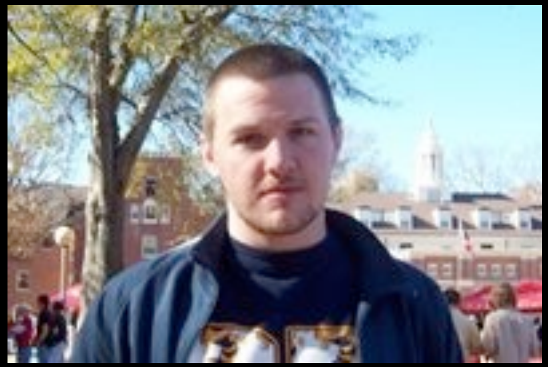
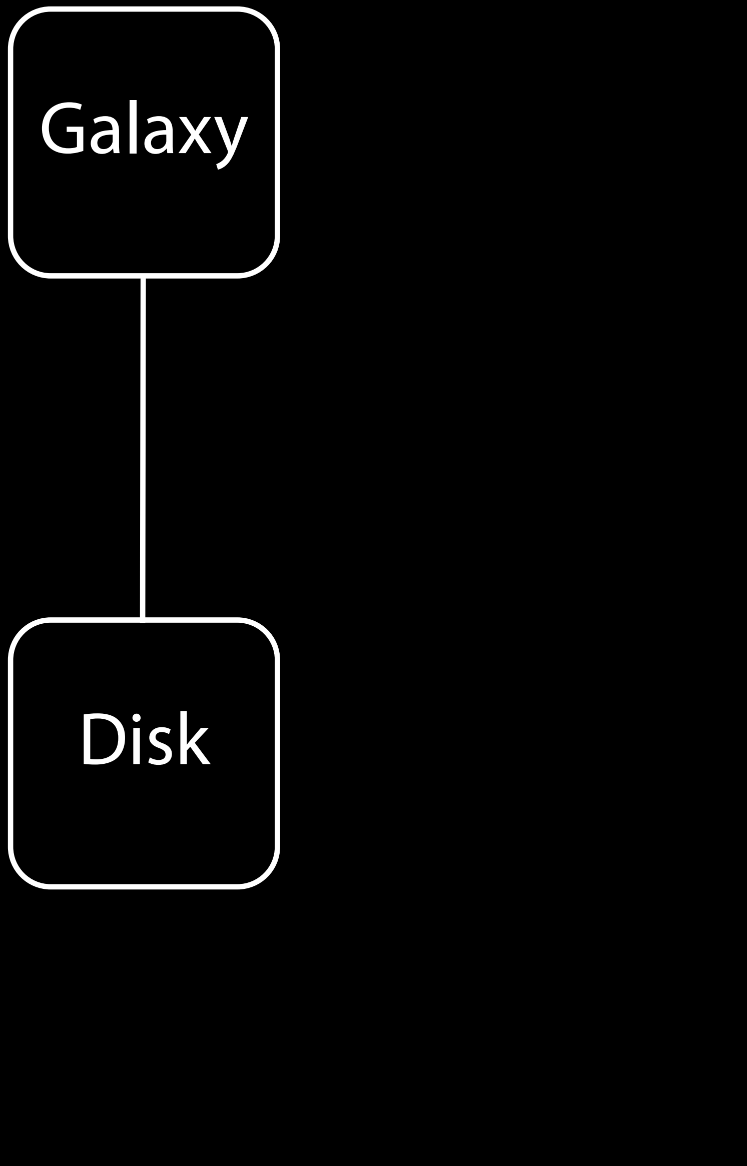
Galaxy Object Store



```
# Dataset files are stored in this directory.
file_path = database/files
```

Abstraction



Galaxy

Disk

Galaxy

Object Store

Disk

Disk

S3

iRODS

Accessing Data

```
>>> fh = open( dataset.file_path, 'w' )
>>> fh.write( ‘foo’ )
>>> fh.close()
>>> fh = open( dataset.file_path, ‘r’ )
>>> fh.read()
```

Accessing Data

~~>>> fh = open(dataset.file_path, 'w')~~

~~>>> fh.write(‘foo’)~~

~~>>> fh.close()~~

~~>>> fh = open(dataset.file_path, ‘r’)~~

~~>>> fh.read()~~

Accessing Data

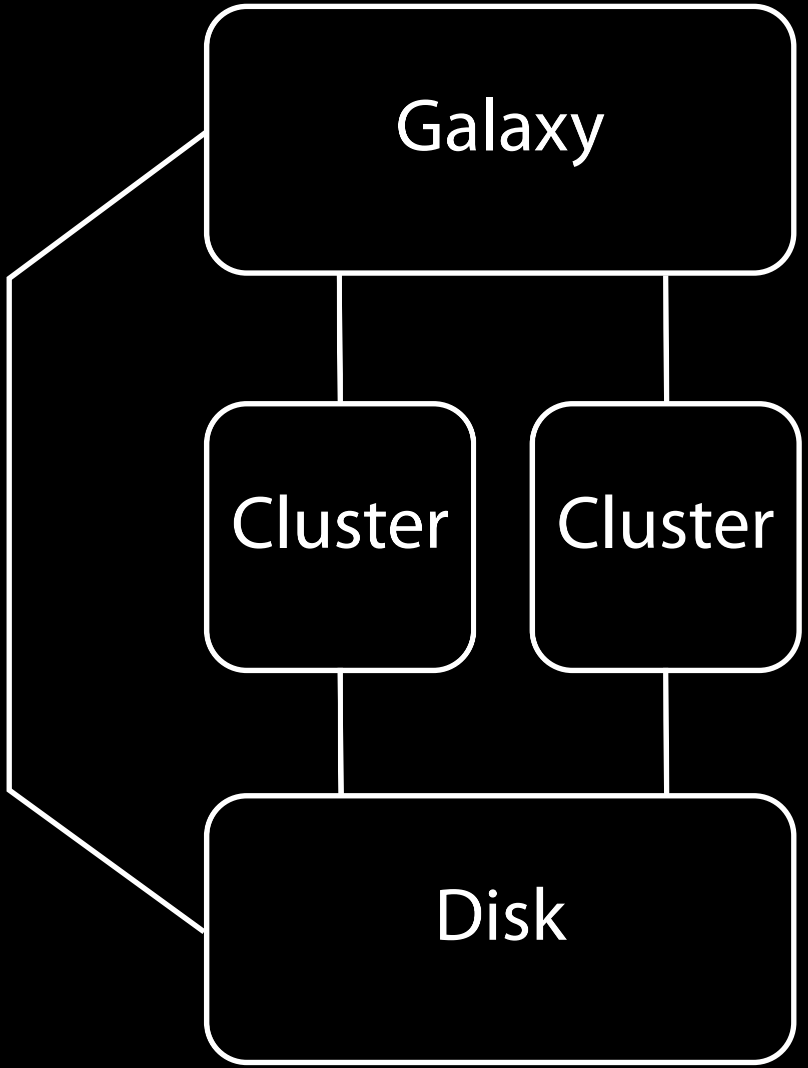
```
~~>>> fh = open( dataset.file_path, 'w' )~~
~~>>> fh.write( ‘foo’ )~~
~~>>> fh.close()~~
~~>>> fh = open( dataset.file_path, ‘r’ )~~
~~>>> fh.read()~~
```

```
>>> update_from_file( dataset, file_name=‘foo.txt’ )
>>> get_data( dataset )
>>> get_data( dataset, start=42, count=4096 )
```

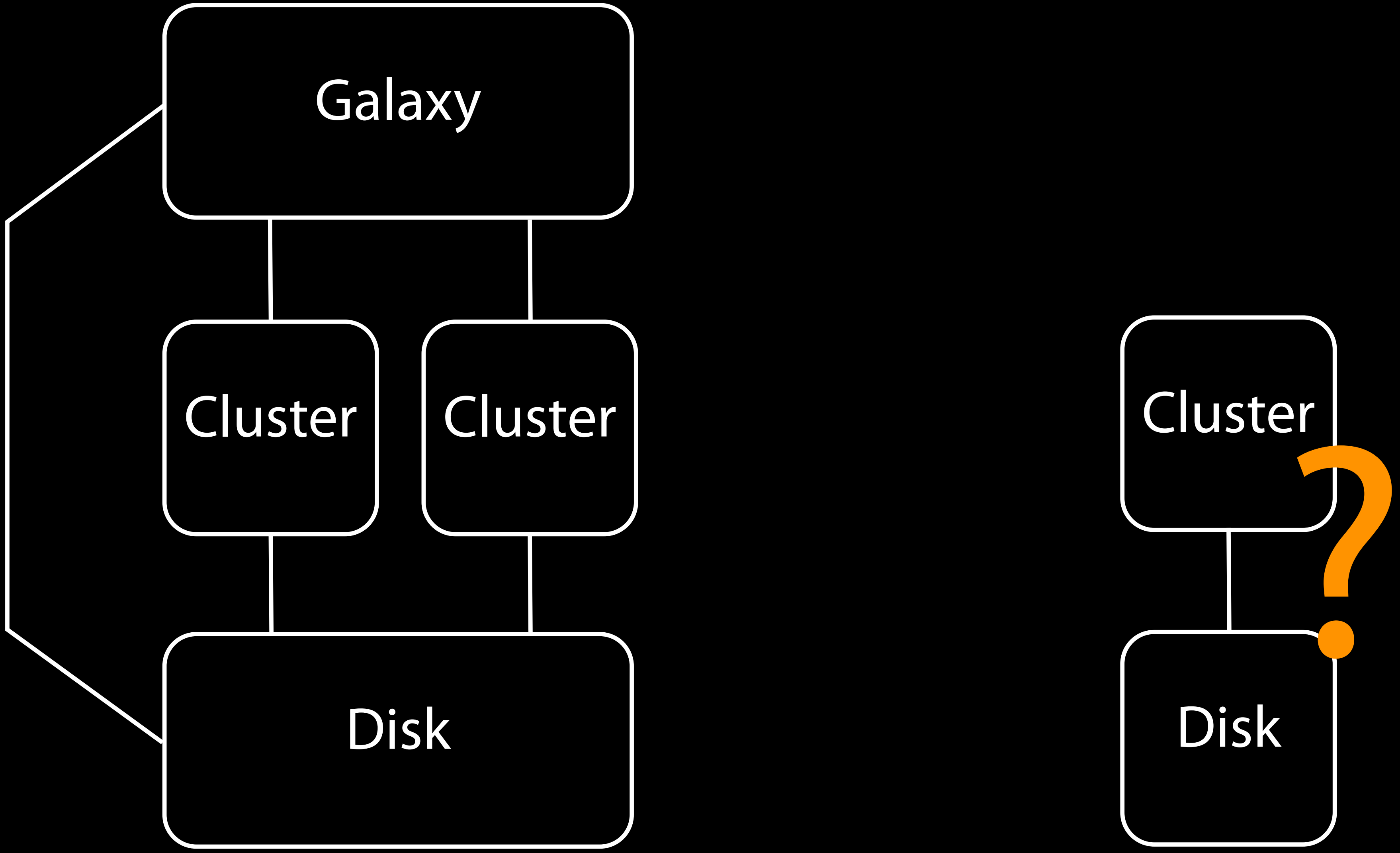
Real Benefits

- Grow beyond your original capacity
- Avoid migrating data from one resource to another
- Tier storage
- Let your users bring their own storage
- Write policies to define where data lives (soon!)

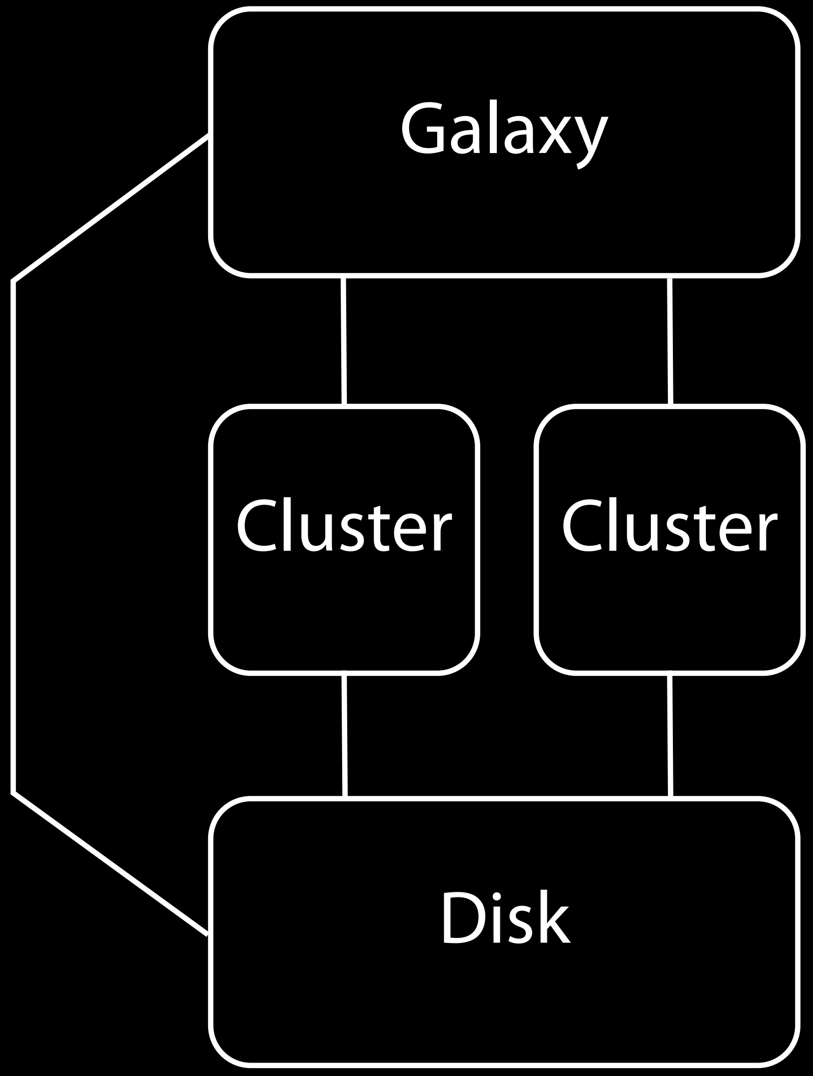
Use disparate resources?



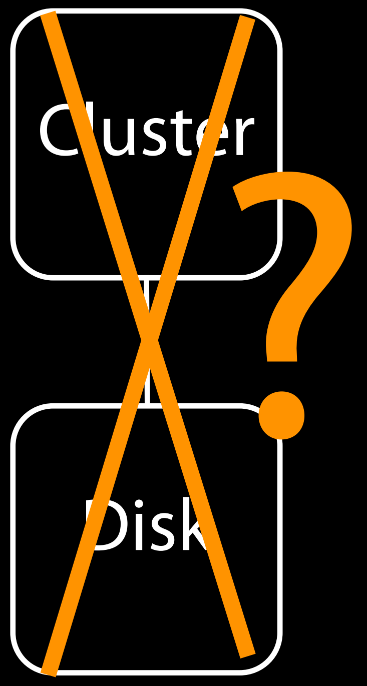
Use disparate resources?



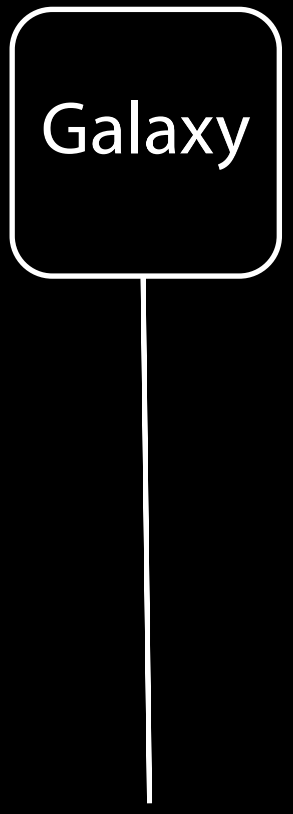
Use disparate resources?



NOPE



Use disparate resources



Galaxy

STUFF

(lots of cores, lots of storage pools, physically anywhere)

Scale

- Couple tightly with data management layers (e.g. iRODS, S3) instead of using FUSE mounts
- Remove the single filesystem IO bottleneck
- Put data as close as possible to the compute resources that need it