unifyFS Tutorial

PRESENTER: **CHEN WANG**

CONTRIBUTORS: MICHAEL BRIM, ADAM MOODY, SEUNG-HWAN LIM, ROSS MILLER, SWEN BOEHM, CAMERON STANAVIGE, KATHRYN MOHROR(PI), SARP ORAL

Tutorial: How do I modify my application for UnifyFS?

- Example MPI application
  - To use UnifyFS, change the file path(s) to point to the UnifyFS mount point at `/unifyfs`

```
int main(int argc, char * argv[]) {
    FILE *fp;
     // program initialization
     // MPI setup

     // perform I/O
    fp = fopen("/unifyfs/dset.txt", "w");
    fprintf(fp, "Hello World! I'm rank %d", rank);
    fclose(fp);

     // clean up
    return 0;
}
```

Tutorial: How does UnifyFS intercept I/O calls?

- Static Linking
  - To intercept I/O calls using a static link you'll need to add flags to your link line.
  - UnifyFS installs a `unifyfs-config` script that returns those flags:

```
$ mpicc -o hello hello.c `unifyfs-config --pre-ld-flags` \
                         `unifyfs-config --post-ld-flags`
```

- Dynamic Linking (Recommended method)

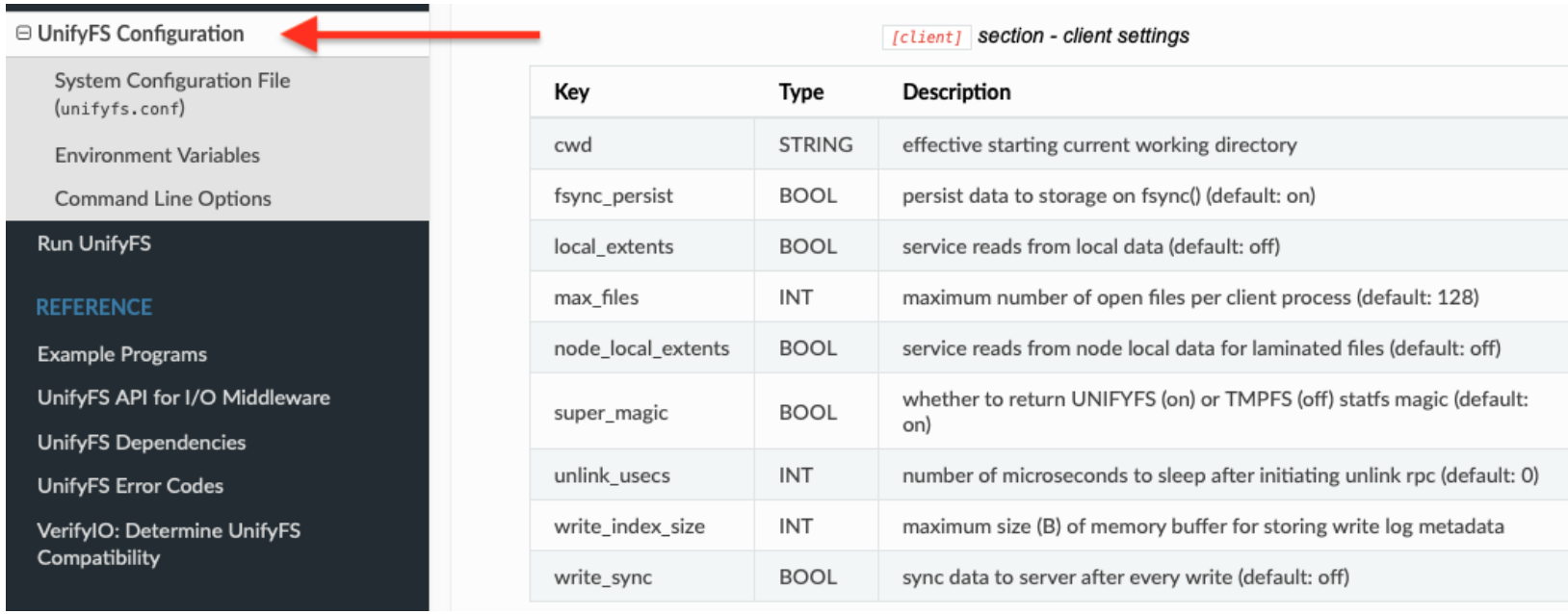
```
$ mpicc -o hello hello.c \
    -L<unifyfs_dir>/lib -lunifyfs_mpi_gotcha
```

Tutorial: How do I set up my code to run with UnifyFS?

- UnifyFS provides the following ways to set configuration settings:
  - Configuration file: `$INSTALL_PREFIX/etc/unifyfs/unifyfs.conf`
  - **Environment variables**
  - Command line options to `unifyfs start`
    - Available for a subset of config options (`unifyfs start -h`)

Tutorial: How do I set up my code to run with UnifyFS?

- UnifyFS is a user-level, customizable file system



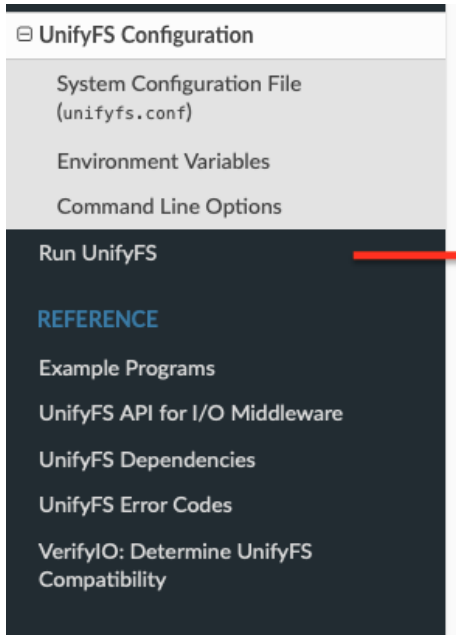
`[client]` *section - client settings*

Key	Type	Description
cwd	STRING	effective starting current working directory
fsync_persist	BOOL	persist data to storage on fsync() (default: on)
local_extents	BOOL	service reads from local data (default: off)
max_files	INT	maximum number of open files per client process (default: 128)
node_local_extents	BOOL	service reads from node local data for laminated files (default: off)
super_magic	BOOL	whether to return UNIFYFS (on) or TMPFS (off) statfs magic (default: on)
unlink_usecs	INT	number of microseconds to sleep after initiating unlink rpc (default: 0)
write_index_size	INT	maximum size (B) of memory buffer for storing write log metadata
write_sync	BOOL	sync data to server after every write (default: off)

- **Link to detailed breakdown of all UnifyFS configuration options: https://unifyfs.readthedocs.io/**

Tutorial: How do I set up my code to run with UnifyFS?

- For example, enabling `client.local_extents` may significantly improve read performance for extents written by the same process.



[client] section - client settings

Key	Type	Description
cwd	STRING	effective starting current working directory
fsync_persist	BOOL	persist data to storage on fsync() (default: on)
local_extents	BOOL	service reads from local data (default: off)
max_files	INT	maximum number of open files per client process (default: 128)
node_local_extents	BOOL	service reads from node local data for laminated files (default: off)
super_magic	BOOL	whether to return UNIFYFS (on) or TMPFS (off) statfs magic (default: on)
unlink_usecs	INT	number of microseconds to sleep after initiating unlink rpc (default: 0)
write_index_size	INT	maximum size (B) of memory buffer for storing write log metadata
write_sync	BOOL	sync data to server after every write (default: off)

- **Link to detailed breakdown of all UnifyFS configuration options: https://unifyfs.readthedocs.io/**

Tutorial: How do I set up my code to run with UnifyFS?

- A minimum setup

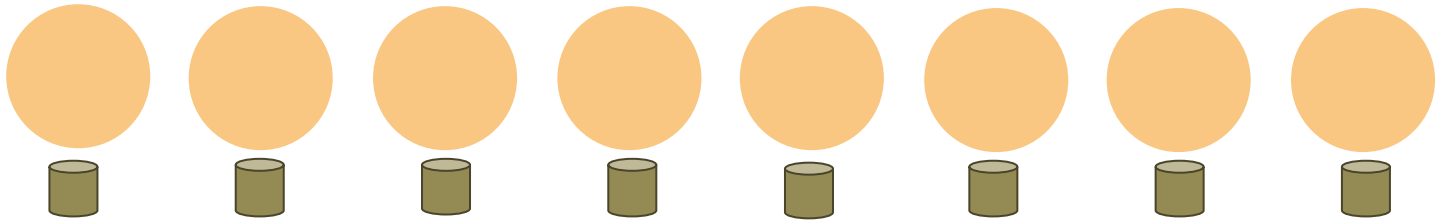
```
### UnifyFS Log file locations
$ export UNIFYFS_LOG_DIR=/xxx/unifyfs-workdir/

### Burst Buffer devices
$ export UNIFYFS_LOGIO_SPILL_DIR=/mnt/ssd/$USER
```

Tutorial: How do I run my code with UnifyFS?

- Easiest method: Start & stop UnifyFS in your batch script

```
### allocate nodes and options for resource manager
$ salloc -N 2 --tasks-per-node=4
```

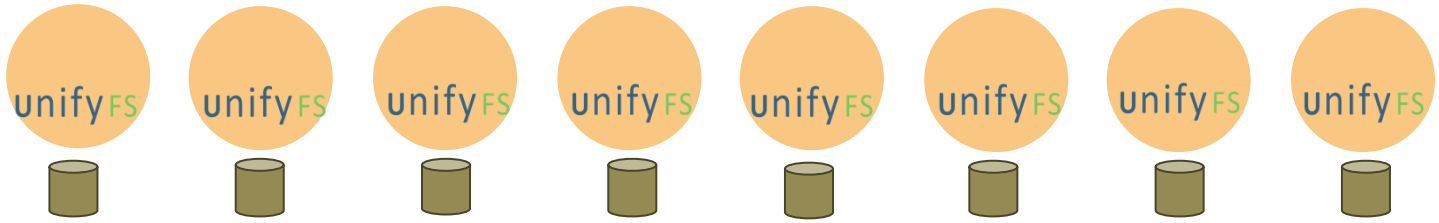


Tutorial: How do I run my code with UnifyFS?

- Easiest method: Start & stop UnifyFS in your batch script
  - Command ‘unifyfs start’ launches UnifyFS for your job and sets up the file system

```
### allocate nodes and options for resource manager
$ salloc -N 2 --tasks-per-node=4

### shell command portion of batch script
unifyfs start --share-dir=/xxx/unifyfs-workdir/
```

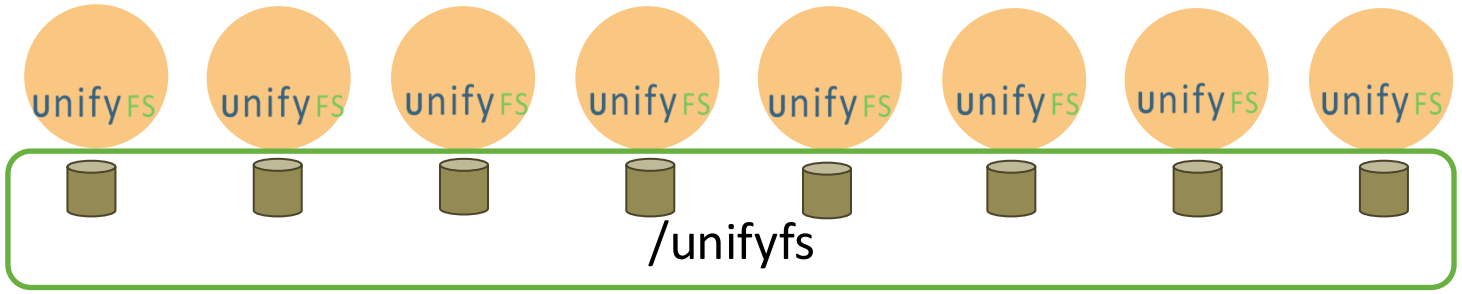


Tutorial: How do I run my code with UnifyFS?

- Easiest method: Start & stop UnifyFS in your batch script
  - Command ‘unifyfs start’ launches UnifyFS for your job and sets up the file system

```
### allocate nodes and options for resource manager
$ salloc -N 2 --tasks-per-node=4

### shell command portion of batch script
unifyfs start --share-dir=/xxx/unifyfs-workdir/
```

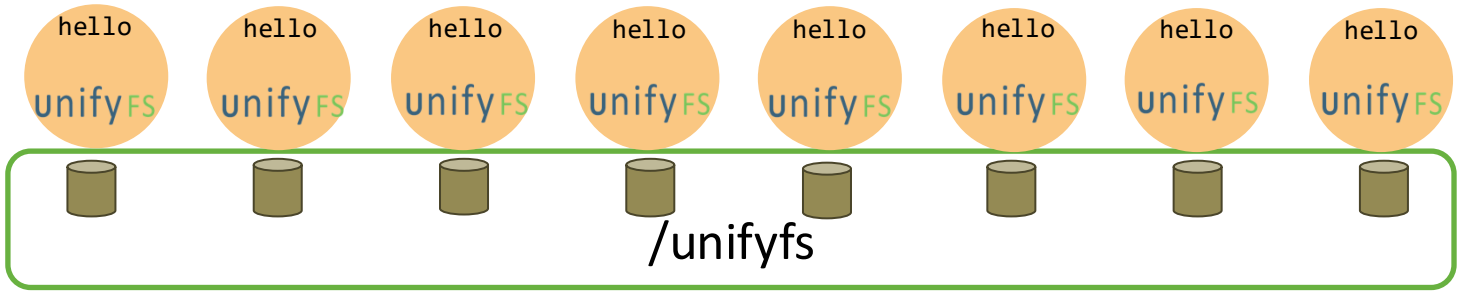


Tutorial: How do I run my code with UnifyFS?

- Easiest method: Start & stop UnifyFS in your batch script
  - Command ‘unifyfs start’ launches UnifyFS for your job and sets up the file system
  - Run your command as usual and use the path /unifyfs to direct data to UnifyFS

```
### allocate nodes and options for resource manager
$ salloc -N 2 --tasks-per-node=4

### shell command portion of batch script
unifyfs start --share-dir=/xxx/unifyfs-workdir/
srun ./hello
```

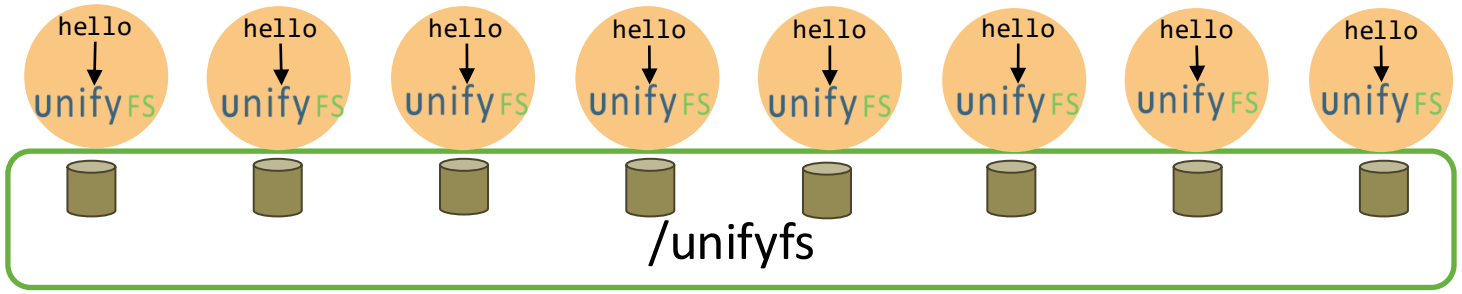


Tutorial: How do I run my code with UnifyFS?

- Easiest method: Start & stop UnifyFS in your batch script
  - Command ‘unifyfs start’ launches UnifyFS for your job and sets up the file system
  - Run your command as usual and use the path /unifyfs to direct data to UnifyFS

```
### allocate nodes and options for resource manager
$ salloc -N 2 --tasks-per-node=4

### shell command portion of batch script
unifyfs start --share-dir=/xxx/unifyfs-workdir/
srun ./hello
```

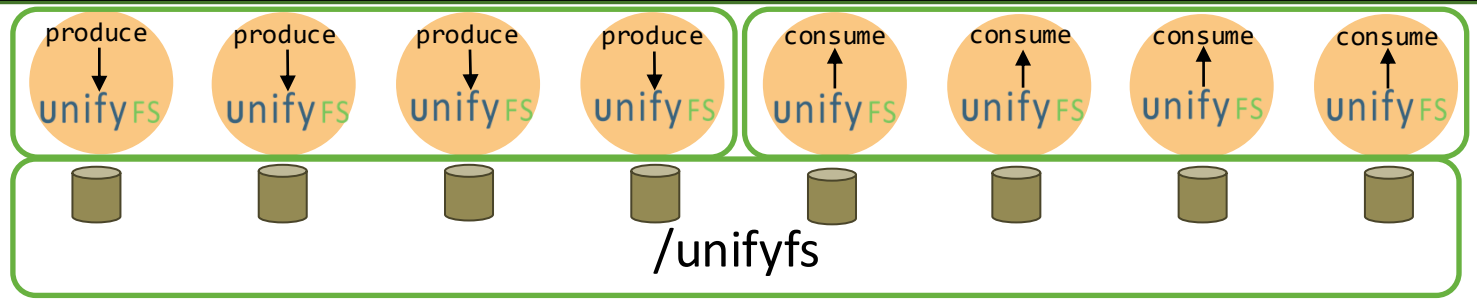


Tutorial: How do I run my code with UnifyFS?

- Easiest method: Start & stop UnifyFS in your batch script
  - Command ‘unifyfs start’ launches UnifyFS for your job and sets up the file system
  - Run your command as usual and use the path /unifyfs to direct data to UnifyFS

```
### allocate nodes and options for resource manager
$ salloc -N 2 --tasks-per-node=4

### shell command portion of batch script
unifyfs start --share-dir=/xxx/unifyfs-workdir/
srun ./hello
```

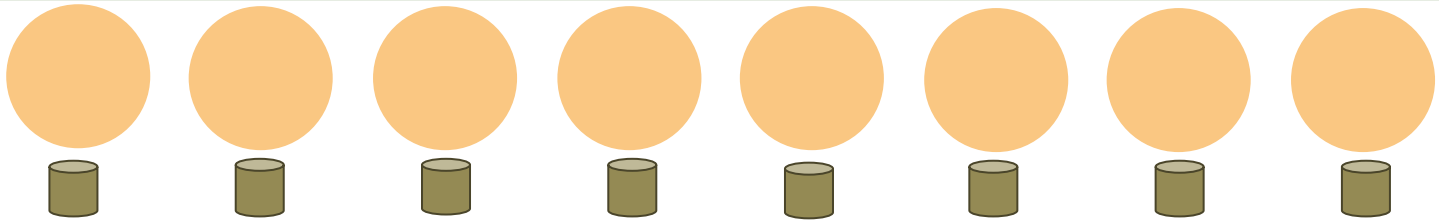


Tutorial: How do I run my code with UnifyFS?

- Easiest method: Start & stop UnifyFS in your batch script
  - Command ‘unifyfs start’ launches UnifyFS for your job and sets up the file system
  - Run your command as usual and use the path /unifyfs to direct data to UnifyFS
  - Command ‘unifyfs terminate’ cleans up the UnifyFS file system and tears it down

```
### allocate nodes and options for resource manager
$ salloc -N 2 --tasks-per-node=8

### shell command portion of batch script
unifyfs start --share-dir=/xxx/unifyfs-workdir/
srun ./hello
unifyfs terminate
```



Tutorial: How do I move my data into and out of UnifyFS?

- Three ways to move data between UnifyFS and the parallel file system
- UnifyFS transfer API
  - `unifyfs_transfer_file_parallel("/unifyfs/out.txt", "/scratch/out.txt");`
- UnifyFS transfer program
  - `unifyfs-stage $MY_MANIFEST_FILE`
- Stage in and out options with UnifyFS commands "unifyfs start" & "unifyfs terminate"
  - `unifyfs start --stage-in=$MY_INPUTS_MANIFEST_FILE`
  - `unifyfs terminate --stage-out=$MY_OUTPUTS_MANIFEST_FILE`

Demo & Hands-on