Session 4: MPI Parallel-Distributed Computation on Kubernetes

Sunday, August 5, 2018 · 2:30PM - 3:30PM

Christopher Paolini

Assistant Professor of Electrical and Computer Engineering

San Diego State University

PI NSF OAC CC*Storage #1659169, Co-PI NSF OAC CC-NIE #1245312,

Co-PI NSF OAC MRI #0922702, Co-PI NSF OAC CI-TEAM #0753283

PACIFIC RESEARCH PLATFORM

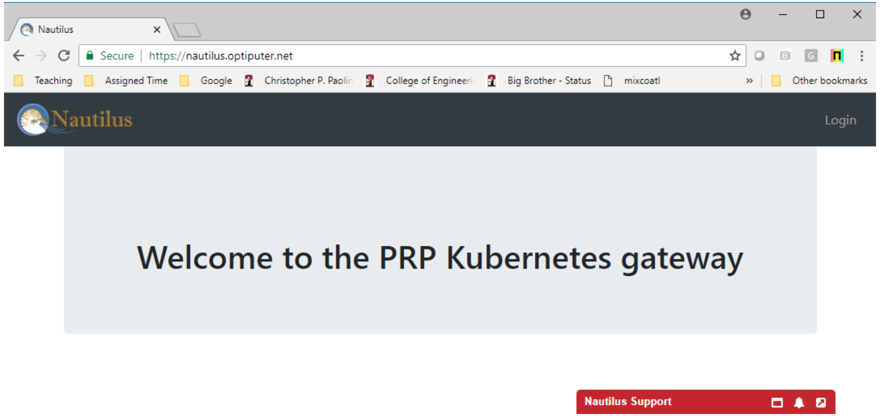
SAN DIEGO STATE UNIVERSITY

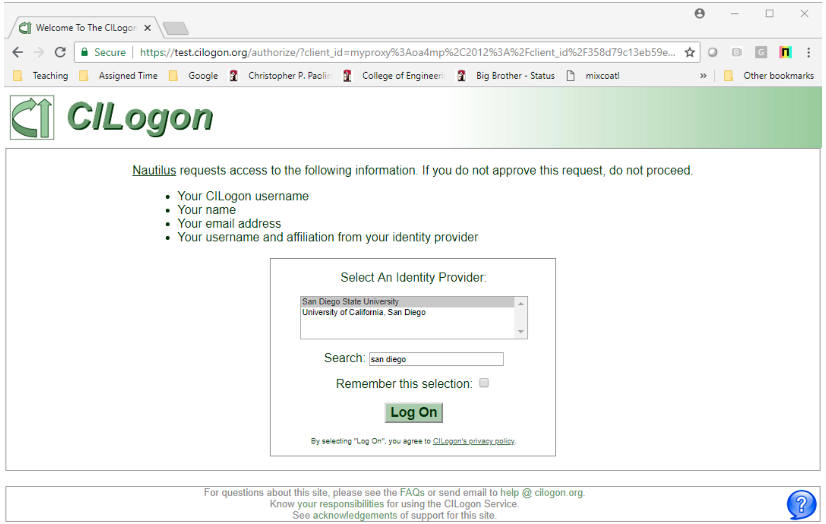
Topics

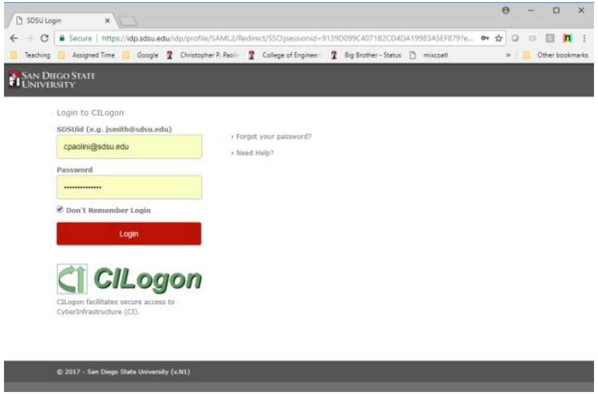
- Configuring and testing an environment for using OpenMPI on K8s
- Benchmarking PRP K8s with LINPACK
- Example PRP science driver: subsurface CO_2 and waste-water injection simulation

PRP Kubernetes Gateway: *kubectl Client Config*

- **Login to https://nautilus.optiputer.net/ to obtain the latest kubectl client config bundle**





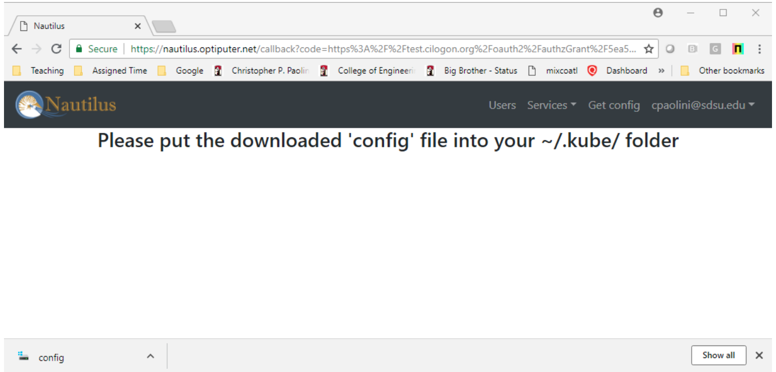


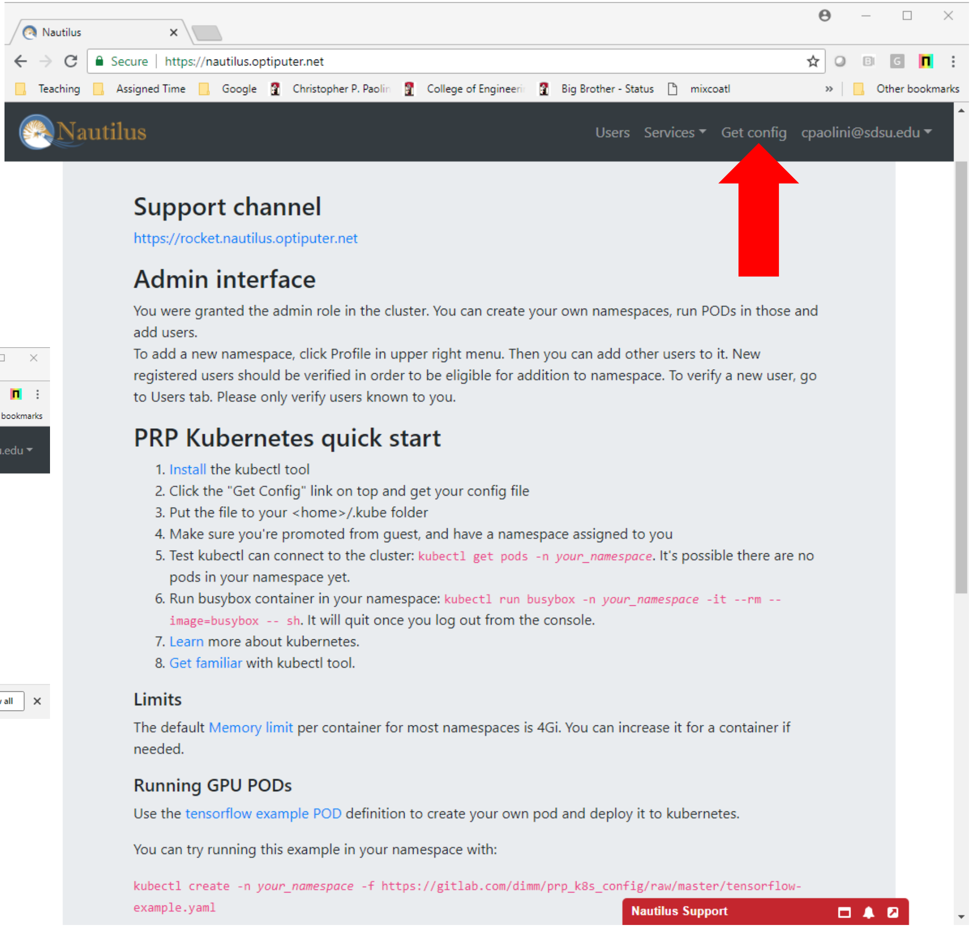
PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

PRP Kubernetes Gateway: *kubectl Client Config cont.*

- **Download kubectl client config bundle via URL https://nautilus.optiputer.net/authConfig and move file to `$HOME/.kube/config` on your station**





PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Using OpenMPI on K8s: *Create Pods*

- **Check the kubectl configuration**

```
[paolini@ps-40g ~]$ kubectl -n sdsu cluster-info
Kubernetes master is running at https://67.58.53.147:6443
```

- **Create cluster of pods using a YAML configuration**

```
[paolini@fiona k8s]$ kubectl create -f subflow.yaml
```

- **Verify all pods are in the *Running* state**

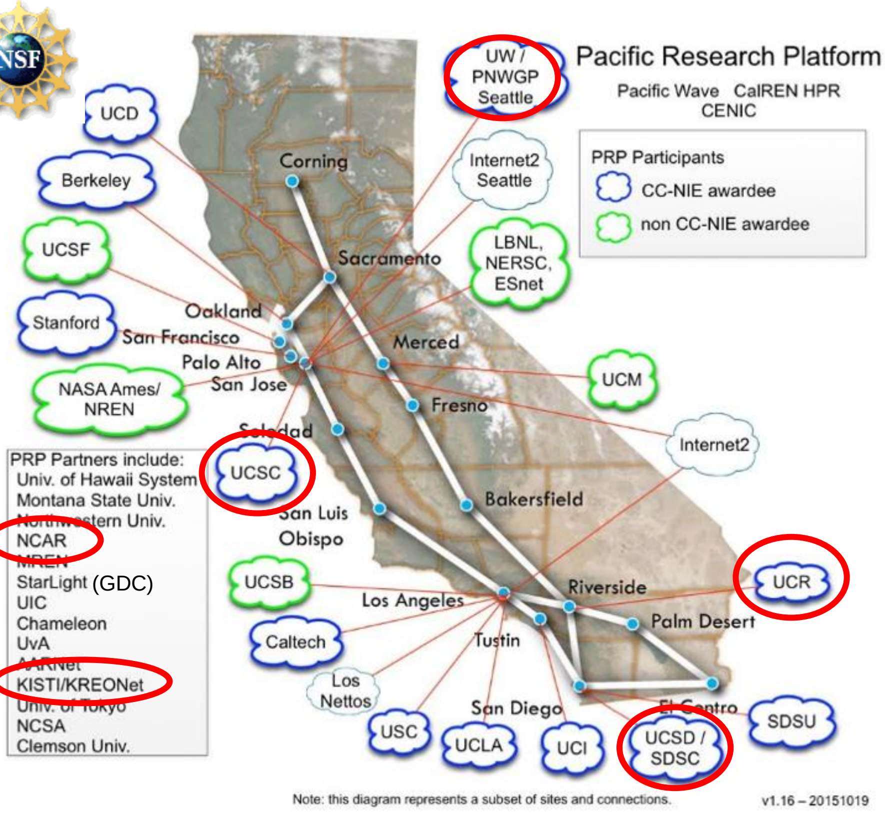
```
[paolini@fiona k8s]$ kubectl get pods -o wide -n sdsu
NAME                          READY     STATUS    RESTARTS   AGE       IP              NODE
subflow-74c57d67d4-bqnc2      1/1       Running   0          3d        10.244.1.151    k8s-nvme-01.sdsc.optiputer.net
subflow-74c57d67d4-gd8sn      1/1       Running   0          3d        10.244.16.45    fiona.cac.washington.edu
subflow-74c57d67d4-jjxl5      1/1       Running   0          3d        10.244.15.182   dtn-main.ucr.edu
subflow-74c57d67d4-jl72q      1/1       Running   0          3d        10.244.11.71    siderea.ucsc.edu
subflow-74c57d67d4-l9n8t      1/1       Running   0          3d        10.244.24.46    dtn2-daejeon.kreonet.net
subflow-74c57d67d4-lj8xw      1/1       Running   0          3d        10.244.10.96    fiona-dtn-1.ucsc.edu
subflow-74c57d67d4-q8mbj      1/1       Running   0          3d        10.244.19.134   fiona.nwsc.ucar.edu
subflow-74c57d67d4-x5j6r      1/1       Running   0          3d        10.244.12.100   k8s-epyc-01.sdsc.optiputer.net
```

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Using OpenMPI on K8s: *Create Pods cont.*

```
[paolini@fiona k8s]$ kubectl get pod -n sdsu -o=custom-columns=NODE:.spec.nodeName
NODE
k8s-nvme-01.sdsc.optiputer.net
fiona.cac.washington.edu
dtn-main.ucr.edu
siderea.ucsc.edu
dtn2-daejeon.kreonet.net
fiona-dtn-1.ucsc.edu
fiona.nwsc.ucar.edu
k8s-epyc-01.sdsc.optiputer.net
```



PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Using OpenMPI on K8s: *YAML configuration*

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: subflow
  namespace: sdsu
spec:
  replicas: 8
  template:
    metadata:
      labels:
        k8s-app: subflow
    spec:
      containers:
      - name: subflow
        image: phusion/baseimage:0.9.19
        imagePullPolicy: IfNotPresent
        args: ["sleep", "infinity"]
        resources:
          limits:
            memory: "48Gi"
          requests:
            memory: "32Gi"
        volumeMounts:
        - name: nfs
          mountPath: /nfs
      volumes:
      - name: nfs
        nfs:
          server: 10.109.158.238
          path: "/"
```

Request each container to have 32GiB of allocatable memory, with an upper limit of 48GiB

Mount existing NFS (Network File System) share (nfs://10.109.158.238/) on mount-point /nfs in pod.

Persistent location of git source clone and build for k8s pods

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Using OpenMPI on K8s: *Initialize Cluster*

```
[paolini@fiona k8s]$ ./initcluster.sh
[paolini@fiona k8s]$ cat initcluster.sh
                #!/bin/bash
                hosts=""
                n=0
                for host in `kubectl get pod -n sdsu -o=custom-columns=IP:status.podIP,NAME:.metadata.name | grep subflow`
                do
                    hosts="$hosts $host"
                    if [ $n -eq 1 ];
                    then
                        cores=`kubectl exec -it $host -n sdsu -- bash -c "grep -c ^processor /proc/cpuinfo"`
                        cores=`tr -dc '[[:print:]]' <<< " $cores"`
                        hosts+=$cores
                        n=0
                    else
                        n=1
                    fi
                done
                pods=`kubectl get pod -n sdsu -o=custom-columns=NAME:.metadata.name|sed -e '/NAME/d'|egrep '^subflow-'`
                for pod in $pods
                do
                    kubectl exec -it $pod -n sdsu -- bash -c "/nfs/subflow/k8s/setup.sh $hosts"
                done
                for pod in $pods
                do
                    kubectl exec -it $pod -n sdsu -- bash -c "/nfs/subflow/k8s/updatekeys.sh"
                done
```

Build string array of 3-tuples (IP Address, hostname, processor count) needed for MPI *hostfile*

Invoke setup.sh on each pod with string array as augment list

setup.sh starts ssh server on each pod

Gather ssh public keys of all pods for genenerating /etc/ssh/ssh_known_hosts (requires sshd be running on all pods)

PACIFIC RESEARCH PLATFORM

Using OpenMPI on K8s: *Initialize Pod*

```
[paolini@fiona k8s]$ cat setup.sh
#!/bin/bash

apt-get --quiet update;
apt-get --quiet -y install screen emacs git python mlocate openmpi-bin openmpi-doc libopenmpi-dev make
valgrind g++ m4 gfortran liblapacke-dev libnetcdf-dev iputils-ping openssh-server cmake mesa-utils-extra

if [ ! -f /etc/hosts.DIST ]
then
    cp /etc/hosts /etc/hosts.DIST
fi

if ! id subflow >/dev/null 2>&1
then
    groupadd subflow; useradd -g subflow -d /home/subflow -m -c 'Subflow Execution' -s /bin/bash subflow
fi

if [ ! -f /root/.ssh/id_rsa.pub ]
then
    echo -e 'y\n'|/usr/bin/ssh-keygen -f /root/.ssh/id_rsa -t rsa -N ''
fi
```

Install required packages for running MPI programs and building target

Backup original /etc/hosts file

Create fictitious user account if you don't want to run MPI processes as root

Generate protocol version 2 ssh authentication keys for passwordless, public key authentication between pods

PACIFIC RESEARCH PLATFORM

UNIVERSITY

Using OpenMPI on K8s: *Initialize Pod cont.*

```
c=`grep -c "HostbasedAuthentication yes" /etc/ssh/ssh_config`
if [ "$c" -eq "0" ]
then
    echo "Enable SSH client HostbasedAuthentication"
    echo "HostbasedAuthentication yes" >> /etc/ssh/ssh_config
fi
c=`grep -c "HostbasedAuthentication yes" /etc/ssh/sshd_config`
if [ "$c" -eq "0" ]
then
    echo "Enable SSH server HostbasedAuthentication"
    echo "HostbasedAuthentication yes" >> /etc/ssh/sshd_config
fi

c=`grep -c "IgnoreRhosts no" /etc/ssh/sshd_config`
if [ "$c" -eq "0" ]
then
    echo "Enable SSH server rhosts authentication"
    echo "IgnoreRhosts no" >> /etc/ssh/sshd_config
fi

c=`grep -c "HostbasedUsesNameFromPacketOnly yes" /etc/ssh/sshd_config`
if [ "$c" -eq "0" ]
then
    echo "Enable SSH server HostbasedUsesNameFromPacketOnly"
    echo "HostbasedUsesNameFromPacketOnly yes" >> /etc/ssh/sshd_config
fi
```

Permit user authentication between pods listed in /etc/ssh/shosts.equiv (enable in client and server config files)

Enable authentication for root user, based on names of (trusted) pods in `/root/.rhosts`

Configure sshd to accept the hostname information provided in the connection itself, rather than use DNS resolution

Using OpenMPI on K8s: *Initialize Pod cont.*

```
c=`grep -c "^PermitRootLogin yes" /etc/ssh/sshd_config`
if [ "$c" -eq "0" ]
then
    echo "Enable SSH server root login"
    echo "PermitRootLogin yes" >> /etc/ssh/sshd_config
fi

cp /etc/hosts.DIST /etc/hosts
rm -f /etc/ssh/shosts.equiv /etc/mpihostfile

while [[ $# -gt 0 ]]
do
    echo "$1 $2" >> /etc/hosts
    echo "$2" >> /etc/ssh/shosts.equiv
    echo "$2 slots=$3 max-slots=$3" >> /etc/mpihostfile
    shift
    shift
    shift
done
cp /etc/ssh/shosts.equiv /root/.rhosts
/etc/init.d/ssh stop
/etc/init.d/ssh start
```

Permit root authentication between pods to allow MPI processes to run as root

Rebuild /etc/hosts, /etc/ssh/shosts.equiv, and /etc/mpihostfile using string array argument (IP Address, hostname, processor count)

Restart ssh server on pod to enable updated configuration

Using OpenMPI on K8s: *Initialize Pod cont.*

```
[paolini@fiona k8s]$ cat updatekeys.sh
#!/bin/bash

rm -f /etc/ssh/ssh_known_hosts

for host in `cat /etc/ssh/shosts.equiv`
do
    ssh-keyscan -t rsa $host >> /etc/ssh/ssh_known_hosts
    cp /etc/ssh/ssh_known_hosts /root/.ssh/known_hosts
done
```

Each host gathers the public ssh host keys of all hosts

Using OpenMPI on K8s: *Using a Custom Image*

```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: subflow
  namespace: sdsu
spec:
  replicas: 8
  template:
    metadata:
      labels:
        k8s-app: subflow
    spec:
      containers:
      - name: subflow
        image: dimm0/subflow:latest
        imagePullPolicy: IfNotPresent
        args: ["sleep", "infinity"]
        resources:
          limits:
            memory: "48Gi"
          requests:
            memory: "32Gi"
        volumeMounts:
        - name: nfs
          mountPath: /nfs
      volumes:
      - name: nfs
        nfs:
          server: 10.109.158.238
          path: "/"
```

```
[paolini@fiona image]$ cat Dockerfile
FROM phusion/baseimage:0.9.19

MAINTAINER Dmitry Mishin <dmishin@sdsc.edu>

RUN apt-get -y update && \
    apt-get -y install screen emacs git python mlocate openmpi-bin
openmpi-doc libopenmpi-dev make valgrind g++ m4 gfortran liblapacke-dev
libnetcdf-dev iputils-ping openssh-server cmake mesa-utils-extra

[root@fiona image]# docker pull dimm0/subflow
Using default tag: latest
latest: Pulling from dimm0/subflow
f069f1d21059: Pull complete
ecbeec5633cf: Pull complete
ea6f18256d63: Pull complete
54bde7b02897: Pull complete
a3ed95caeb02: Pull complete
ce9e695a6234: Pull complete
346026b9659b: Pull complete
d8a2ef8e4be5: Pull complete
Digest:
sha256:eb39c3151cd0ed776f6188949bc7f93a64db2eb04ab4391591c2670e0c821806
Status: Downloaded newer image for dimm0/subflow:latest
```

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Using OpenMPI on K8s: *Using a Custom Container*

```
[root@fiona image]# docker images
REPOSITORY                              TAG        IMAGE ID       CREATED         SIZE
rocketchat/rocket.chat                  latest     e98d34f3e7ab   16 hours ago    1.38GB
dimm0/subflow                           latest     6aab3414d3be   44 hours ago    1.17GB
us.gcr.io/prp-k8s/nautilus-portal       latest     e5f31849537f   7 days ago      38.7MB
us.gcr.io/prp-k8s/nautilus-portal       <none>     1c868229707b   7 days ago      38.7MB
us.gcr.io/prp-k8s/perfsonar_testpoint_h latest     2c3315ea9b6d   10 days ago     818MB
us.gcr.io/prp-k8s/perfsonar_testpoint_h <none>     3f220a2c3190   10 days ago     818MB
rocketchat/rocket.chat                  <none>     587a250b077b   12 days ago     1.37GB
us.gcr.io/prp-k8s/meshconfig            latest     ee2c46848fde   2 weeks ago     34.2MB
sameersbn/postgresql                    latest     3c0142eb3992   2 weeks ago     204MB
robcurrie/tensorflow-gpu                latest     90592bb75ee1   4 weeks ago     4.28GB
nvidia/k8s-device-plugin                1.10       a7c090961376   5 weeks ago     63.1MB
gcr.io/google_containers/kube-proxy-amd64 v1.11.0  1d3d7afd77d1   5 weeks ago     97.8MB
us.gcr.io/prp-k8s/prp-tuner             latest     62dbae4fe2e1   2 months ago    52.2MB
gcr.io/runconduit/proxy                 v0.4.1     fd47f6c1933d   3 months ago    111MB
gcr.io/runconduit/proxy-init            v0.4.1     60eb863cee15   3 months ago    105MB
quay.io/calico/node                     v3.1.1     d94b64ac210d   3 months ago    248MB
quay.io/calico/cni                      v3.1.1     482f47df27e2   3 months ago    68.8MB
us.gcr.io/prp-k8s/perfsonar_testpoint_h <none>     2fafb9b60974   3 months ago    784MB
us.gcr.io/prp-k8s/perfsonar_testpoint   latest     fec1df3352a3   3 months ago    784MB
quay.io/coreos/kube-rbac-proxy          v0.3.0     543e2018dcac   4 months ago    40.2MB
rook/rook                               v0.7.1     ee1353c748c0   4 months ago    435MB
traefik                                 1.5.3      8c72b944d569   5 months ago    49.6MB
k8s.gcr.io/pause                        3.1        da86e6ba6ca1   7 months ago    742kB
quay.io/prometheus/node-exporter        v0.15.2    ff5ecdcfc4a2   8 months ago    22.8MB
```

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Using OpenMPI on K8s: *Using a Custom Container*

```
[paolini@fiona k8s]$ kubectl create -f comet.yaml
deployment.apps "subflow-comet" created

[paolini@fiona k8s]$ ./initcluster.sh sdsu-comet
Generating public/private rsa key pair.
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:cQvHueOgYQI/x0zWRSyEwNyT6lgNNEmcFTNMWZ6pPdM root@subflow-comet-6788dc75d5-778bz
The key's randomart image is:
+---[RSA 2048]----+
|   *=O*Booo      |
|    B.B=.=..     |
|  .  +o.B.=      |
|   oo=.o * o     |
|   ++ B S E      |
|  . .= o = .     |
|      .   .      |
|                 |
|                 |
+----[SHA256]-----+
Enable SSH client HostbasedAuthentication
Enable SSH server HostbasedAuthentication
Enable SSH server rhosts authentication
Enable SSH server HostbasedUsesNameFromPacketOnly
Enable SSH server root login
 * Stopping OpenBSD Secure Shell server sshd                                                         [ OK ]
 * Starting OpenBSD Secure Shell server sshd                                                         [ OK ]
.
.
.
# subflow-comet-6788dc75d5-778bz:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-p8z2g:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-pgw78:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-q6wvc:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-778bz:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-p8z2g:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-pgw78:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-q6wvc:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-778bz:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-p8z2g:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-pgw78:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-q6wvc:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-778bz:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-p8z2g:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-pgw78:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
# subflow-comet-6788dc75d5-q6wvc:22 SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.4
```

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Using OpenMPI on K8s: *Initialize Pod cont.*

```
root@subflow-74c57d67d4-bqnc2:/# cat /etc/mpihostfile
subflow-74c57d67d4-bqnc2 slots=16 max-slots=16
subflow-74c57d67d4-gd8sn slots=12 max-slots=12
subflow-74c57d67d4-jjxl5 slots=12 max-slots=12
subflow-74c57d67d4-jl72q slots=8 max-slots=8
subflow-74c57d67d4-l9n8t slots=12 max-slots=12
subflow-74c57d67d4-lj8xw slots=12 max-slots=12
subflow-74c57d67d4-q8mbj slots=12 max-slots=12
subflow-74c57d67d4-x5j6r slots=96 max-slots=96
```

Number of *slots* is the number of processors available to the pod

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Using OpenMPI on K8s: *MPI Ring Test*

```
[paolini@fiona k8s]$ more RUN
#!/bin/bash

KUBE_NAMESPACE=sdsu
export KUBE_NAMESPACE

MPI_CLUSTER_NAME=subflow
export MPI_CLUSTER_NAME

case "$1" in
    ring)
        kubectl -n $KUBE_NAMESPACE exec -it $MPI_CLUSTER_NAME-74c57d67d4-bqnc2 -- mpirun --allow-run-as-root \
            --hostfile /etc/mpihostfile \
            --mca btl tcp,self \
            -n 8 -npernode 1 --bind-to core \
            /nfs/subflow/k8s/ring
        ;;
```

Direct Open MPI to use TCP-based communications over IP interfaces. Modular Component Architecture (MCA) parameter: byte transfer layer (BTL)

Bind each MPI process to a core.

Using OpenMPI on K8s: *MPI Ring Test cont.*

```
[paolini@fiona k8s]$ ./RUN ring
process 0 of 8 (on subflow-74c57d67d4-bqnc2)
process 2 of 8 (on subflow-74c57d67d4-jjxl5)
process 1 of 8 (on subflow-74c57d67d4-gd8sn)
process 6 of 8 (on subflow-74c57d67d4-q8mbj)
process 5 of 8 (on subflow-74c57d67d4-lj8xw)
process 3 of 8 (on subflow-74c57d67d4-jl72q)
process 7 of 8 (on subflow-74c57d67d4-x5j6r)
process 4 of 8 (on subflow-74c57d67d4-l9n8t)
token= 8192.000001, matches TIMES_AROUND*nprocs (things look ok).
doing long send...
...passed long send test.
[paolini@fiona k8s]$
```

Pass a (float) token around the "ring" of 8 processes 2^{10} times. Each process increments the token by +1 when the token is received from its "left" process neighbor, prior to sending the token to its "right" neighbor.

Benchmarking PRP K8s with LINPACK

- *HPL LINPACK Benchmark*: commonly used performance metric to evaluate a cluster's performance
- Project URL: https://www.top500.org/project/linpack/
- FAQs:

  http://www.netlib.org/utk/people/JackDongarra/faq-linpack.html

  http://www.netlib.org/benchmark/hpl/faqs.html
- Download URL: http://www.netlib.org/benchmark/hpl/
- Repeatedly solves a linear system of N equations using LU decomposition with partial row pivoting, giving a performance metric R.
- We want to identify R_{max} for a particular N (with respect to a chosen number of processes and other parameters).

Benchmarking PRP K8s with LINPACK *cont.*

- Solve a linear system $Ax = b$ of order N
- Compute LU factorization of A with row partial pivoting (operations stored in P) of the $n \times (n+1) = N$ coefficient matrix $[A\ b]$
- [A b] = [[L,U] y]
- $PA=LU \rightarrow LUx=Pb$
- Solve $Ly=Pb$ for y, using forward substitution, then solve $Ux=y$ for x using backward substitution
- $O(\frac{2}{3}n^3)$ FLOP
- $n \times (n+1)$ coefficient matrix logically partitioned into $NB \times NB$ blocks, that are cyclically assigned onto a $P \times Q$ process grid.

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Benchmarking PRP K8s with LINPACK: *Block LU*

- Partition *A* into blocks (sub-matrices) of size *NBxNB* and distribute to each process:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

- Partitioning yields the following *block equations* for partitioned LU factorization:

$$\begin{bmatrix} A_{11} = L_{11}U_{11} & A_{12} = L_{11}U_{12} & A_{13} = L_{11}U_{13} \\ A_{21} = L_{21}U_{11} & A_{22} = L_{21}U_{12} + L_{22}U_{22} & A_{23} = L_{21}U_{13} + L_{22}U_{23} \\ A_{31} = L_{31}U_{11} & A_{32} = L_{31}U_{12} + L_{32}U_{22} & A_{33} = L_{31}U_{13} + L_{32}U_{23} + L_{33}U_{33} \end{bmatrix}$$

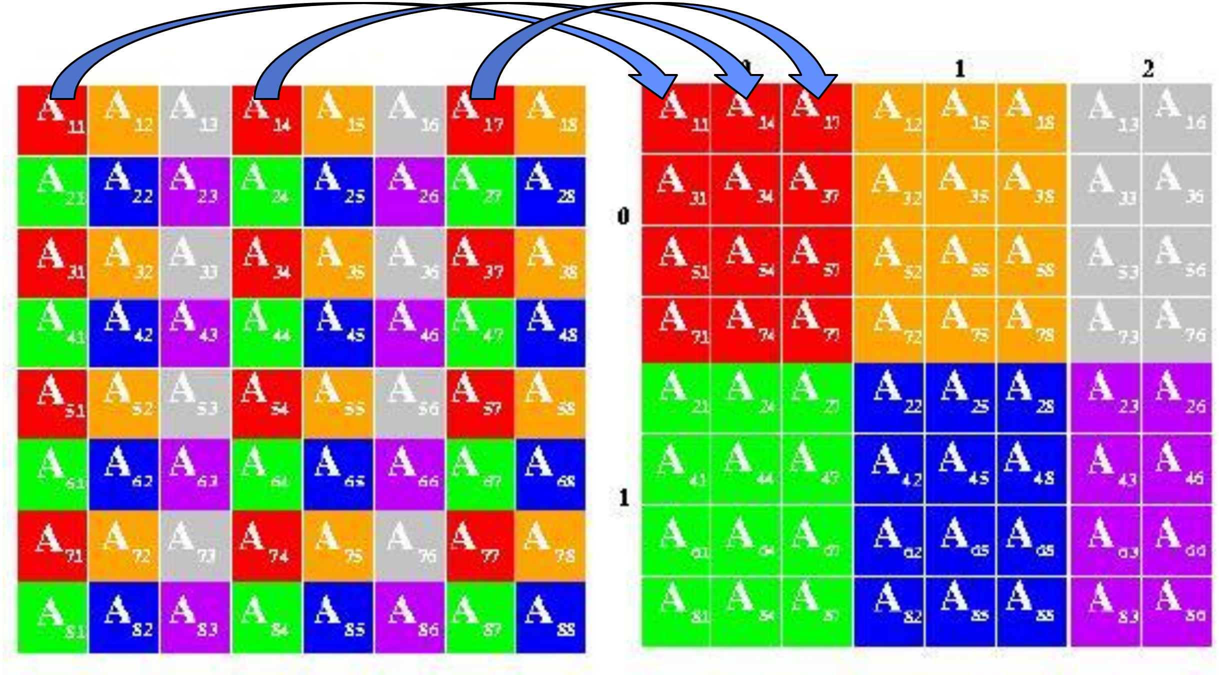
- Block size, *NB*, will affect performance. We can experiment and see how changing the block size yields different performance metrics *R* (Gflops) on the PRP K8s cluster.

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Benchmarking PRP K8s with LINPACK: *Processor Grid (PxQ)*

- Example: *P x Q = 2 x 3;* each color corresponds to an MPI process assignment.



PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

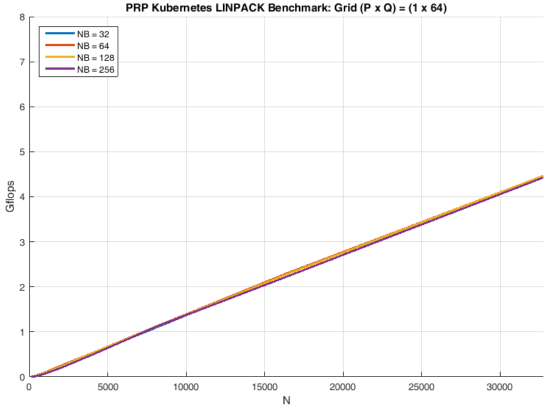
Benchmarking PRP K8s with LINPACK: *Launching HPL*

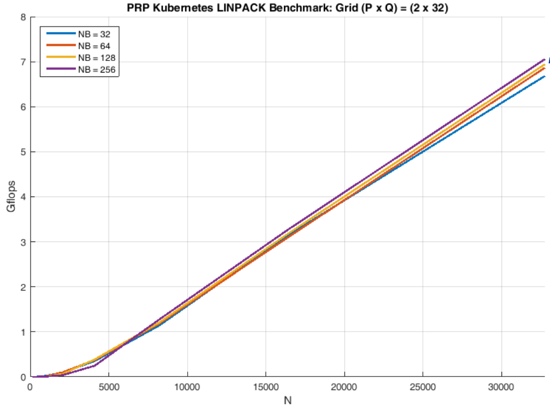
```
[paolini@fiona k8s]$ more RUN
#!/bin/bash

KUBE_NAMESPACE=sdsu
export KUBE_NAMESPACE

MPI_CLUSTER_NAME=subflow
export MPI_CLUSTER_NAME

case "$1" in
    .
    .
    .
    hpl)
        kubectl -n $KUBE_NAMESPACE exec -it $MPI_CLUSTER_NAME-74c57d67d4-bqnc2 -- mpirun --allow-run-as-root \
            --hostfile /etc/mpihostfile \
            --mca btl tcp,self \
            --display-map \
            -n 64 -npernode 8 --bind-to core:overload-allowed \
            sh -c 'cd /nfs/hpl-2.2/bin/k8s; ./xhpl > xhpl.out 2>&1'
        ;;
```

Max is 8 nodes with 8 processes per node (64)

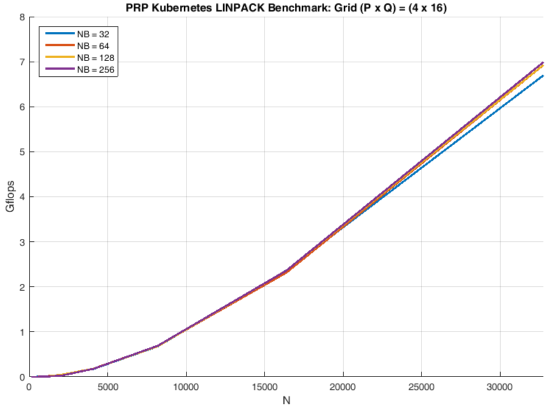
HPL application is `xhpl`

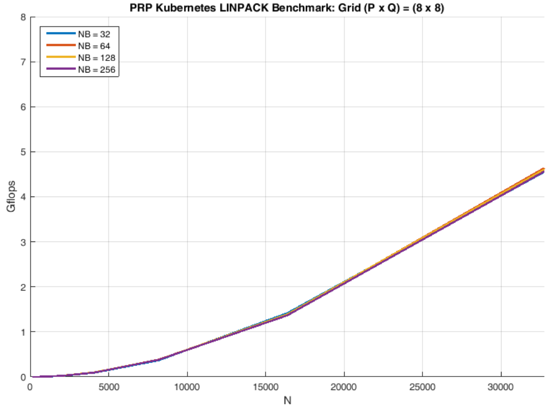
Benchmarking PRP K8s with LINPACK: *Results*





- **Blocksize = 256**
- **(PxQ)=(2x32)**
- **Panel fact.: Crout**
- **Recursive Panel fact.: Crout**
- **7.056 Gflops**



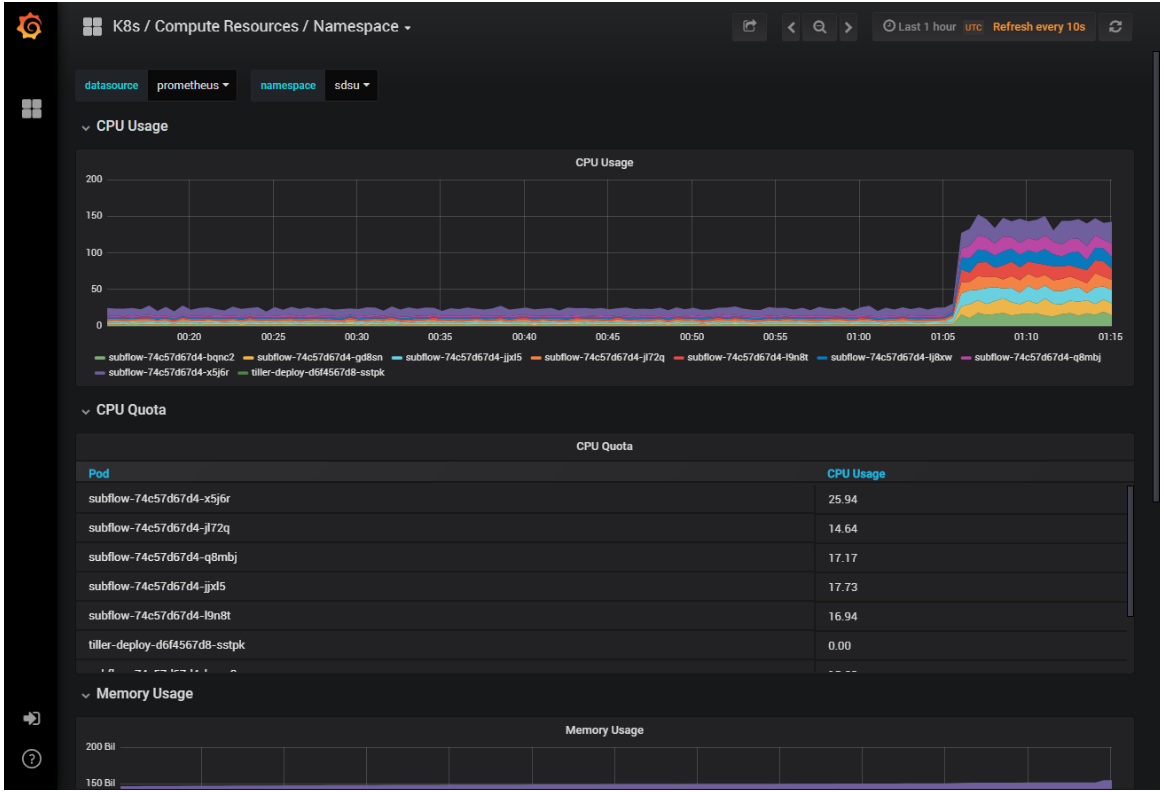


PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Benchmarking PRP K8s with LINPACK: *Grafana Monitoring*

- **https://grafana.nautilus.optiputer.net/**
- **https://grafana.nautilus.optiputer.net/d/4r_GftHmz/k8s-compute-resources-namespace?refresh=10s&orgId=1&var-datasource=prometheus&var-namespace=sdsu**



PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Benchmarking PRP K8s with LINPACK: *HPL.dat*

```
root@subflow-74c57d67d4-bqnc2:/nfs/hpl-2.2/bin/k8s# cat HPL.dat
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out      output file name (if any)
6            device out (6=stdout,7=stderr,file)
9           # of problems sizes (N)
128 256 512 1024 2048 4096 8192 16384 32768 Ns
4            # of NBs
32 64 128 256      NBs
0            PMAP process mapping (0=Row-,1=Column-major)
4            # of process grids (P x Q)
1  2  4  8   Ps
64 32 16 8   Qs
16.0         threshold
1            # of panel fact
1        PFACTs (0=left, 1=Crout, 2=Right)
2            # of recursive stopping criterium
2 4          NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
1        RFACTs (0=left, 1=Crout, 2=Right)
1            # of broadcast
0            BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
1            # of lookahead depth
0            DEPTHs (>=0)
2            SWAP (0=bin-exch,1=long,2=mix)
64           swapping threshold
0            L1 in (0=transposed,1=no-transposed) form
0            U  in (0=transposed,1=no-transposed) form
1            Equilibration (0=no,1=yes)
8            memory alignment in double (> 0)
```

11 values of *N*, from 2^7 to 2^{15} such that N is NB aligned

Test with 4 block sizes *NB*.
"good" block sizes are almost always in the [32 .. 256] interval. -- Jack Dongarra

Test with 4 processor grid configurations with Q ≥ P and PQ = 64.

Crout's algorithm typically performs better than left- and right- looking panel methods

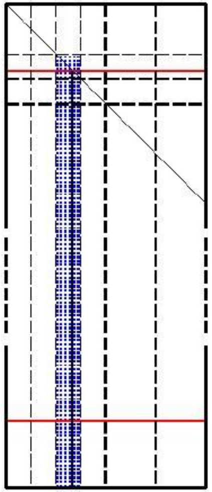
PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Benchmarking PRP K8s with LINPACK: *Panel Factorization*

```
1            PFACTs (0=left, 1=Crout, 2=Right)
1            # of recursive stopping criterium
4            NBMINs (>= 1)
1            # of panels in recursion
2            NDIVs
1            # of recursive panel fact.
1            RFACTs (0=left, 1=Crout, 2=Right)
```

- Crout matrix decomposition: returns lower triangular matrix and a *unit* upper triangular matrix (Prescott Durand Crout).
- Recursively divides each panel into NDIV subpanels at each step.
- Recursion stops when the current panel is made of less than or equal to NBMIN columns.
- RFACT: method used for recursive sub- panel factorization.
- PFACT: method used after recursive factorization completes.
- Example: NDIV=2, NBMIN=4
- Right- and left-looking factorization algorithms: http://www.netlib.org/utk/papers/outofcore/node3.html



PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Benchmarking PRP K8s with LINPACK: *Right Looking*

- Suppose we have factored the blocks in the first row and column of *A*:

$$A = LU = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} L_{11} & & \\ L_{21} & & \\ L_{31} & & \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ & & \\ & & \end{bmatrix}$$

Factored blocks

Remaining block submatrix to factor

- Since the remaining green block submatrix is independent of the yellow completed region, the green region can be factored in isolation:

$$\begin{bmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{bmatrix} \Rightarrow \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = P \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = P \begin{bmatrix} L_{11}U_{11} & L_{11}U_{12} \\ L_{21}U_{11} & L_{21}U_{12} + L_{22}U_{22} \end{bmatrix}$$

- Factoring the first column of the block submatrix yields L_{11}, L_{21}, and U_{11} . We can solve for U_{12}:

$$A_{12} = L_{11}U_{12} \rightarrow U_{12} = L_{11}^{-1}A_{12}$$

- Finally, we update block A_{22} and repeat this process:

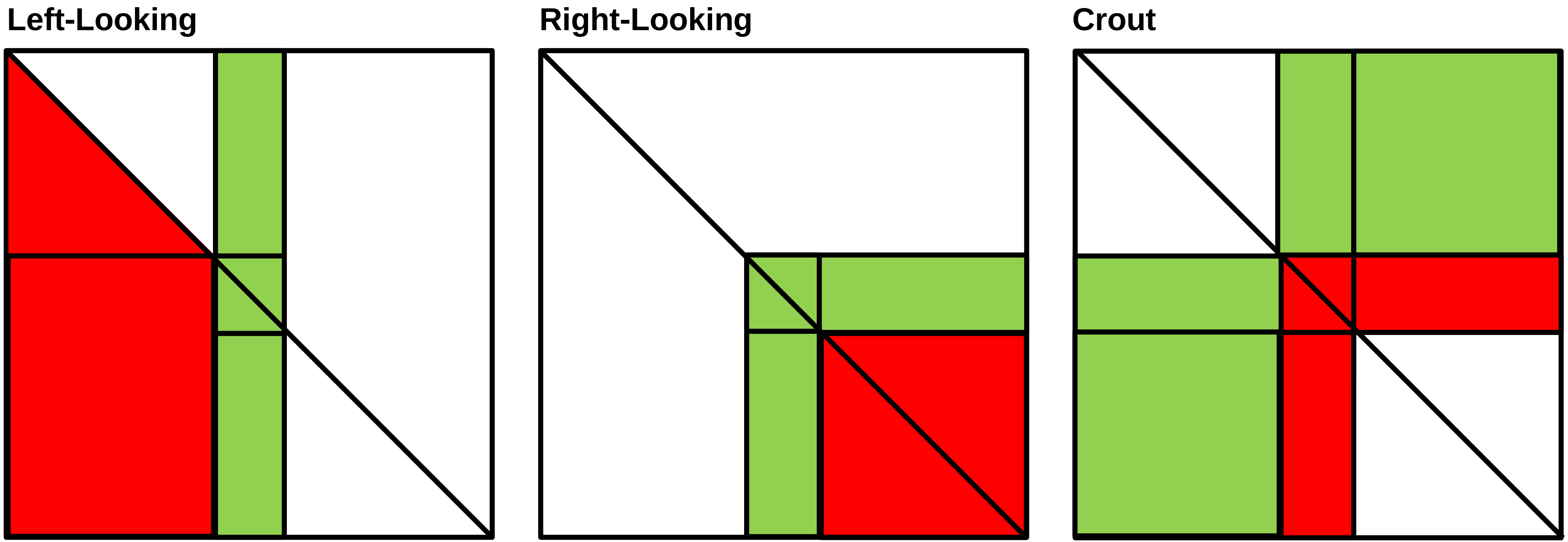
$$A_{22} = L_{21}U_{12} + L_{22}U_{22} \Rightarrow L_{22}U_{22} = \underbrace{A_{22} - L_{21}U_{12}}_{\text{known}}$$

  - Perform LU factorization to solve for L_{22} *and* U_{22}

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Benchmarking PRP K8s with LINPACK: *Memory Access*



Unfactored blocks updated during previous iteration

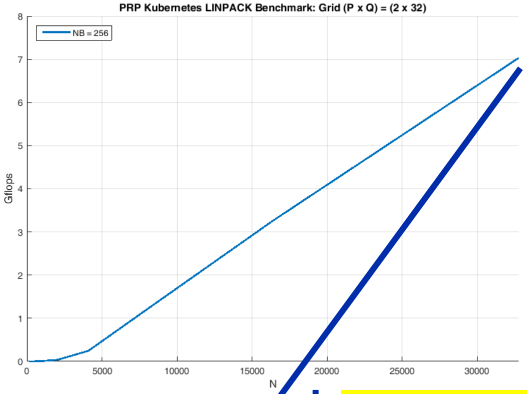
Blocks currently being factored

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

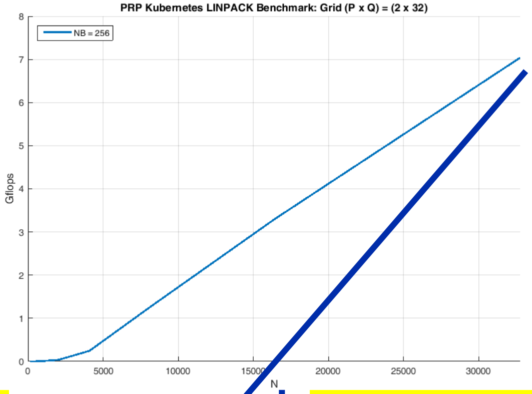
Benchmarking PRP K8s with LINPACK: *Comparison*

Left-Looking



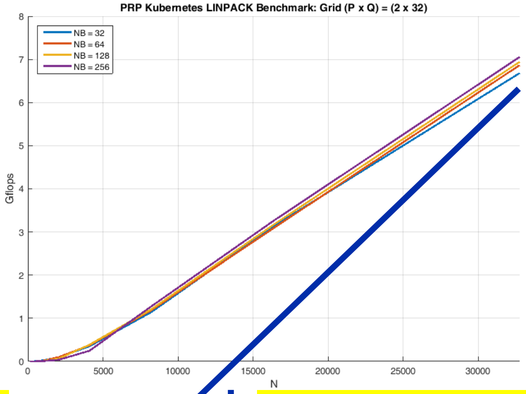
- Blocksize = 256
- (PxQ)=(2x32)
- Panel fact.: Left
- Recursive Panel fact.: Left
- 7.033 Gflops

Right-Looking



- Blocksize = 256
- (PxQ)=(2x32)
- Panel fact.: Right
- Recursive Panel fact.: Right
- 7.039 Gflops

Crout



- Blocksize = 256
- (PxQ)=(2x32)
- Panel fact.: Crout
- Recursive Panel fact.: Crout
- 7.056 Gflops

Benchmarking PRP K8s with LINPACK: *SDSC Comet*

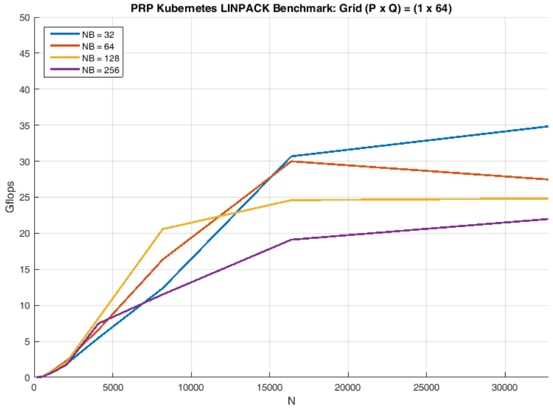
- Comet: eXtreme Science and Engineering Discovery Environment (XSEDE) cluster
- 1,944 nodes, 24 cores/node → 46,656 cores (Haswell)
- FDR InfiniBand Interconnect (MPI latency 1.03-1.97 µs)
- K8s on Comet configured by Dmitry Mishin <dmishin@ucsd.edu>

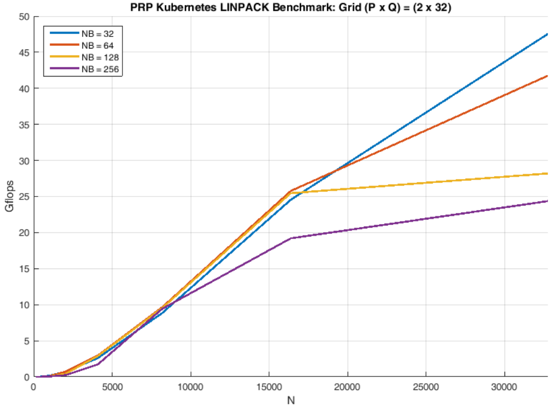
```
[paolini@fiona k8s]$ kubectl create -f comet.yaml
[paolini@fiona k8s]$ kubectl get pods -o wide -n sdsu-comet
NAME                            READY    STATUS    RESTARTS   AGE      IP            NODE
subflow-comet-7c8b94f6b6-2hpr6  1/1      Running   0          1h       10.244.66.4   comet-k8s-1.sdsc.optiputer.net
subflow-comet-7c8b94f6b6-b42tb  1/1      Running   0          1h       10.244.68.4   comet-k8s-2.sdsc.optiputer.net
subflow-comet-7c8b94f6b6-gnzgr  1/1      Running   0          1h       10.244.72.4   comet-k8s-3.sdsc.optiputer.net
subflow-comet-7c8b94f6b6-htxxd  1/1      Running   0          1h       10.244.68.3   comet-k8s-2.sdsc.optiputer.net
subflow-comet-7c8b94f6b6-rwc48  1/1      Running   0          1h       10.244.66.3   comet-k8s-1.sdsc.optiputer.net
subflow-comet-7c8b94f6b6-th6qd  1/1      Running   0          1h       10.244.67.5   comet-k8s-0.sdsc.optiputer.net
subflow-comet-7c8b94f6b6-v7cpn  1/1      Running   0          1h       10.244.67.4   comet-k8s-0.sdsc.optiputer.net
subflow-comet-7c8b94f6b6-vfwkh  1/1      Running   0          1h       10.244.72.3   comet-k8s-3.sdsc.optiputer.net
```

PACIFIC RESEARCH PLATFORM

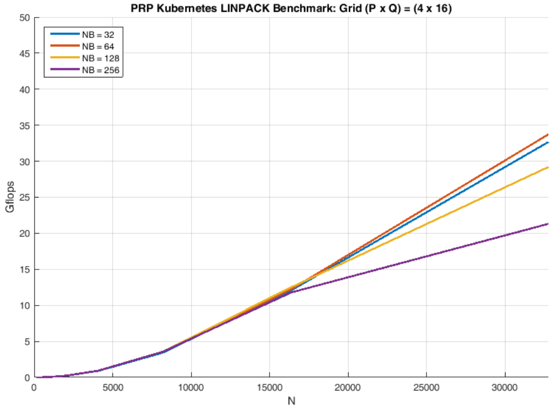
SAN DIEGO STATE UNIVERSITY

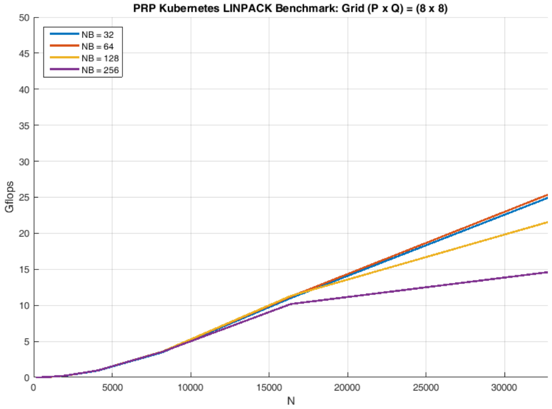
Benchmarking PRP K8s with LINPACK: *Comet Results*





- Blocksize = 32
- (PxQ)=(2x32)
- Panel fact.: Crout
- Recursive Panel fact.: Crout
- 47.54 Gflops



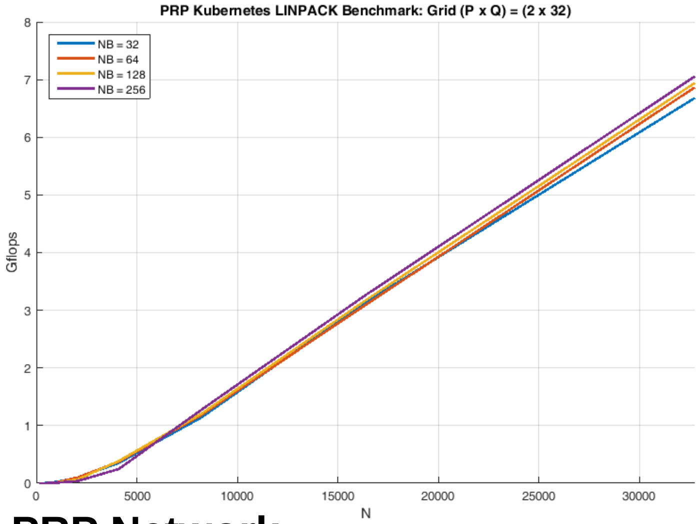


PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

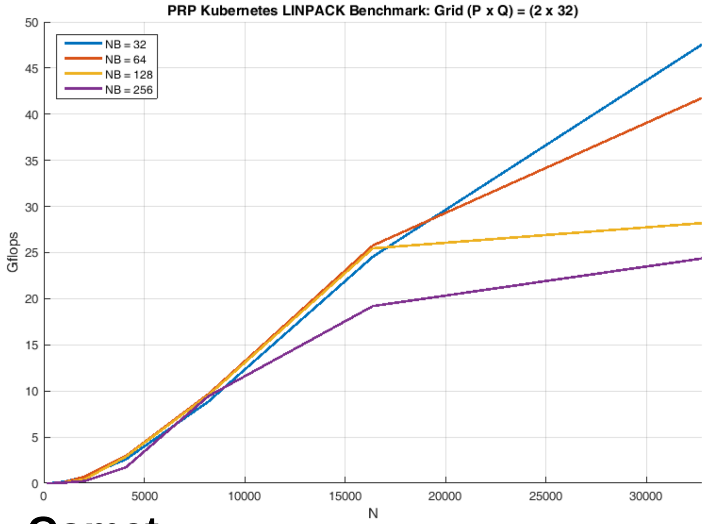
Benchmarking PRP K8s with LINPACK: *Pros and Cons*

- Slower numerical performance; however:
- Unlimited walltime (so far)
- No formal approval required (so far)
- Required limited administrator intervention



PRP Network

- Superior numerical performance; however:
- Maximum walltime for Comet jobs is 2 days
- Requires XSEDE allocation via approved proposal
- Idle VMs waste allocation SUs
- Required administrator intervention (Dmitry Mishin)

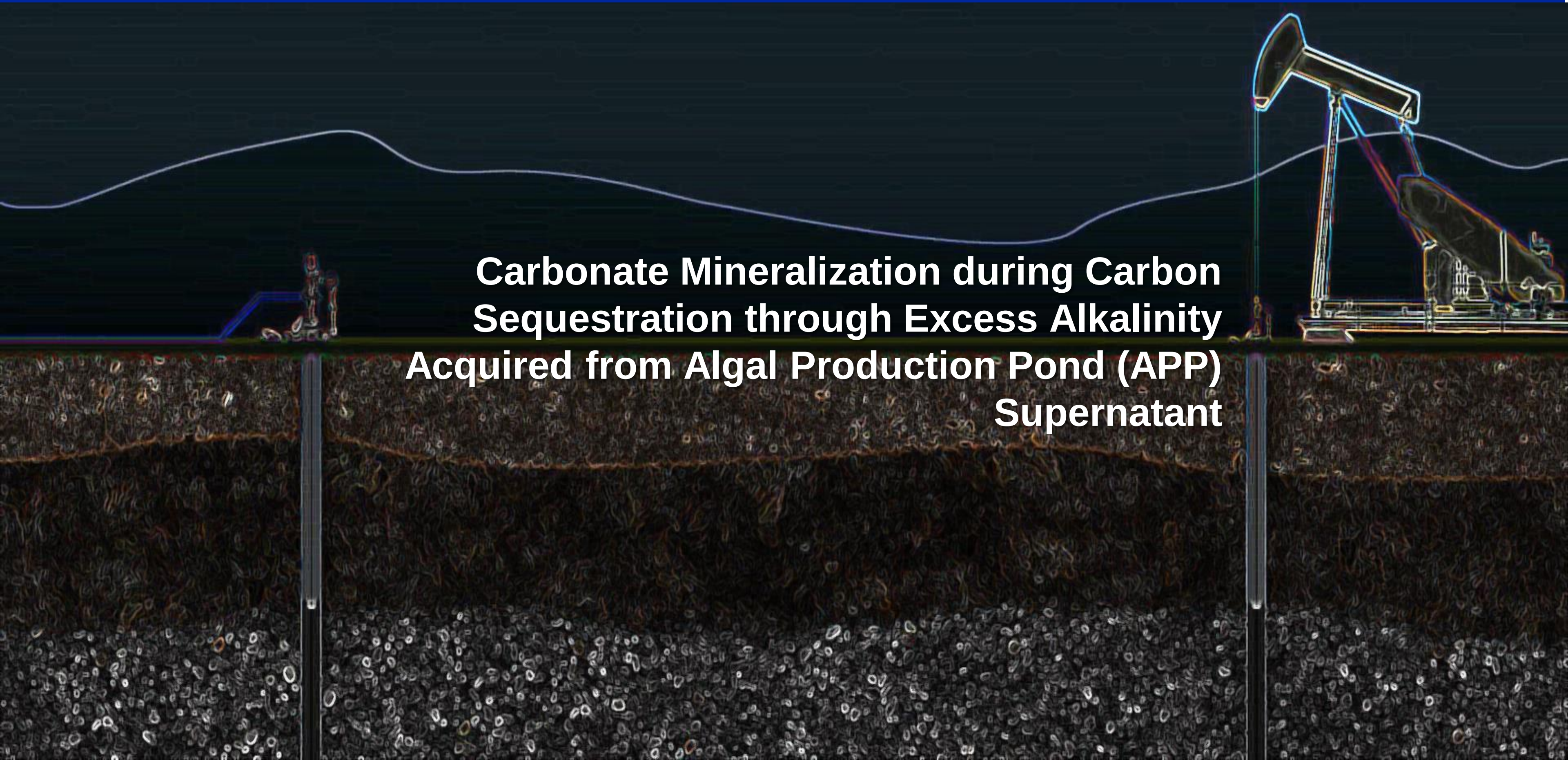


Comet

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

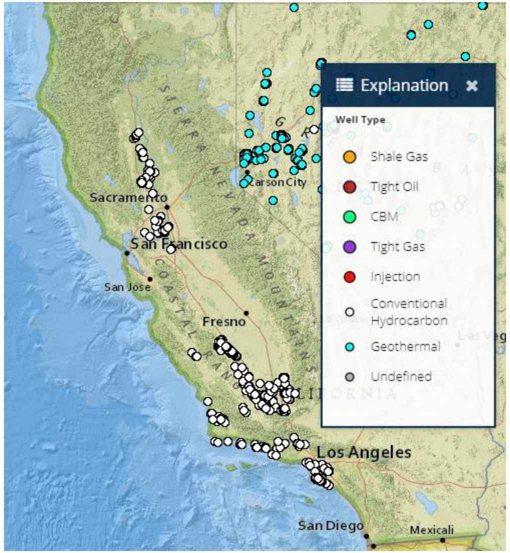
PRP Science Driver: *Numerical CO_2 Sequestration*

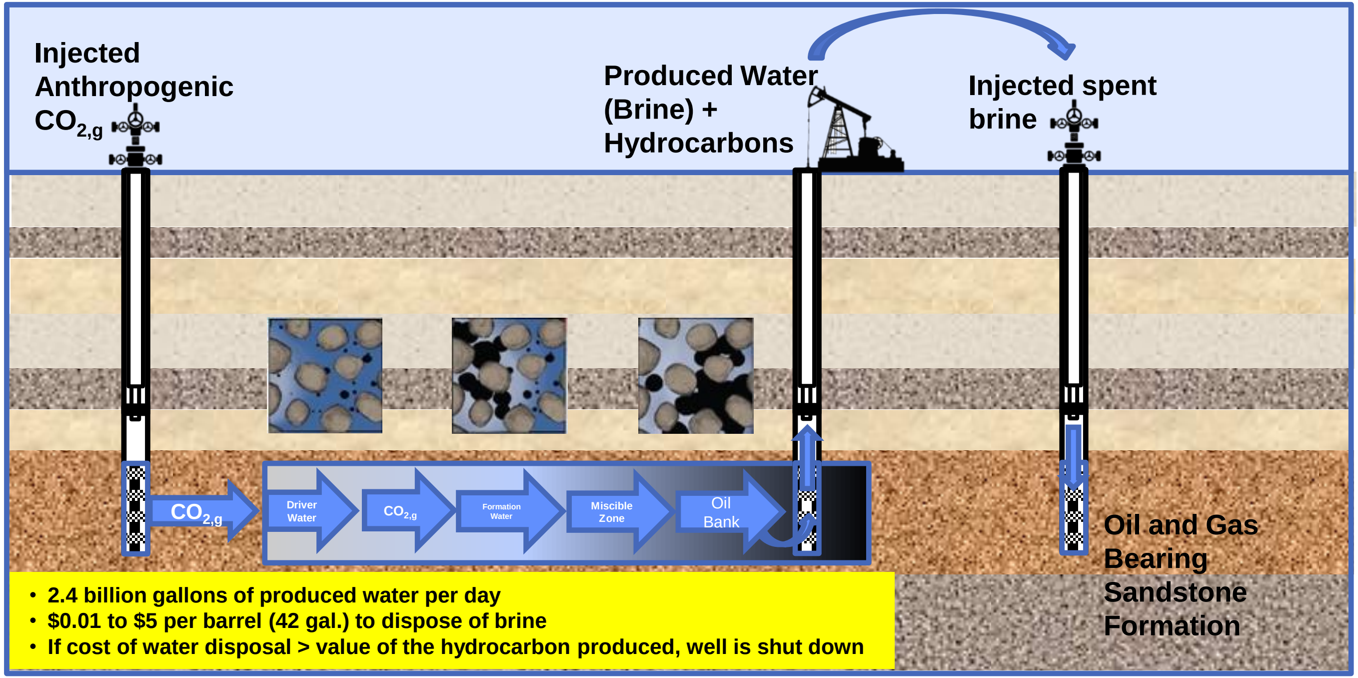


Carbonate Mineralization during Carbon Sequestration through Excess Alkalinity Acquired from Algal Production Pond (APP) Supernatant

Synergistic CCS, EOR, and Biofuel Generation

- ***Exploring Synergies in an Enhanced Waste Management System that Produces Biofuel, Recovers Oil, Sequesters Carbon, and Treats Wastewater***





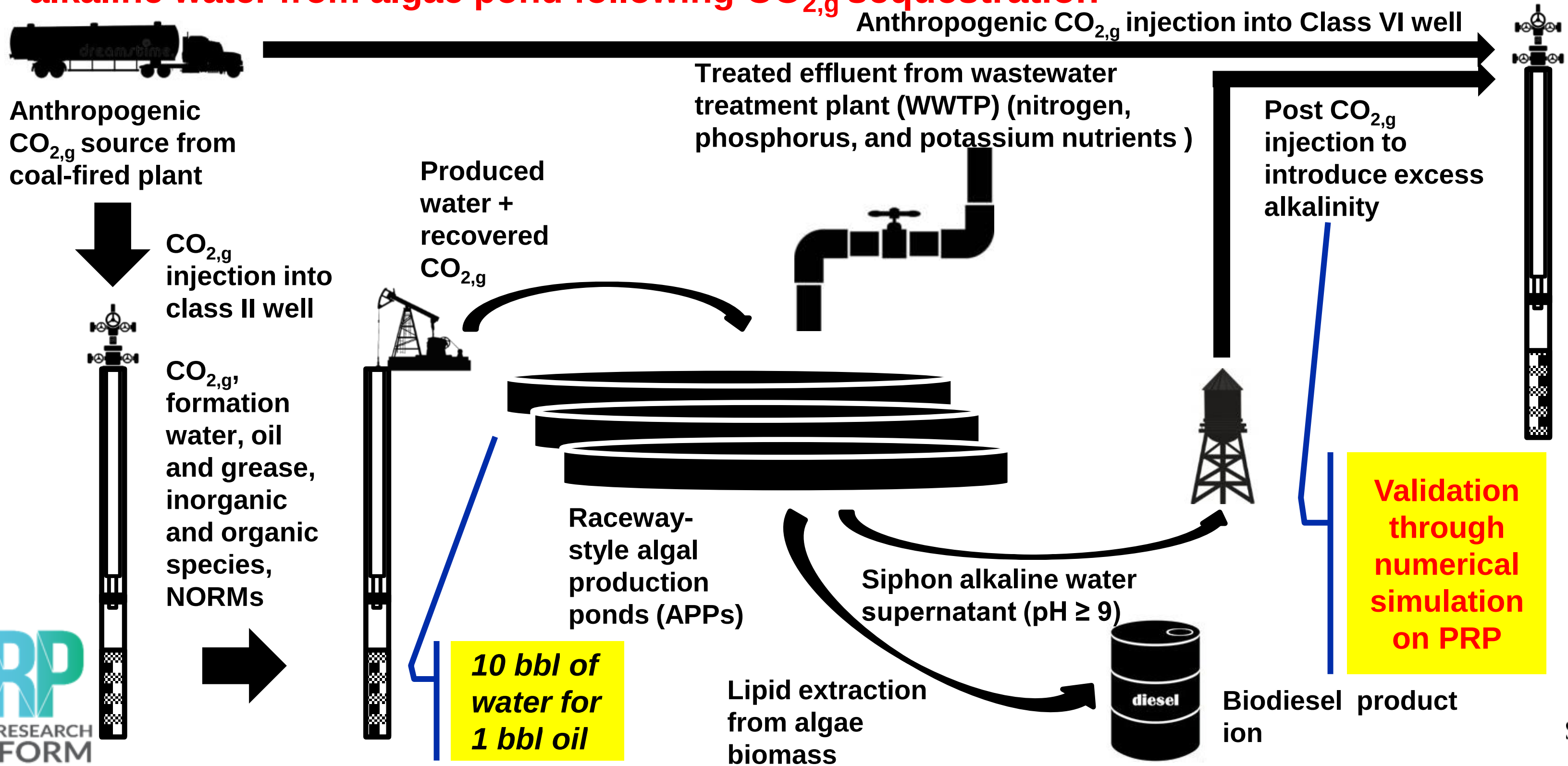
- **Primary recovery (natural pressure): 10% original oil in place recovered**
- **Secondary recovery (water injection): 20% - 40% original oil recovered**
- **Tertiary EOR (CO_2 injection): 30% to 60%**
- **US: 114 CO_2 injection projects, 2 billion cubic feet of CO_2 produce 280,000 barrels of oil per day**
- **Water management: majority (55%) of well operating expenses**

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Synergistic CCS, EOR, and Biofuel Generation

- *Novel idea:* proposed pilot project to (1) reuse produced water as medium for algal growth to eliminate cost of disposal and (2) increase rate of carbonate mineralization through injection of alkaline water from algae pond following $CO_{2,g}$ sequestration

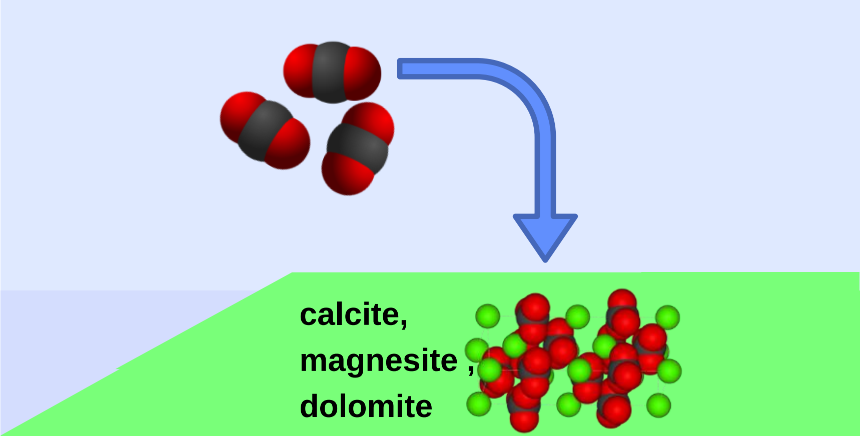


PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

“Locking” Captured $CO_{2,g}$ in Place

- Ideal storage mechanism: transform atmospheric $CO_{2,g}$ into subsurface solid phase carbonate mineral (e.g. $CaCO_3$, $MgCO_3$, $CaMg(CO_3)_2$)
- Neutralization of carbonic acid by alkaline earth metals (Ca^{2+},Mg^{2+})
- Solid phase storage: lower risk from upward gas migration and release into atmosphere



- Chemical reaction among dissolved $CO_{2,g}$ and mineral solutes in formation water
- “Mineral Trapping”: typically occurs 100 to 10,000 years after injection
- How can we achieve rapid mineralization to lower risk?

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

How CO_2 is Sequestered in Rock

- ***Structural and Stratigraphic Trapping*:**
  $CO_{2(g)}$ rises through porous rock until it reaches impermeable cap rock (e.g. shale)
- ***Residual-Phase Trapping***
  $ScCO_2$ stored as residual droplets in (sandstone) rock pore space
- ***Dissolution Trapping***
  CO_2 rich water sinks to the bottom of a formation because $\rho_{H20(l)+CO2(g)+H2CO3(aq)} > \rho_{H20(l)}$
- ***(Ideally) Mineral Trapping***
  Reaction between dissolved CO_2 in formation water with surrounding solutes to create carbonate minerals (e.g. $CaCO_3$, $MgCO_3$, $CaMg(CO_3)_2$)

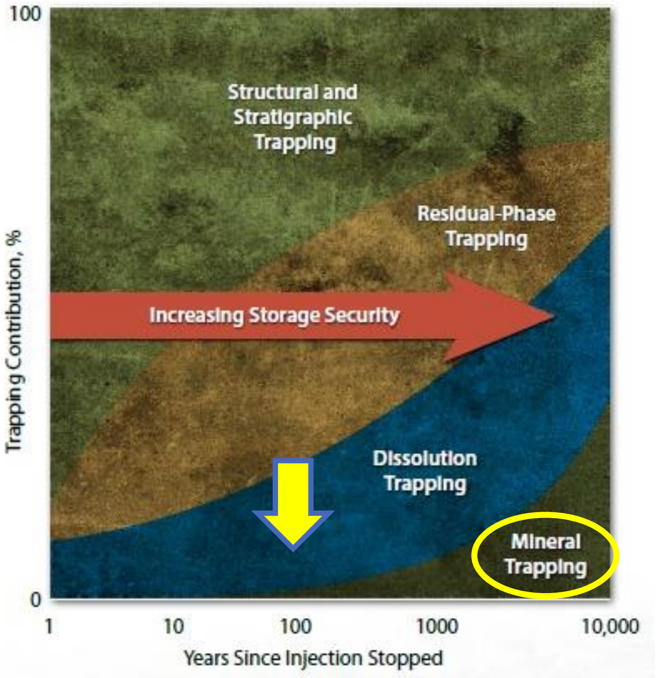


Figure: Plains CO2 Reduction (PCOR) Partnership Atlas, *4th Edition, Revised,* 2013.

Question: how might we **stimulate** a reservoir, post CO_2 injection, to induce carbonate mineral formation **sooner**?

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Problem: Calcite-CO_2 Equilibrium

- **CO_2 solvation**

$$\boxed{\begin{gathered} CO_{2,g} \rightleftharpoons CO_{2,aq} \\ CO_{2,aq} + H_2O \rightleftharpoons H_2CO_{3,aq} \end{gathered}} \quad (1)$$

$$H_2CO_{3,aq} \rightleftharpoons HCO_3^- + H^+$$

$$HCO_3^- \rightleftharpoons CO_3^{2-} + H^+$$

$$H_2O \rightleftharpoons H^+ + OH^-$$

- **Solid calcite precipitation ↔ dissolution**

$$\boxed{CaCO_{3,s} + H_2CO_{3,aq} \rightleftharpoons Ca^{+2} + 2HCO_3^-} \quad (2)$$

Le Chatelier's principle

- Increase $[CO_{2,g}]$ → (1,2) shifts to right → $CaCO_{3,s}$ dissolves
- Decease $[CO_{2,g}]$ → (1,2) shifts to left → $CaCO_{3,s}$ precipitates
- *Problem*: how to store $CO_{2,g}$ and (rapidly) precipitate calcite?

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Carbonate Alkalinity

- **Carbonate alkalinity A_C**

$$A_C = [HCO_3^-] + 2[CO_3^{2-}] + [OH^-] - [H^+]$$

- Alkalinity A_C increases through carbonate dissolution:

$$CaCO_{3,s} \rightleftharpoons Ca^{+2} + CO_3^-$$

- Carbonate precipitation thus reduces A_C and increases P_{CO2} by converting bicarbonate to CO_2:

$$CaCO_{3,s} + H_2CO_{3,aq} \rightleftharpoons Ca^{+2} + 2HCO_3^-$$

$$CO_{2,aq} + H_2O \rightleftharpoons H_2CO_{3,aq}$$

$$CO_{2,g} \rightleftharpoons CO_{2,aq}$$

- Thus, carbonate precipitation may not completely sequester carbon. What can be done?

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Excess Alkalinity and *Hardness*

$$A_C = [HCO_3^{\ -}] + 2[CO_3^{\ 2-}] + [OH^-] - [H^+]$$

- Idea 1: introduce external source of alkalinity as OH^-

$$CaCO_{3,s} + H_2O \rightleftharpoons Ca^{+2} + HCO_3^- + OH^-$$

- Idea 2: introduce external source of hardness as Ca^{2+} and Mg^{2+}

- Can we show, through modeling and numerical simulation **on a PRP cluster**, increased carbon sequestration as carbonate mineralization from underground injection of excess alkalinity and hardness?

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Excess Alkalinity Source

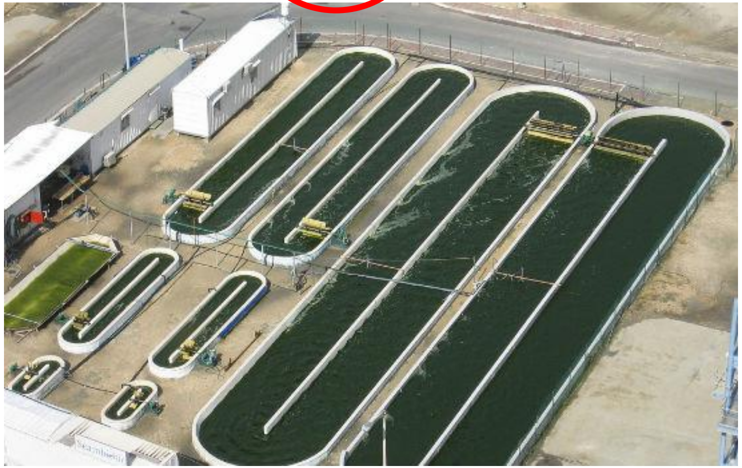
- Reduction of carbonic acid to formaldehyde and hydroxide production from photosynthesis

$$H_2CO_3 \rightleftharpoons CH_2O + O_2$$

$$HCO_3^- + H_2O \rightleftharpoons CH_2O + O_2 + OH^-$$

$$CO_3^{2-} + 2H_2O \rightleftharpoons CH_2O + O_2 + 2OH^-$$

High rate algal pond (HRAP) CO_2 addition sump (Craggs et al., 2014)



Seambiotic's commercial algae farm in China uses CO_2 captured from power plant flue gas for photosynthesis

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Produced Water Chemistry

- Produced waters of different geologic ages compared with average concentrations in 35‰ seawater (Collins, A.G. 1975, *Geochemistry of Oilfield Waters*, Elsevier Scientific Publishers, New York. 496 pp)

Source of hardness

Element/Ion	Seawater	Produced Water	
		Highest Concentration (Age[1])	Range of Mean Concentrations
Salinity	35,000	---	<5000 - >300,000,000
Sodium	10,760	120,000 (J)	23,000 – 57,300
Chloride	19,353	270,000 (P)	46,100 – 141,000
Calcium	416	205,000 (P)	2530 – 25,800
Magnesium	1294	26,000 (D)	530 – 4300
Potassium	387	11,600 (D)	130 – 3100
Sulfate	2712	8400 (T)	210 – 1170
Bromide	87	6000 (J)	46 – 1200
Strontium	0.008	4500 (P)	7 – 1000
Ammonium	---	3300 (P)	23 – 300
Bicarbonate	142	3600 (T)	77 – 560
Iodide	167	1410 (P)	3 – 210
Boron	4.45	450 (T)	8 – 40
Carbonate	---	450 (M)	30 – 450
Lithium	0.17	400 (J)	3 – 50

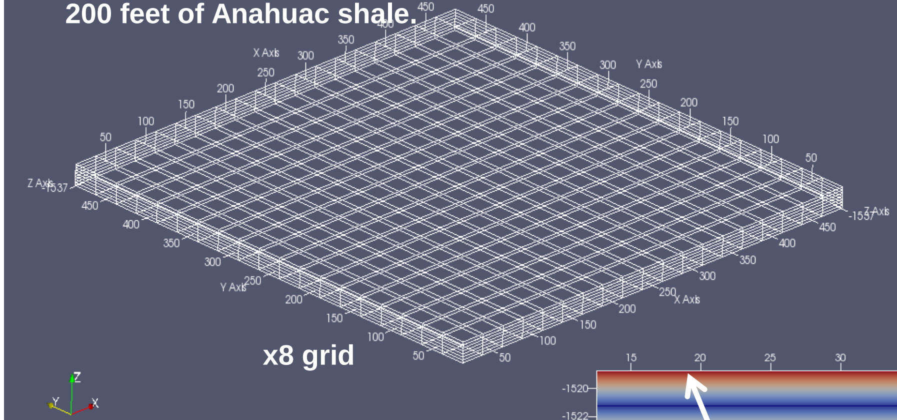
[1] D, Devonian, J, Jurassic, M, Mississippian, P, Pennsylvanian, T, Tertiary.

(Neff, J. M., et al., Produced Water: Overview of Composition, Fates and Effects, 2014)

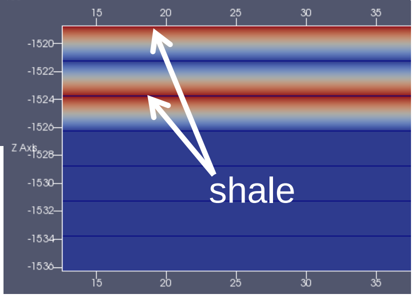
Modeled Reservoir based on Frio Formation

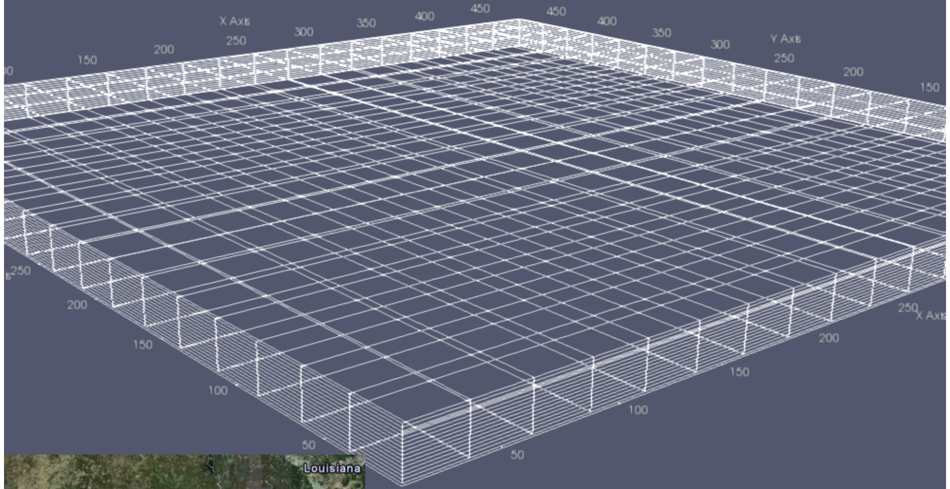
- **0.5 km x 0.5 km x 17.5 m volume**
- **Three sandstone layers separated by two shale layers**
- **4 grid configurations: 20x20x{8, 16, 32, 64}**

- **Frio location is 30 miles northeast of Houston, in the South Liberty oilfield.**
- **Injection Zone is a brine-sandstone system with a top seal of 200 feet of Anahuac shale.**

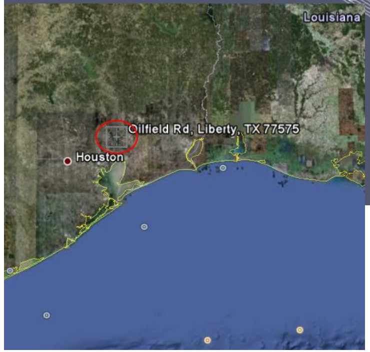


x8 grid





x16 grid



PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Molar Concentration of $CO_{2,aq}$

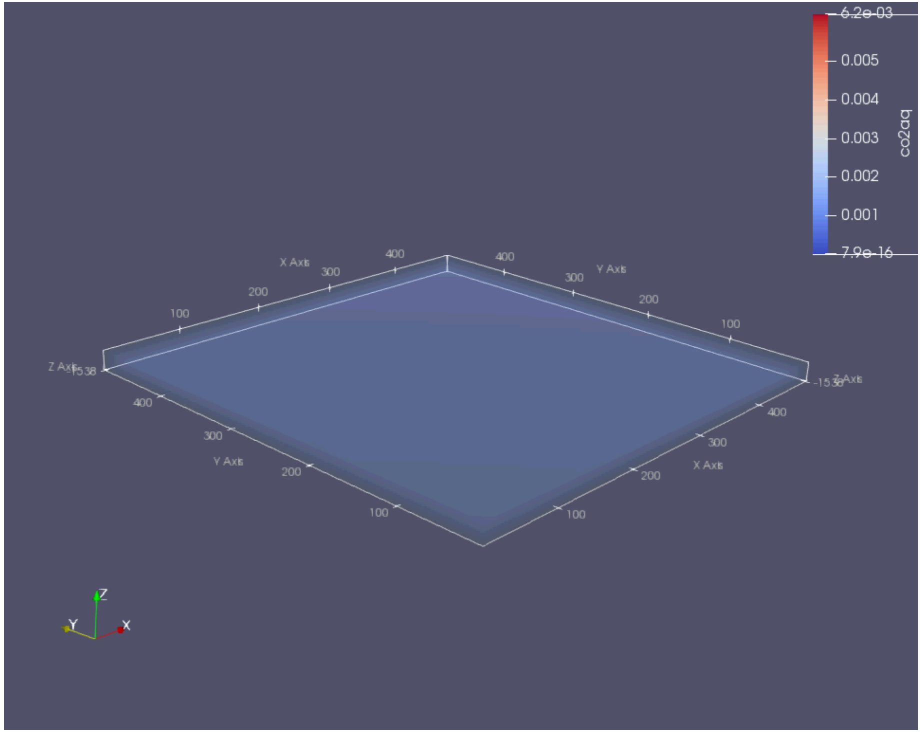
Mineralogical Composition

Shale (impervious caprock, κ=7x10^{-3} mD)

Mineral	Volume Fraction
Calcite ($CaCO_3$)	0.20
Quartz (SiO_2)	0.28
k-feldspar ($KAlSi_3O_8$)	0.01
Illite ($(K,H_3O)(Al,Mg,Fe)_2(Si,Al)_4O_{10}[(OH)_2)$)	0.41

Sandstone (Semi-pervious, κ= 17.2 mD)

Mineral	Volume Fraction
Quartz (SiO_2)	0.53
k-feldspar ($KAlSi_3O_8$)	0.11 (subarkose)



10 mM CO_2-rich Injectant

Species	Molarity
H^+	2.6e-06
H2O	1
$CO_{2aq}/HCO_3^- /CO_3^{--}$ complex	0.01

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Modeling Porous Media Flow

- **Darcy's Law (fluid flux [g/(cm² s)])**

$$v_{\text{res}} = -\frac{K}{\mu}\rho_0\left(\nabla p_{\text{res}} - \rho g \nabla z\right)$$

- **Fluid Conservation (continuity equation)**

$$\partial_t(\rho_0\phi) + \nabla \cdot v_{\text{res}} = q_{\text{res}}$$

v	Fluid mass flux [g/(cm² s)]	g	Acceleration due to gravity [cm/s²]
K	Permeability tensor of porous medium [1/cm]	∇z	Gravitational direction unit vector [cm]
		ϕ	Porosity of medium (dimensionless)
μ	Dynamic viscosity [g/(cm s)]	q	Fluid source density [g/(cm³ s)]
ρ	Fluid density [g/cm³]		
p	Fluid pressure [gfsc]		

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Poroelastic Models

- **Strain: the symmetric gradient of displacement**

$$\epsilon(u) = \frac{1}{2}\left(\nabla u + \nabla u^{T}\right)$$

- **Rock Stress**

$$\sigma(u) = \lambda\left(\nabla \cdot u\right) I + 2G\epsilon(u)$$

- **Poroelastic Stress**

$$\sigma_{por}(u, p) = \sigma(u) - \alpha p I$$

λ	First Lame parameter [gfsc]
G	Second Lame parameter (Shear modulus) [gfsc]
α	Biot-Willis coefficient (dimensionless)

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Solute Mass Transport Model

- **Elemental Conservation of Mass per Unit Volume**

$$\frac{\partial e_\beta}{\partial t} = \sum_{\alpha=1}^{N_\alpha} \left[\phi D_\alpha \nabla^2 c_\alpha - \phi \nabla \cdot (c_\alpha u)\right] - \sum_{\gamma=1}^{M} \nu_{\beta\gamma} A_\gamma G_\gamma$$

(Park, 2014)

Elemental mass rate of change term: rate of increase of concentration of a solute atom β in a fluid element

Diffusive term: net rate of increase of solute activity in a fluid element due to diffusive forces

Advective term: net rate of flow of solute activity out of a fluid element due to advective forces

Source term: net rate of the increase or decrease of a mineral in a fluid element due to chemical kinetics

- Evolution of chemical elemental mass depends on mass-transfer from diffusive and advective forces as well as the precipitation and dissolution of minerals governed by kinetic reaction rates

e	elemental mass in fluid [mol/cm^3]	ϕ	porosity [unitless]
D	diffusion coefficient [cm^2/s]	β	solute atom index
c	solute concentration [mol/cm^3]	γ	mineral index
α	aqueous solute species index	G	mineral reaction rate [mol/(cm^2 s)]
u	fluid velocity [cm/s]	A	mineral spec. surface area [cm^2/cm^3]

PACIFIC RESEARCH PLATFORM

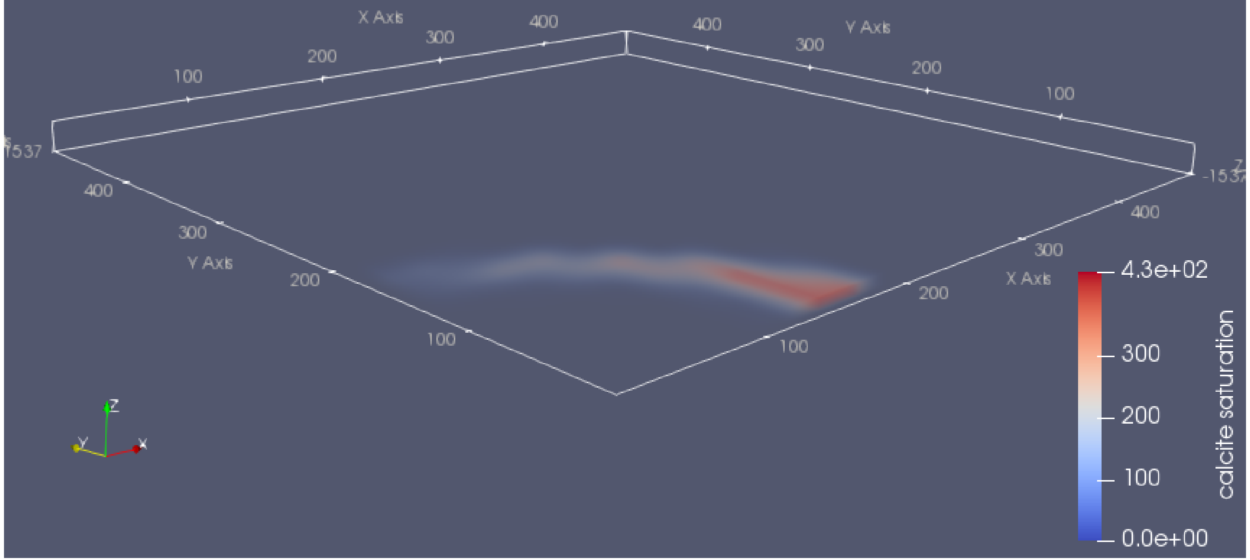
SAN DIEGO STATE UNIVERSITY

Calcite Saturation

Mineral saturation Ω is the ratio of the Ion Activity Product (*IAP*) to the (equilibrium) solubility product (K_{sp}). For calcite,

Precipitation requires
$\Omega > 1$ or $IAP > K_{sp}$

$$\Omega = \frac{IAP}{K_{sp}} = \frac{a_{Ca^{2+},actual}\, a_{CO_3^{2-},actual}}{a_{Ca^{2+},equilibrium}\, a_{CO_3^{2-},equilibrium}}$$



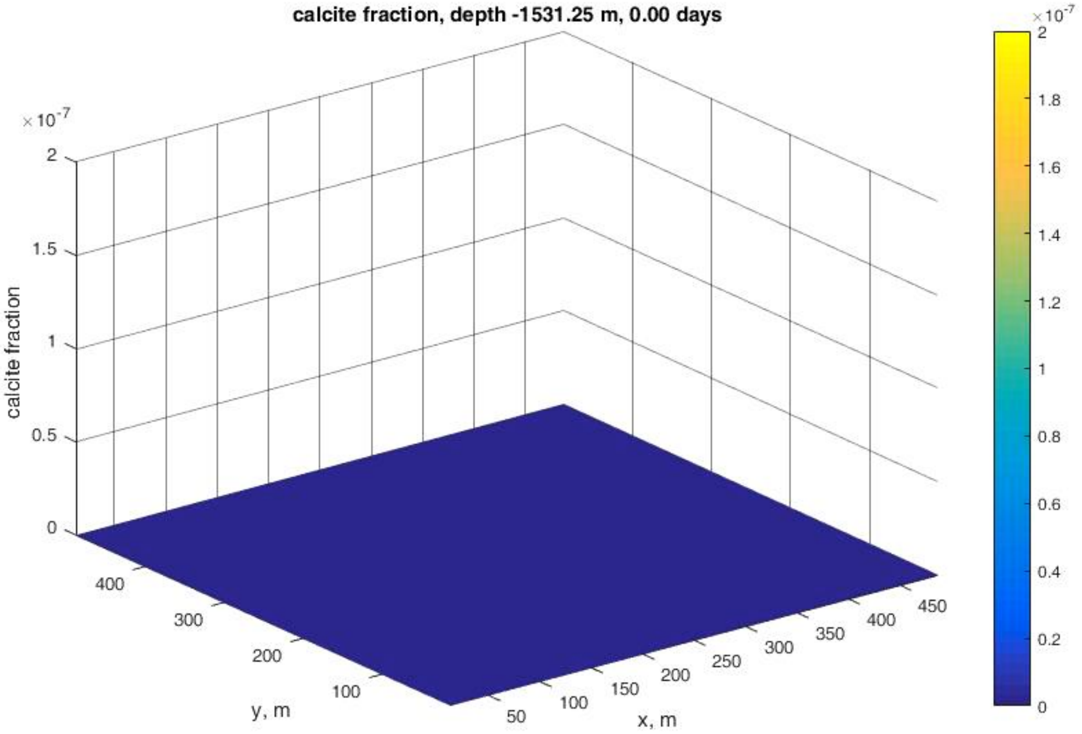
Calcite saturation after 1 day of produced water solution #1

PRP Results: 1 kg/s, 5 years Equilibrium

Calcite fraction after 1 day of produced water solution #1 and 5 years of equilibrium (2e-7)

Solute	Concentration
OH^-	1e-2 M, pH = 12
Ca^{++}	2530 ppm
Na^+	23000 ppm
Mg^{++}	530 ppm
Cl^-	46100 ppm

High alkalinity, low salinity, low hardness test water solution #1



PACIFIC RESEARCH PLATFORM

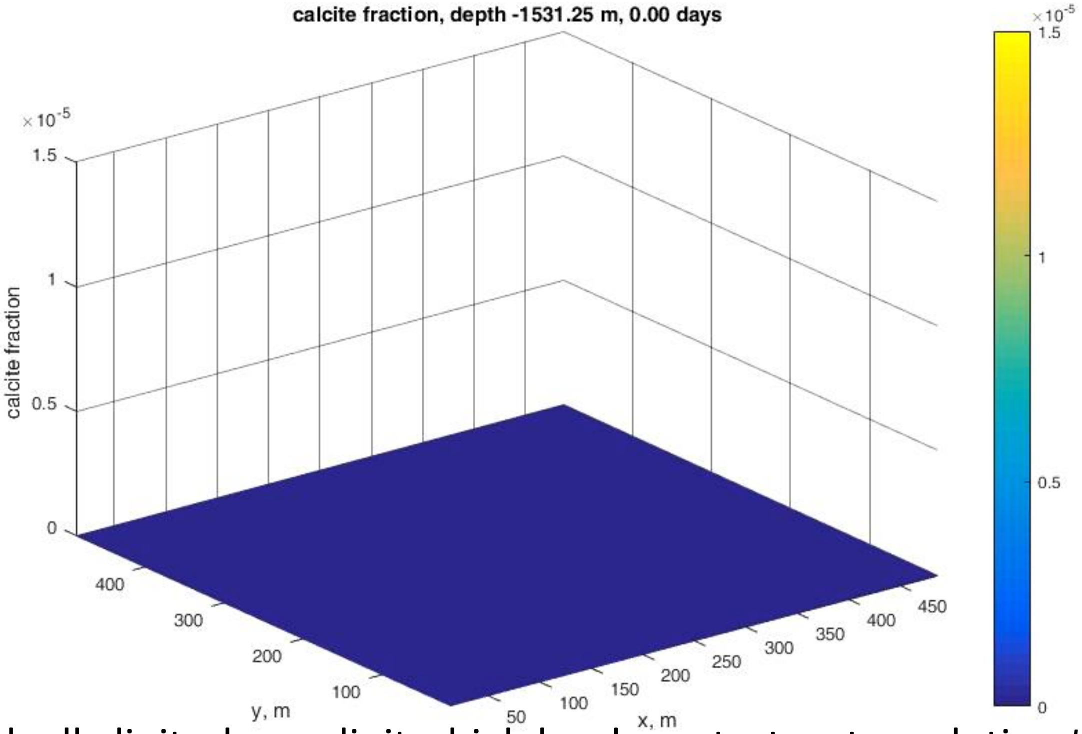
SAN DIEGO STATE UNIVERSITY

Comet Results: High Hardness, 1 kg/s, 1 year Equilibrium

Calcite fraction after 1 day of produced water solution #2 and 1 year of equilibrium (1.5e-5)

Solute	Concentration
OH^-	1e-2 M, pH = 12
Ca^{++}	25800 ppm
Na^+	23000 ppm
Mg^{++}	4300 ppm
Cl^-	46100 ppm

High alkalinity, low salinity, high hardness test water solution #2



High alkalinity, low salinity, high hardness test water solution #2

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

PRP Science Driver: *Running Subflow on Comet*

```
 demo-comet)
        KUBE_NAMESPACE=sdsu-comet
        export KUBE_NAMESPACE
        HEADNODE=$MPI_CLUSTER_NAME-comet-6788dc75d5-pgw78
        export HEADNODE
        kubectl -n $KUBE_NAMESPACE exec -it $HEADNODE -- mpirun --allow-run-as-root \
            --hostfile /etc/ssh/shosts.equiv \
            --display-map -n 4 -npernode 1 \
              --mca btl tcp,self \
            sh -c 'cd /nfs/subflow/exe; GRID=FrioTHMC20x20x16; export GRID; ./subflow -omp 4 -i demo.sdb -g $GRID.grid -k $GR
ID > OUTPUT/$GRID-demo.out 2>&1'
        ;;

[paolini@ps-40g k8s]$ ./RUN.sh demo-comet
 Data for JOB [57040,1] offset 0

 ========================   JOB MAP   ========================

 Data for node: subflow-comet-6788dc75d5-778bz  Num slots: 24   Max slots: 0    Num procs: 1
        Process OMPI jobid: [57040,1] App: 0 Process rank: 0

 Data for node: subflow-comet-6788dc75d5-p8z2g  Num slots: 24   Max slots: 0    Num procs: 1
        Process OMPI jobid: [57040,1] App: 0 Process rank: 1

 Data for node: subflow-comet-6788dc75d5-pgw78  Num slots: 24   Max slots: 0    Num procs: 1
        Process OMPI jobid: [57040,1] App: 0 Process rank: 2

 Data for node: subflow-comet-6788dc75d5-q6wvc  Num slots: 24   Max slots: 0    Num procs: 1
        Process OMPI jobid: [57040,1] App: 0 Process rank: 3

 =============================================================
```

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Concluding Remarks

Simulations show it may be feasible to induce early carbonate mineral formation through alkaline produced water injection following CO_2 injection

Further Work

Intellectual merit

- What species of algae (e.g. *Dunaliella salina*, *Botryococcus braunii, Chlorella, Dunaliella tertiolecta, Gracilaria, Pleurochrysis carterae, Sargassum*) thrive best, if at all, in produced waters from different oil and gas sites?
- How does produced water salinity affect supernatant alkalinity?
- How do trace amounts of naturally occurring radioactive materials (NORMs) such as Radium affect algae growth? Radium isotopes present in produced water and barite (barium sulfate) scale are ^{226}Ra and ^{228}Ra.
- How does alkaline supernatant injection increase the mineralization rates of calcite, magnesite, and dolomite following $CO_{2,g}$ injection?
- What is the optimal ratio of produced water to waste water to promote algae growth?

Broader Impacts

- Cost of produced water disposal is a significant factor in determining the profit of oil and gas production. Reducing or eliminating cost of disposal by diverting produced water to algal production ponds (APPs) for biofuel generation reduces per barrel cost, which lowers fuel cost borne by consumer.
- Transforming atmospheric $CO_{2,g}$ to subsurface solid phase carbonate mineral (e.g. $CaCO_3$, $MgCO_3$, $CaMg(CO_3)_2$) through mineral trapping is a superior technology to mitigate harmful climate-change from fossil fuel combustion. Geologic mineral trapping processes typically occur at the 100yr time scale. Reducing this time reduces risk of unwanted $CO_{2,g}$ subsurface transport and leakage back to the atmosphere.

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY

Questions?

PACIFIC RESEARCH PLATFORM

SAN DIEGO STATE UNIVERSITY