CMB-S4 and FABRIC

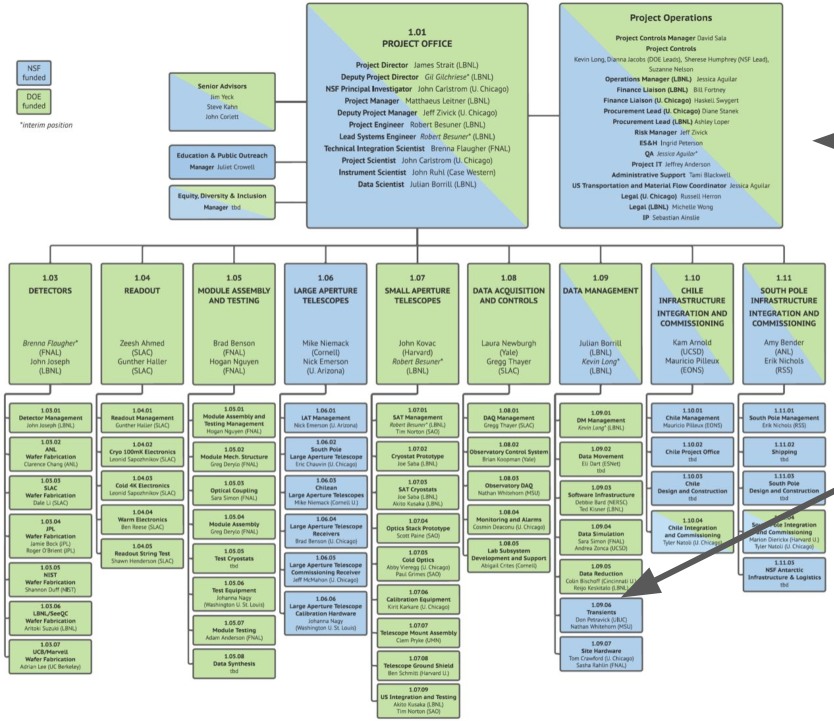
Greg Daues (NCSA)
Felipe Menanteau (NCSA)
Don Petravick (NCSA)

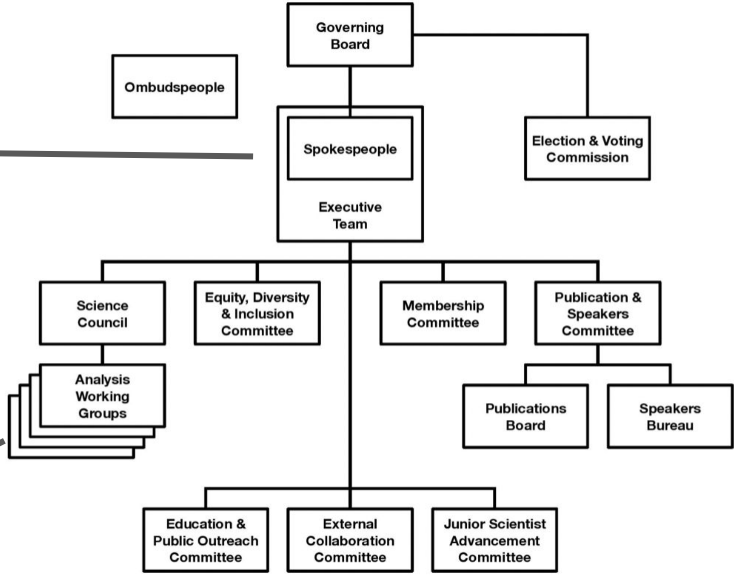
Fabric KNIT-6 workshop, April 26, 2023

Outline

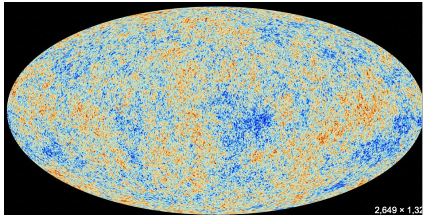
- CMB-S4 overview.
- Transients within the overall project
- Prototyping goals.
- What we did.
- High level view of FABRIC strengths.
- Summary.

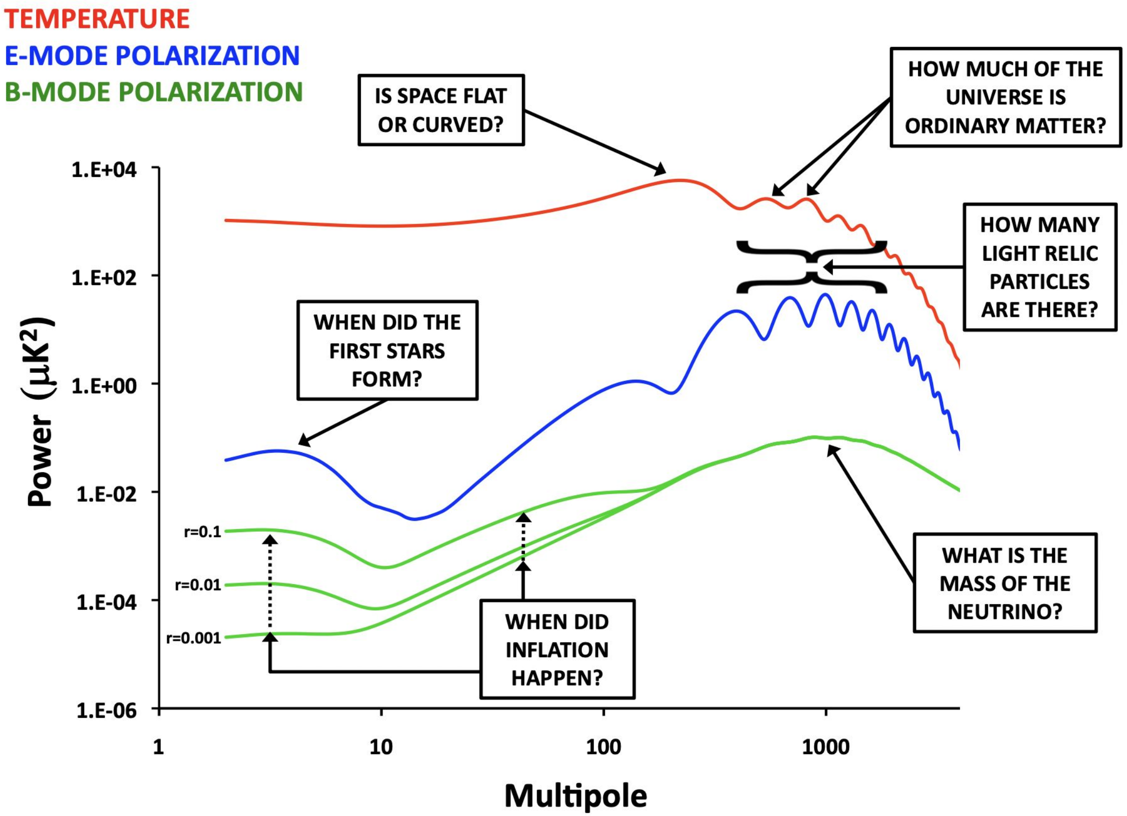
A Project and a Collaboration





Overview: CMB Science Potential





Sensitivity to different polarizations provides the basis for addressing distinct requirements.

Growth in Multipole corresponds to finer and finer spatial scales.

The Vertical axis in in micro Kevins Squared.

Four Science Goals, Two Sites, One Collaboration

Science Requirement 1.0: CMB-S4 shall test models of inflation by putting an upper limit on r of $r \leq 0.001$ at 95% confidence if $r = 0$, or by measuring r at a 5σ level if $r > 0.003$.

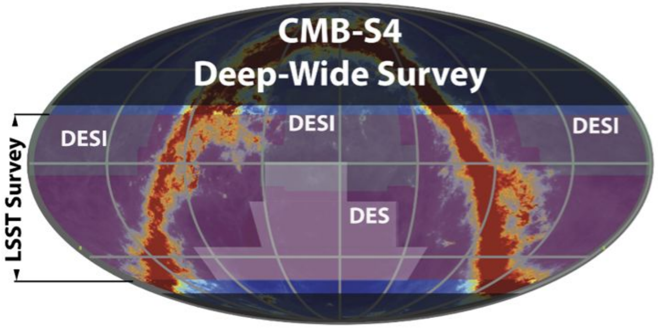
Science Requirement 2.0: CMB-S4 shall determine $N_{\rm eff}$ with an uncertainty ≤ 0.06 at the 95% confidence level.

Science Requirement 3.1: CMB-S4 shall detect at $\geq 5\sigma$ all galaxy clusters at $z \geq 1.5$ with an integrated Compton $Y_{\rm SZ,500} \geq 2.4 \times 10^{-5}$ arcmin^2 over at least 50% of the sky.

Science Requirement 3.2: CMB-S4 shall detect at $\geq 5\sigma$ all galaxy clusters at $z \geq 1.5$ with an integrated Compton $Y_{\rm SZ,500} \geq 1.2 \times 10^{-5}$ arcmin^2 over at least 3% of the sky.

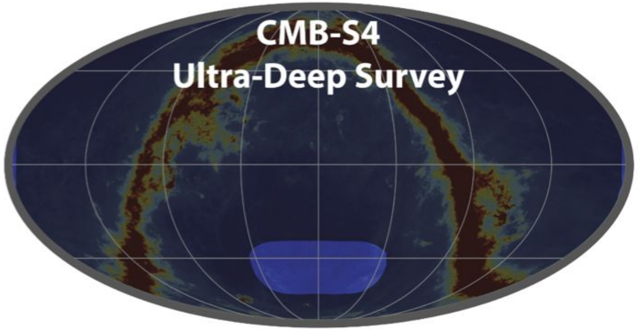
Science Requirement 4.1: CMB-S4 shall detect GRB afterglows brighter than 30 mJy at 93 and 145 GHz over at least 50% of the sky and enable followup by issuing timely alerts to the community.

Science Requirement 4.2: CMB-S4 shall detect GRB afterglows brighter than 9 mJy at 93 and 145 GHz over at least 3% of the sky and enable followup by issuing timely alerts to the community.



Observed from Chile

Large area survey motivated by $N_{\rm eff}$, matter mapping, and time domain science and enabled by the mid-latitude site

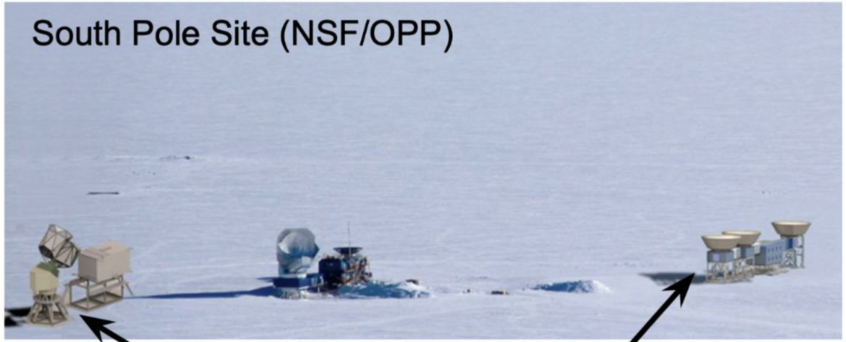


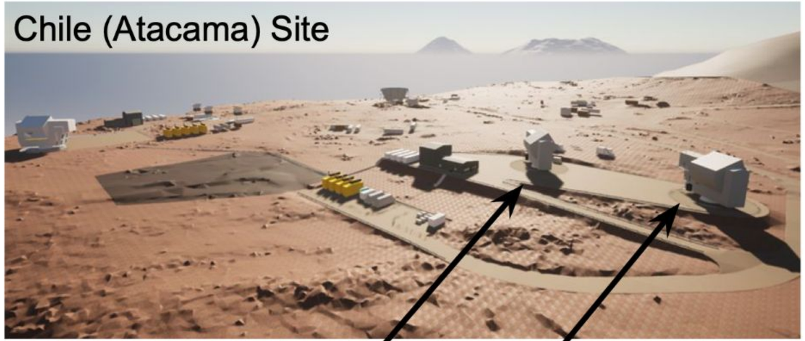
Observed from South Pole

Small area survey primarily targeting inflationary gravitational waves, enabled by the sky coverage, low horizon blockage, and ultra stable atmosphere of the polar site.

Requirements 4.1, 4.2 are the *driving requirements* for transients, and enable detection of other phenomena.

Instrumentation

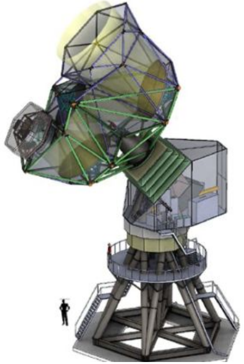


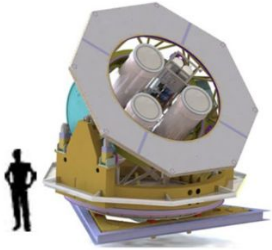


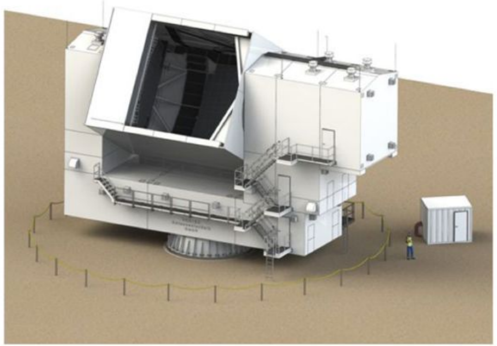
1 Large Aperture
(5 m) Telescope

3 Small Aperture Telescopes
(9 0.5-m aperture optics tubes)

2 Large Aperture (6 m)
Telescopes



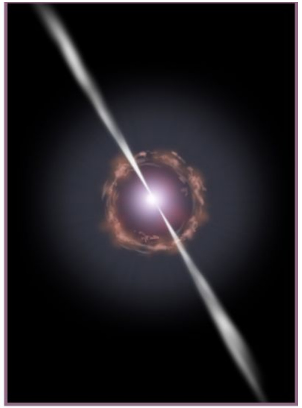




Transient Phenomena and Community

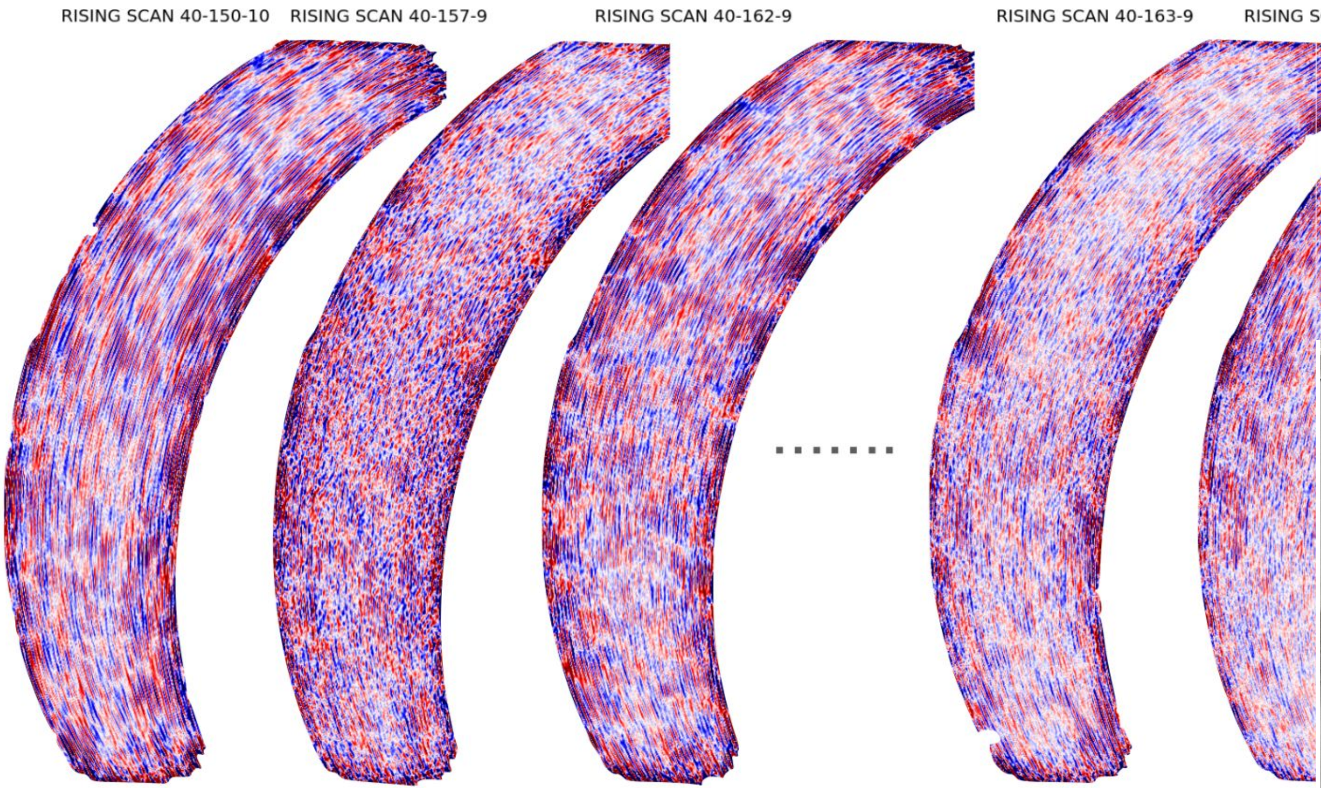
Astro2021: CMB_S4 will produce data sets of unprecedented sensitivity, cadence and spectral coverage…. **Seek broad engagement with astronomers** … **maximize opportunities for transient science**…..

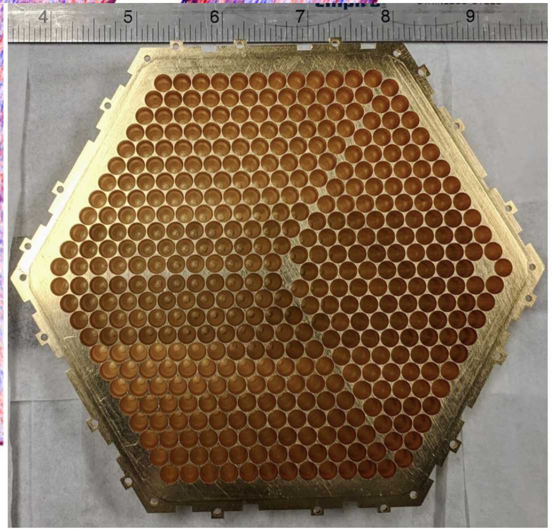
- Stellar Flares
- Tidal Disruption Events
- Active Galactic Nuclei
- X ray Binaries
- Classical Novae
- Planetary Nebulae
- SN Type 1 progenitors
- Magnetars
- Solar system Objects



The infrastructure we are prototyping in FABRIC its to detect and announce announce initial detections to the community. In era of CMB-S4 we are in "discovery space"

DC0 Simulations

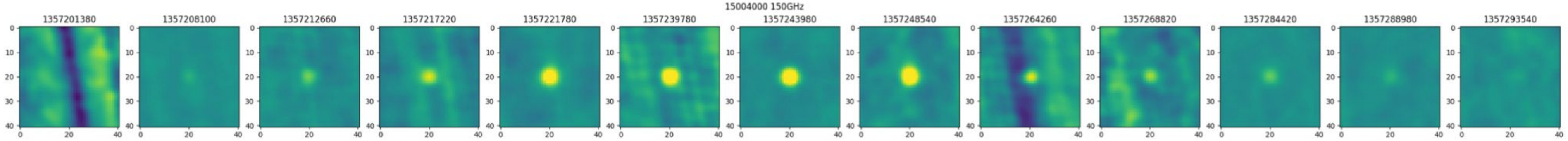


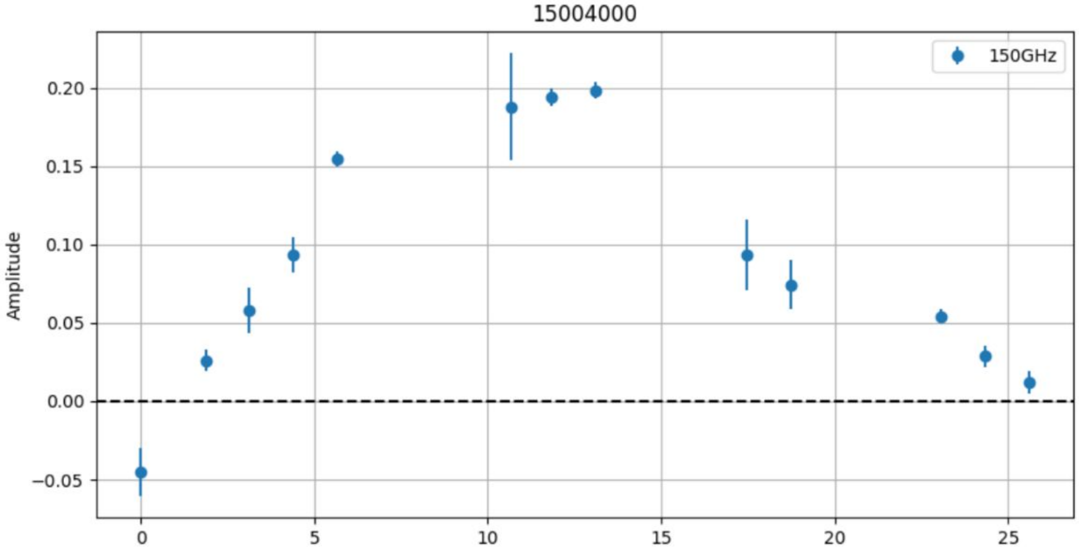


Prototype horn array consisting of 430 spline-profiled horns machined into gold-plated aluminum.

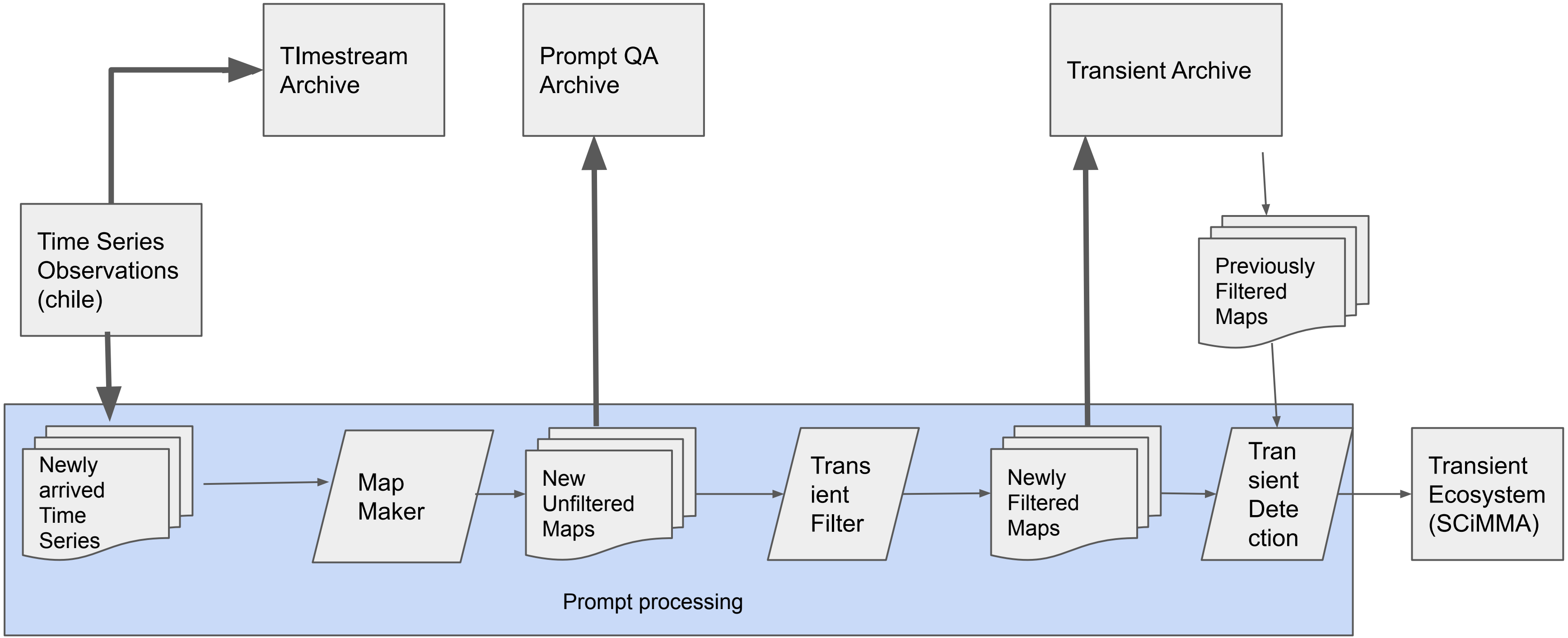
Source Detection and Light Curve Recovery

Using the SPT-3G Transient Pipeline





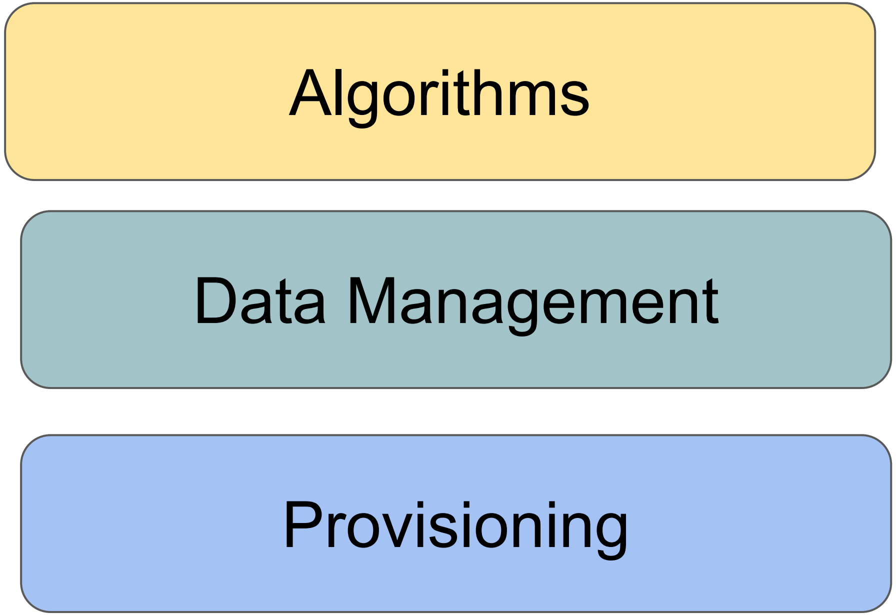
Current (Jan 2023) Baseline (Chilean)



FABRIC as a prototype for “prompt processing”

- Most computing is large scale, and will be done at National Centers using Batch techniques.
- Transients have a distinct requirements for timeliness.
  - Alerts need to be delivered on time scales that allow follow-up observations to occur while the signal is still present.
  - Fabric is an environment…
    - that allows prototyping of the prompt processing.
    - has many interesting properties as well.
- We have the concerns of production people, not network researchers.
  - A distributed computing mesh supporting prompt computations, and good availability is roughly our desiderata.
  - We want to control the setup from an experiment-controlled machine, so that it can be controlled from the experiment’s global environment.

Unpacking what was done



Algorithms are developed on the Illinois Campus Cluster.

Code is Containerized.

Data management stages data into Fabric, and uses HTCondor to run Jobs.

Various Fabric Sites, play roles of Observatory, etc…

Provisioning allocates resources within FABRIC, health and status, and support.

Provisioning Desiderata

- Provide distributed resources to support prompt processing prototyping.
- Implement the provisioning repeatedly, and under configuration control.
  - Eg. meet “terraform” type expectations.
- Agile - Integrate resources needed by Data Management as they emerge, adjust to new versions of FABRIC.

```
plan                show plan, but do not call FABRIC APIs
apply               instantiate the plan by calling FABRIC APIs
delete              delete a slice, if it exists
print               Print information about nodes and networks in a slice
json                emit json describing slice
mass_execute        Execute command(s) on all nodes in the slice. if the
                    command begins with @ treat the argument as a file of
                    commands.
execute             execute a command(s) on a specfic node
debug               call a slice into memory and start the debugger
slices              print the name of all my slices
resources           experimental -- take a peek at resources
health              test topology health by having each node ping all the
                    others
aliases             Print out alias for ssh commands to each node. Note that
                    FABRIC nodes with IPV6 addresses cannot be reached from
                    sites not supporting IPV6. "-u prints correponding unalias
                    commands.
dns                 Setup /etc/hosts so all node can addess all others by name.
                    These names can be made up and don't leak out of FABRIC.
renew               renew a slice so it sticks around
format              format fabric-provided storage at a site.
```

Declarations Toy Example

Declarative approach: Shadow objects collect attributes, relationships for fabric objects.

Lightweight enough to deal with evolution of the research infrastructure.

Canonical enough so that data management gets what it needs, and has an abstract model.

```
from  fabric_objects import *

image =  'default_rocky_8'
slice = CfSlice(__name__, rocky_linux_workaround=False)

node1 = CfNode(slice, 'CMBS4Node1', image,
                disk=100, cores=32 ram=128, site='TACC',
               storage="CMB-S4-phase-one")
net1 = CfL3Network(slice, 'net1', type='IPv4')
CfNic(slice, "Node1.NIC1", node1, net1)

node2 = CfNode(slice, 'CMBS4Node2', image, site='NCSA')
net2 = CfL3Network(slice, 'net2', type='IPv4')
CfNic(slice, "Node2.NIC1", node2, net2)

#  Leave these function calls in place
def plan(): slice.plan()
def apply():slice.apply()
```

Data Management Desiderata

- Show we can compute using on-demand, performant tool kit
  - Containers
  - HTCondor
  - File transfer
  - Reproducible
  - Inflate/Deflate Slice - compute system (SDI)
- Show we can control the overall system from CMB-S4 controlled resources.
  - Slice generation through Fabric API
  - Stage Apptainer images, HTCondor configuration to Slice machines

Data Management Desiderata

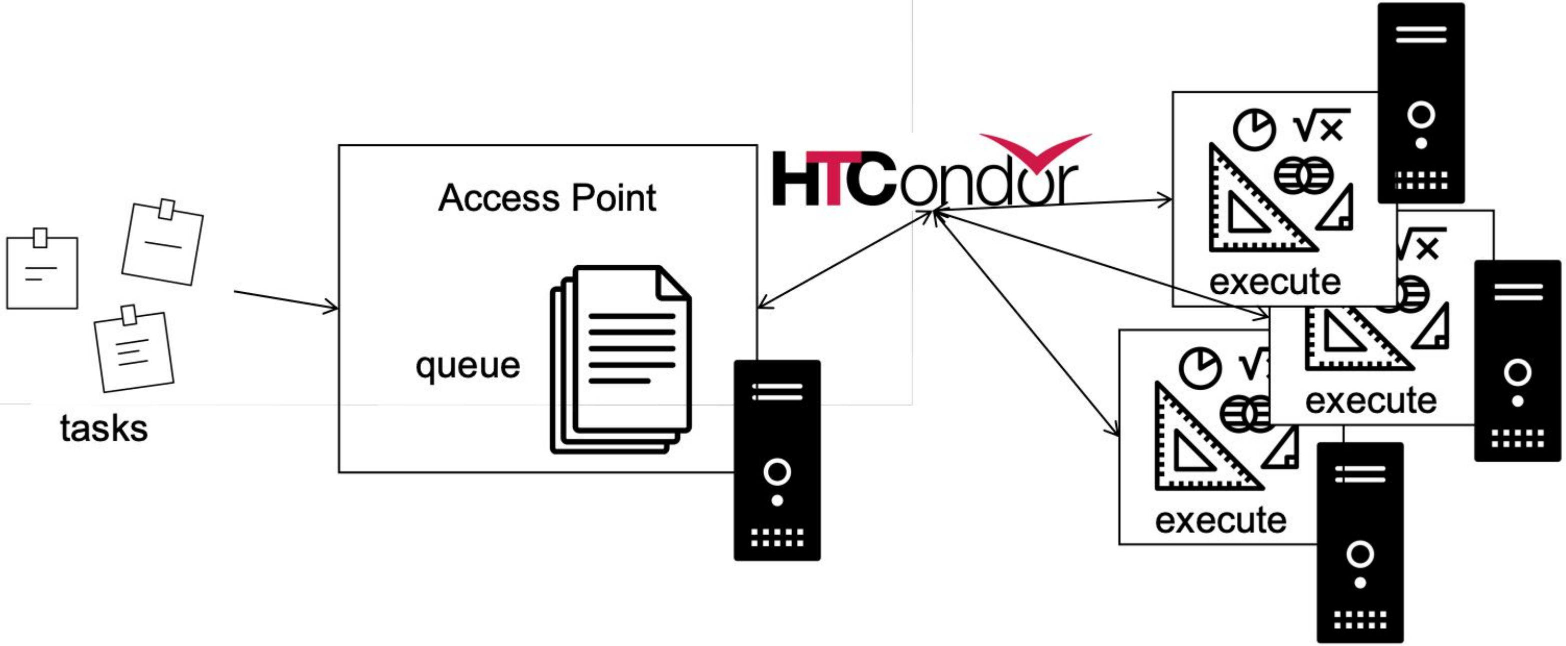
- Model the problem by using several Fabric sites
  - Chilean Observatory — FIU
  - A CMB-S4 project facility — NCSA
  - Prompt processing sites — STAR, TACC, UCSD, …
- Fabric Slice / VMs
  - Add N VMs (N sites) to Slice
  - Add L3 network for each VM
  - Add Routes between L3 networks

L3 networks with Routes

```
# For each pair of nodes:

node1 = slice.add_node(name=node1_name, site=site1)
node2 = slice.add_node(name=node2_name, site=site2)
iface1 = node1.add_component(model='NIC_Basic',
name=node1_nic_name).get_interfaces()[0]

iface2 = node2.add_component(model='NIC_Basic',
name=node2_nic_name).get_interfaces()[0]

...

net1 = slice.add_l3network(name=network1_name, interfaces=[iface1], type='IPv4')

net2 = slice.add_l3network(name=network2_name, interfaces=[iface2], type='IPv4')

...

node1.ip_route_add(subnet=network2.get_subnet(), gateway=network1.get_gateway())

node2.ip_route_add(subnet=network1.get_subnet(), gateway=network2.get_gateway())
```

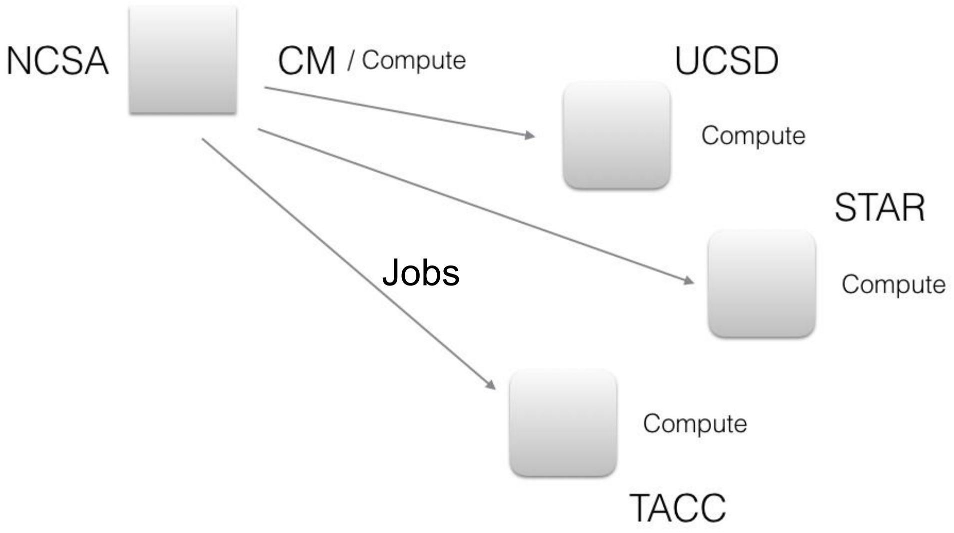
HTCondor on Many Computers

Slide: HTCondorWeek 2022 - HTCondor User Tutorial



HTCondor Computing Pool on FABRIC

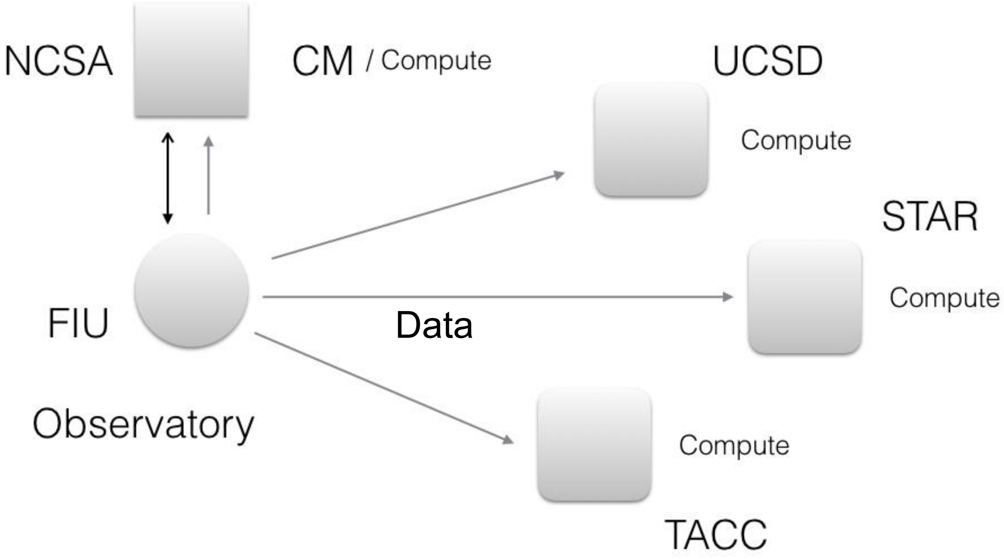
- All nodes configured for an HTCondor role (Manager, Execute)
- Compute Nodes communicate with a Central Manager using L3 networks/routes
- Jobs/Tasks are assigned by the Central manager to the pool of nodes



Data Flow

- A main workflow on Central manager discovers available data at FIU / Observatory

- All jobs on compute nodes copy data from persistent storage at FIU / Observatory using L3 networks/routes (data plane)



Future Work

- Extend Slices with IPv4Ext IPv6Ext service
- Integrate Fabric Slices to CMB-S4
  - NERSC site and storage
  - Illinois / NCSA site, storage
  - Github repositories, etc.
- Full Transients demonstration for Sept reviews
  - To be verified by Science Working Groups

High Level View of FABRIC Strengths

- Private, closed high bandwidth experiment-specific, distributed private networks. Mitigating the need for all distributed protocols to be hardened to the level required on the open internet,Implying greater choice at the application level, and providing an environment to support distributed software development, while security features are being implemented.

- Information security services on par- with current (some) on-prem security services. Unlike on-prem security services, the meta-facility would have the distributed scope of the experiment as its prime concern. FABRIC currently provides bastion services, and hardened access to its Management network. Future infrastructure could provide scanning, log retention, log analysis, vulnerability scanning and other services currently available only in the scope of on-prem, or for a (substantial) fee for commercial cloud-based deployment.

- Supporting software-as-infrastructure provisioning model, among distributed sites. This infrastructure-as-code model supports reliability by the ability to re-deploy the private network and computational resources to new sites. This allows long-running experiments to maintain operations with minimal disruption during hardware upgrades and allows all experiments to have higher availability than if locked into a particular site.

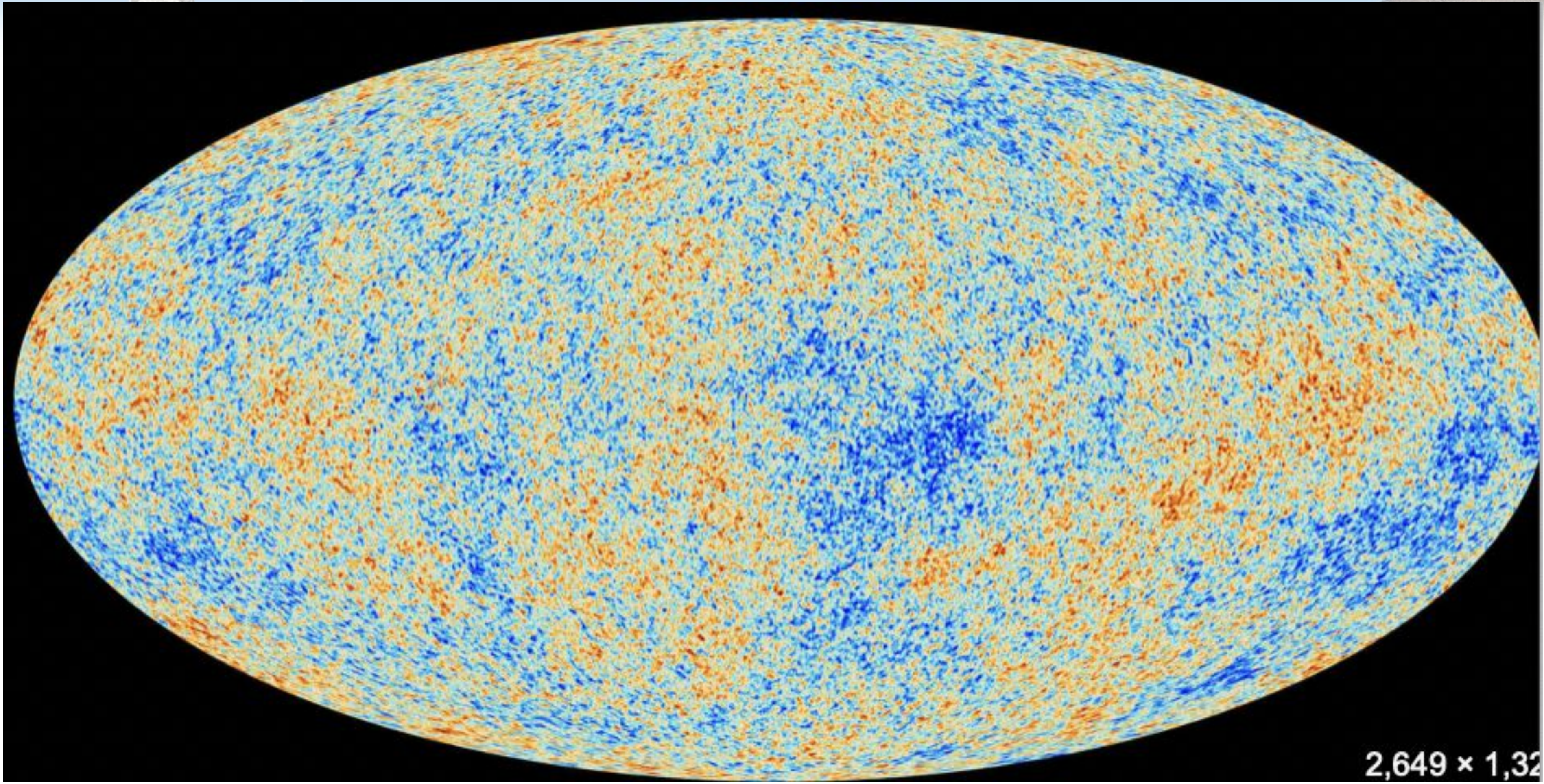
- Satisfying common operations needs. As an example of this, FABRIC has a plug-in system level monitoring capability, allowing system level monitoring of all nodes in a deployment. (e.g work in with peers in a cooperative community)

- For large project, the ability to built a distributed proxy for sites while the actual physical locations where the experiment will run are not yet constructed. Using FABRIC we've been able to have FIU play the role of "observatory", etc…

Summary

- We are being successful prototyping on FABRIC.
  - We can use the normal tools used to build data management systems.
  - We are enjoying have a distributed network, and not developing a security model for it…
- Fabric has Good Bones -- Interfaces/APIs are well thought out.
  - It was not hard at all to build a (semi) declarative interface
- Beyond basic capabilities, FABRIC has features we'd have to build ourselves bastions, etc.
- While FABRIC is a testbed, our investigations indicate an analogous production cyber infrastructure would support the "prompt processing" needs of CMB-S4.
  - Such an infrastructure would have potential role in building a coherent international computing fabric for CMB-S4.

Spares


2,649 × 1,32