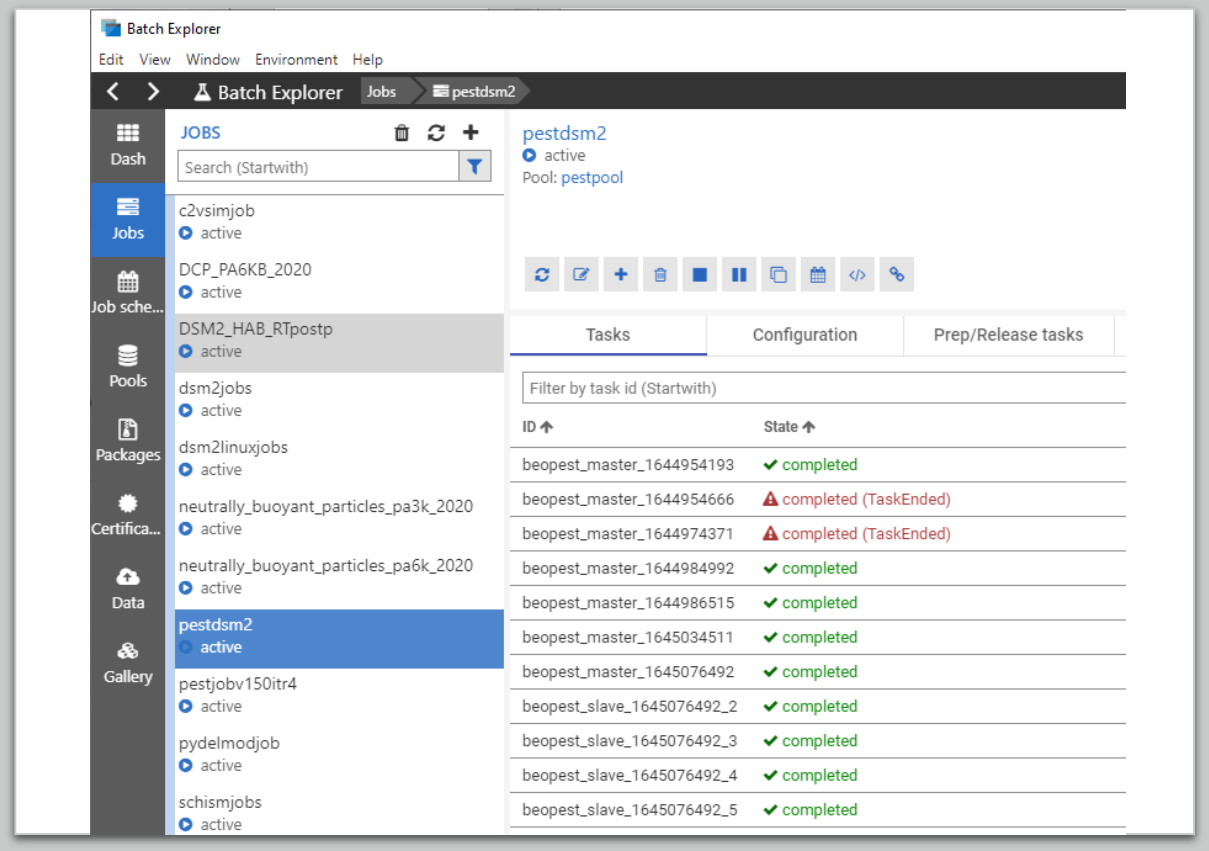
Storage Explorer

- Applications
- Data input/output

Batch Explorer

- Jobs: tasks
- Pools: compute nodes (Virtual Machine)





Python API

- Programmatically
- communicate with Batch service, storage service
- Schedule/submission
- Environment Variable

Imports

In [1]:
```
from dmsbatch import create_batch_client, create_blob_client
```
executed in 4.91s, finished 23:10:33 2021-12-19

First create a batch client from the config file

In [5]:
```
client = create_batch_client('../tests/data/dmsbatch.config')
blob_client = create_blob_client('../tests/data/dmsbatch.config')
```
executed in 74ms, finished 23:11:26 2021-12-19

Application packages

To copy large files and programs it is best to zip (or targz) them and upload them as application packages

Application packages are setup separately in either azure management apis or from the web console or cli tool

These are referenced here by their name and version e.g. DSM2, python and other programs

In [6]:
```
app_pkgs = [('dsm2', '8.2.c5aacef7', 'DSM2-8.2.c5aacef7-win32/bin'),
            ('vista', '1.0-v2019-05-28', 'bin'),
            ('unzip', '5.51-1', 'bin')]
```
executed in 65ms, finished 23:11:31 2021-12-19

Create or resize existing pool

If the pool doesn't exist it will create it If the pool exists, it will resize to the second arg

In [7]:
```
client.create_or_resize_pool('dsm2winpool',
                             1,
                             app_packages=[(app,version) for app,version,_ in app_pkg
                             vm_size='standard_f2s_v2',
                             tasks_per_vm=2,
                             os_image_data=('microsoftwindowsserver', 'windowsserver'
                            )
```
executed in 687ms, finished 23:11:40 2021-12-19

Particle Tracking Model Simulation

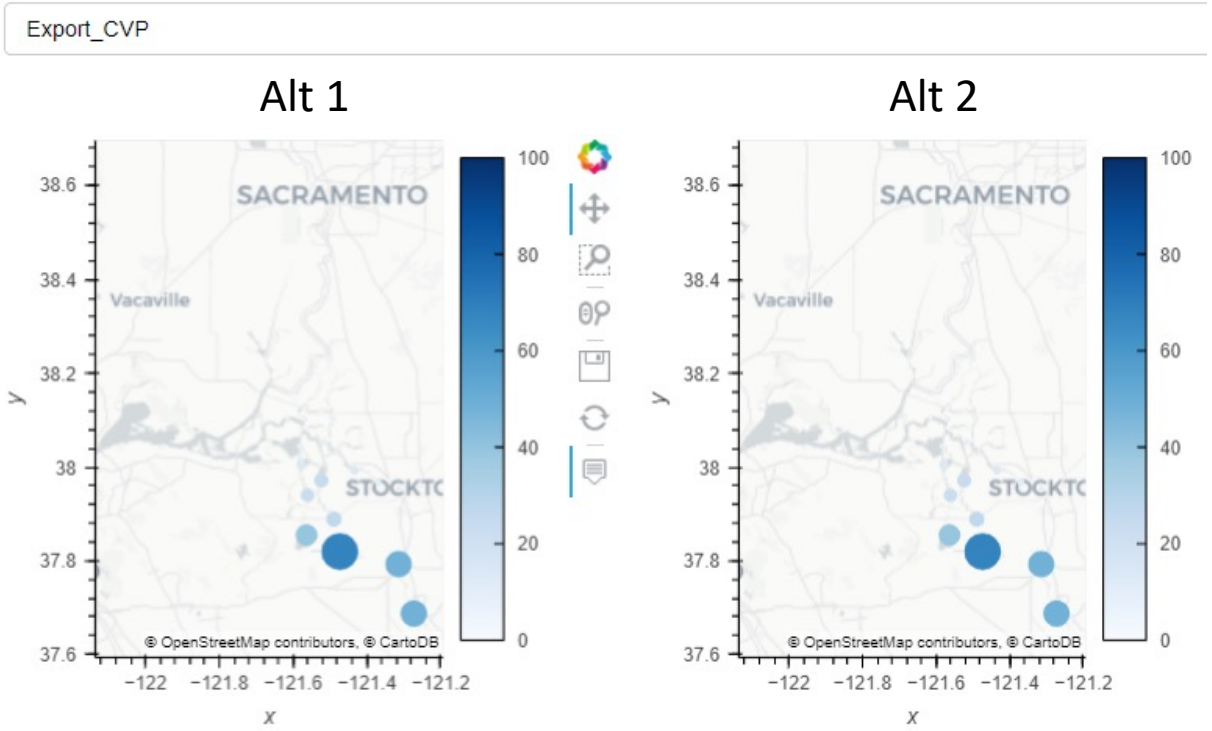
Multiple Insertion Points

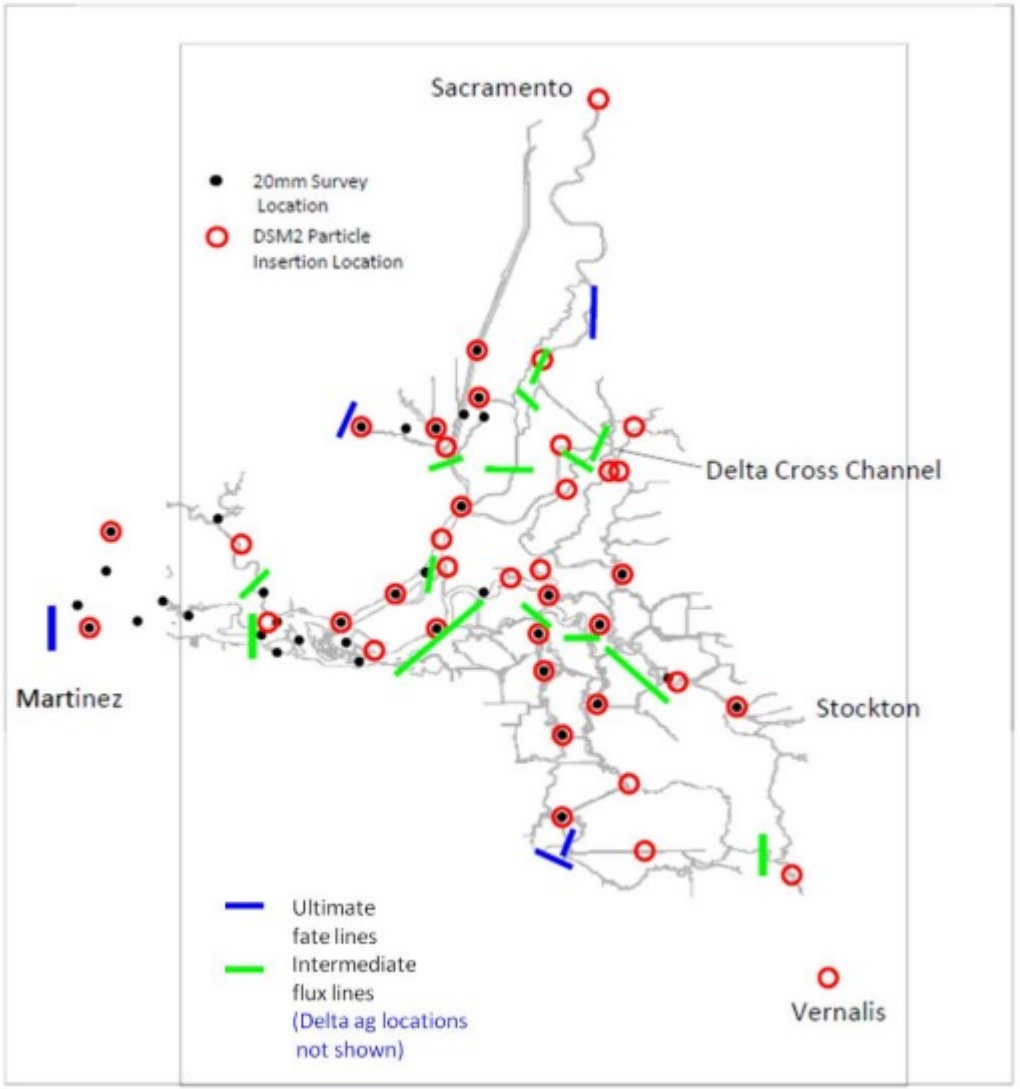
Multiple Hydrology (year and month)

- 1923-2015, Jan-Jun

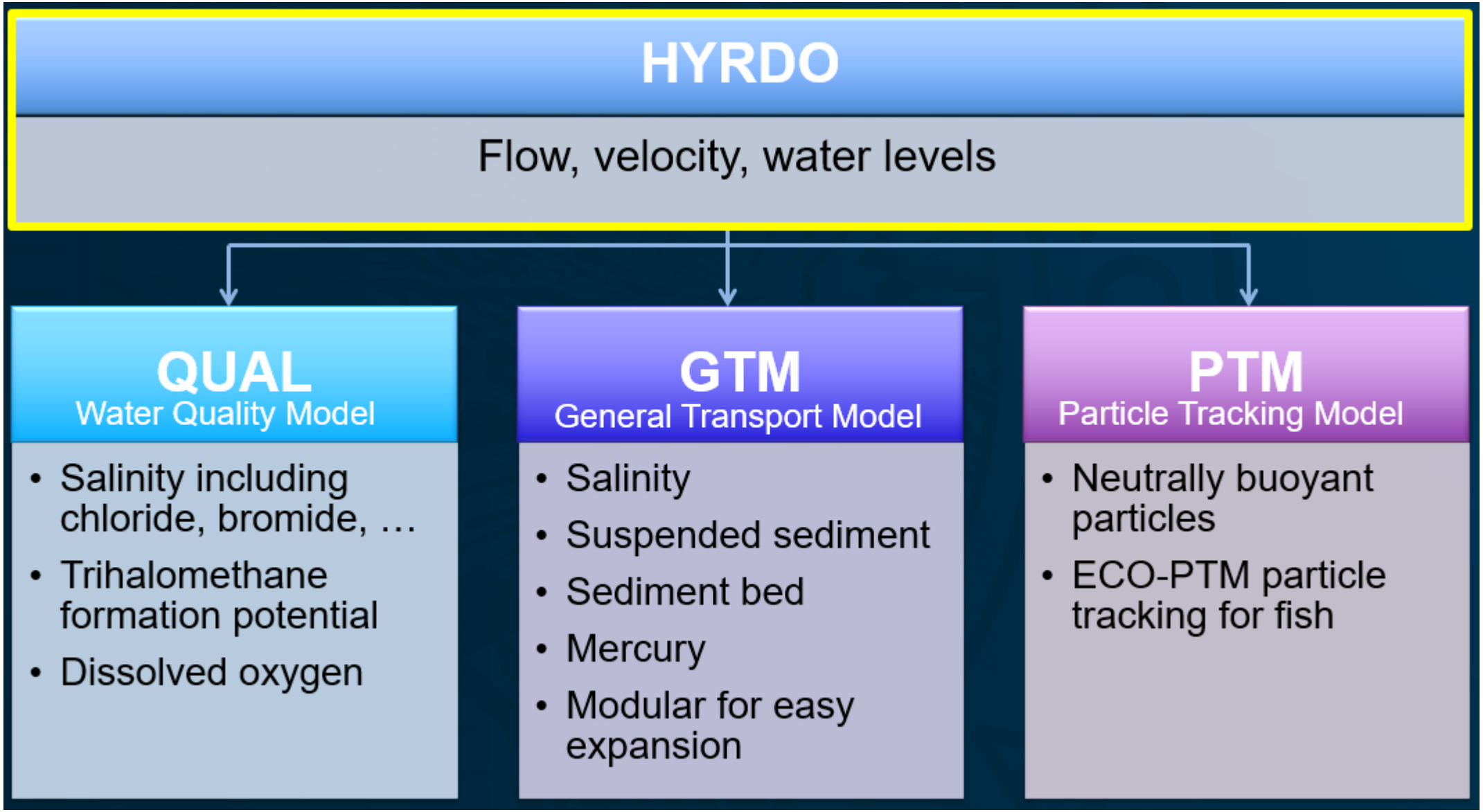
Multiple Types of Particle Behavior

- Neutrally buoyant, surface oriented, eco-ptm



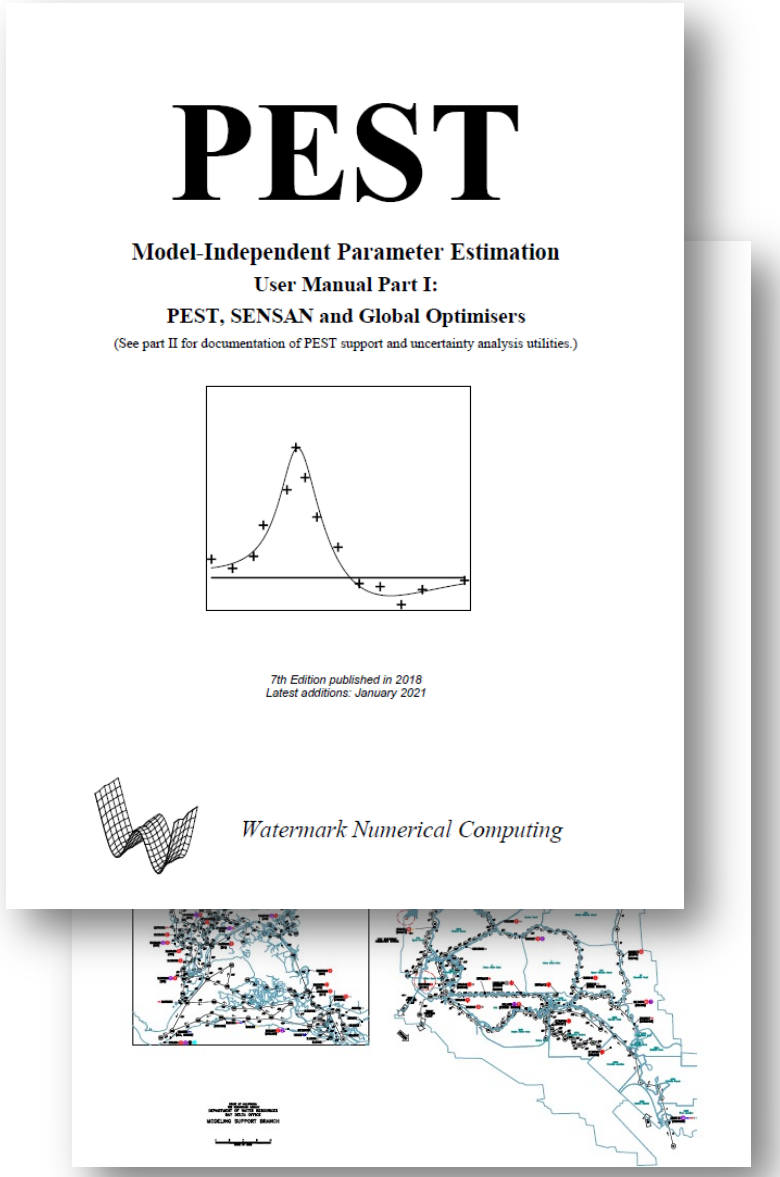


Sub-modules of DSM2



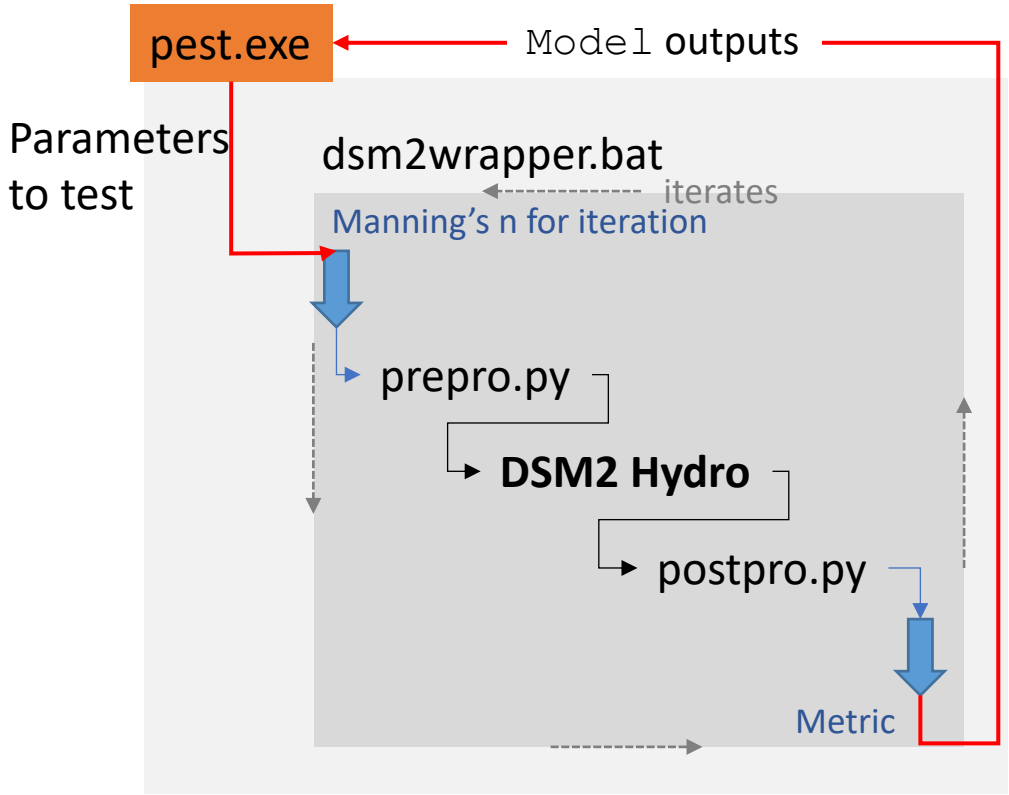
PEST-DSM2 Calibration

- The PEST software packages automates calibration, and calibration-constrained uncertainty analysis of numerical models.
- Use PEST to calculate the optimal set of manning's n for Delta channels by minimizing error between observed and modeled flows and stage.
- This effort will be conducted in tandem with manual calibration.



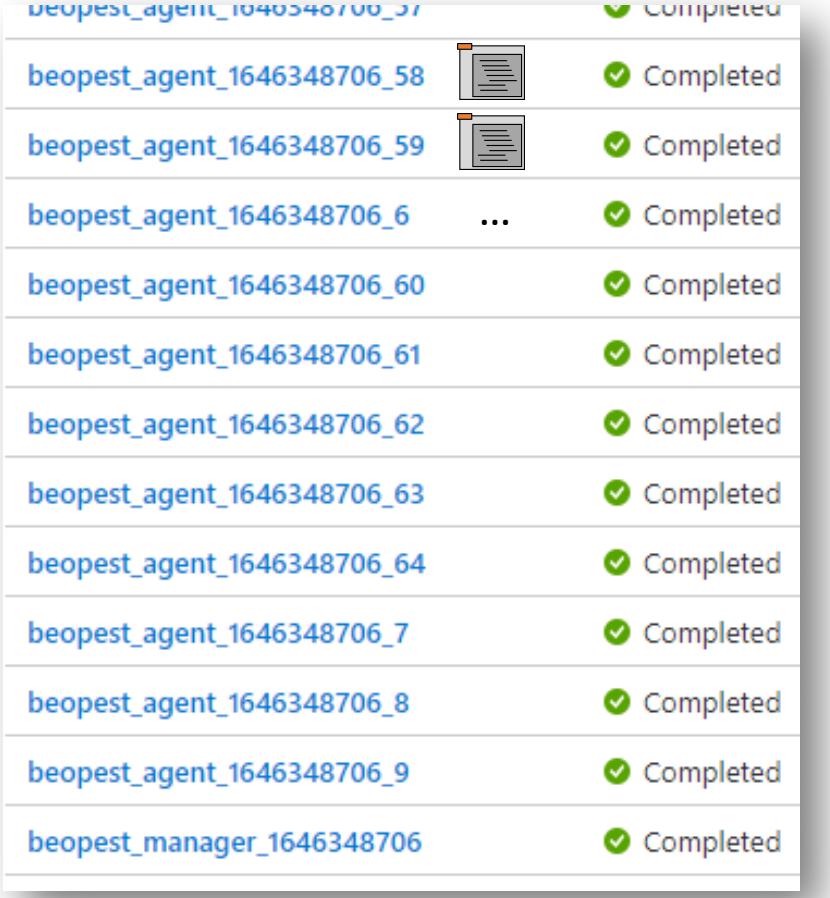
PEST-DSM2 Framework

PEST is model-independent parameter estimation software, so an interface is needed to provide input and output to PEST in the desired format.

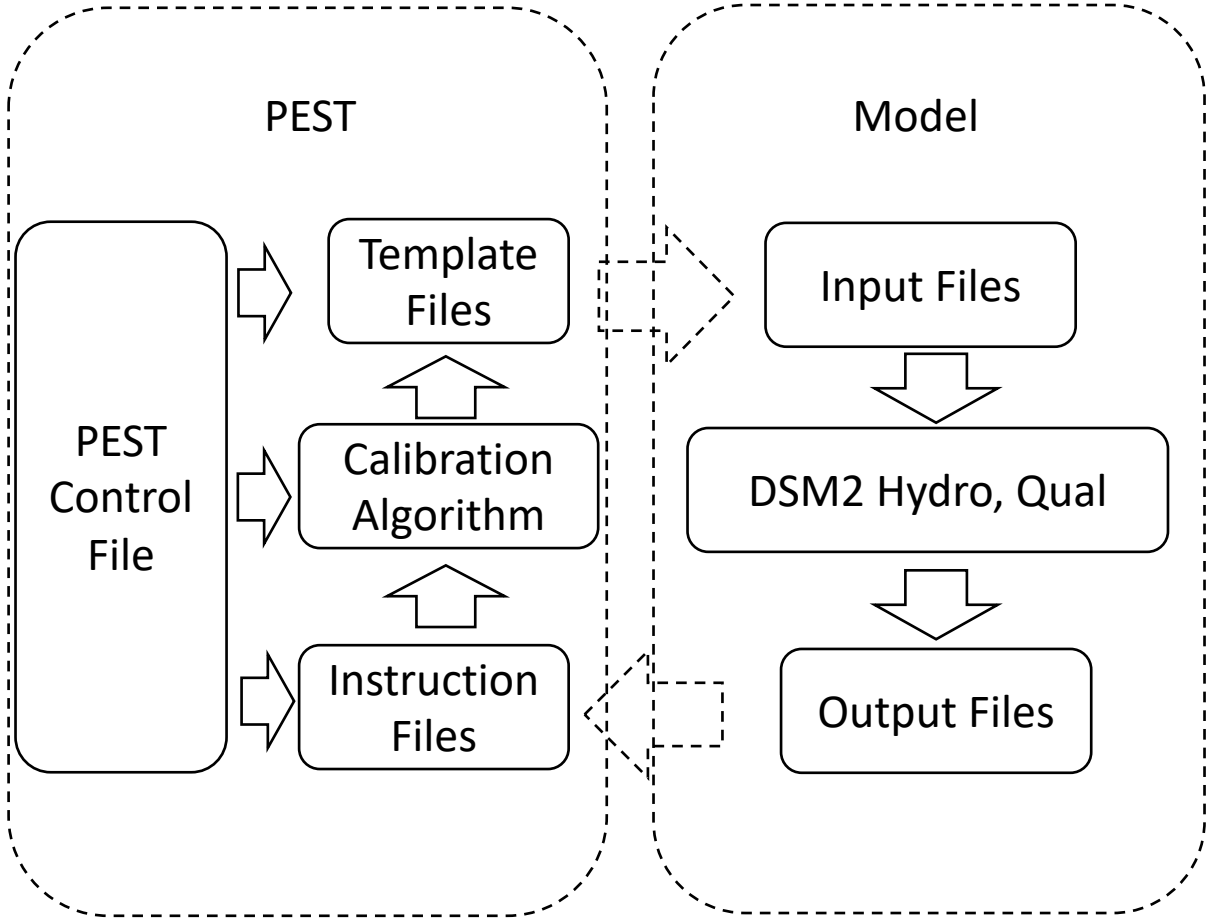


Azure Application

- PEST optimization requires many iterations of DSM2, so leveraging the compute resources of Azure greatly reduces the runtime.
- *n* number of nodes initiated, where *n* is the number of parameters
- Significantly reduced runtimes. For a trial run of 65 parameters and three years:
  - Azure (parallel) run takes 3 hours
  - Local (serial) run takes 40-50 hours



PEST-DSM2


PEST
Model
PEST Control File
Template Files
Calibration Algorithm
Instruction Files
Input Files
DSM2 Hydro, Qual
Output Files

Parameter ESTimation with PEST

- Inverse problem: outputs + inputs -> parameters
- Model-Independent, non-linear
- Weighted least squares residuals sum:

Objective function: $\Phi = \Sigma (h_i - o_i)^2$
Where h = historical observation, and o is model output of DSM2

- Jacobian matrix, finite difference, iterative

$o - o_0 = J (p - p_0)$

$J_{i,j} = \partial o_i / \partial p_j$