Roland Haas Eliu Huerta

Gravity Group and SEAS Group
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

BOSS-LDG: A Novel Computational Framework that Brings Together Blue Waters, Open Science Grid, Shifter and the LIGO Data Grid to Accelerate Gravitational Wave Discovery

Accepted to 13th IEEE eScience International Conference, NZ, October 2017

Huerta, Haas, Fajardo, Katz, Anderson, Couvares, Willis, Bouvet, Enos, Kramer, Leong, Wheeler

NCSA, SDSC, LIGO

Container Analysis Environments Workshop
NCSA, August 14th 2017

Outline

Motivation

Science Case

LIGO DATA GRID meets OPEN SCIENCE GRID

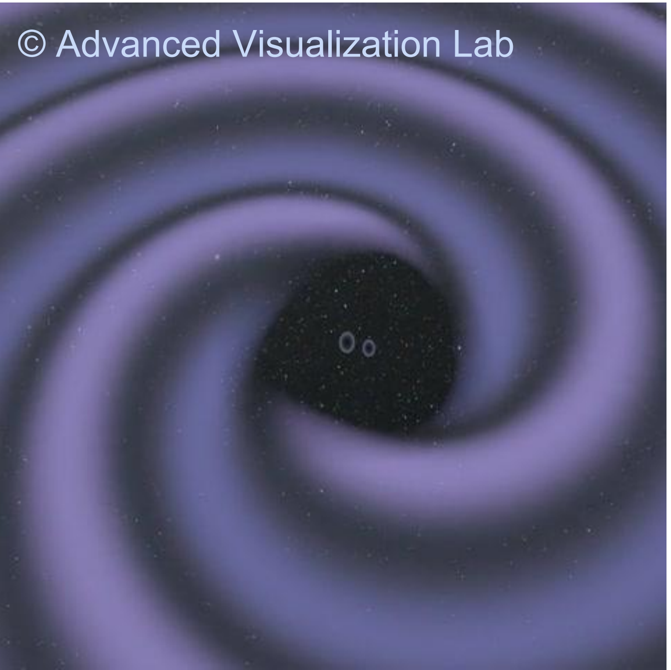
Gravitational wave astrophysics in High Performance Environments

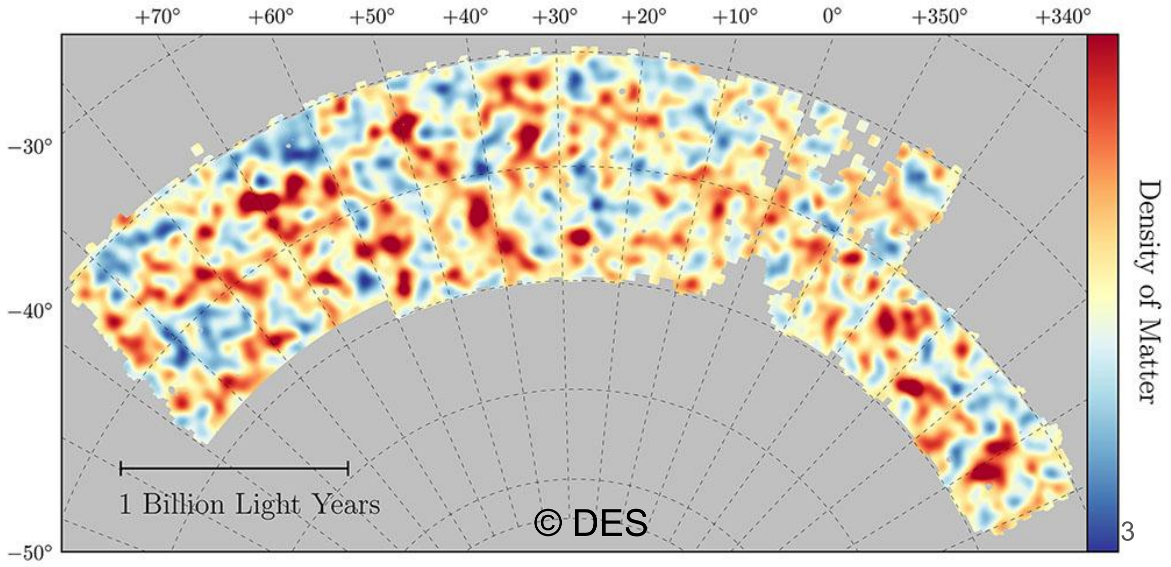
Connecting Blue Waters to the LIGO DATA GRID

SHIFTER

Motivation (*astrophysicists' perspective*)

HPC resources have been traditionally used by astrophysicists in the context of large scale numerical simulations: numerical relativity, cosmology, DES, etc





Science case: How to detect gravitational waves

- Convince NSF to spend >$1e9
- Build two 4km long, 1m diameter vacuum tubes
- Add LASERs
- Hire ~1000 scientists and engineers
- Wait for a gravitational wave to come by and stretch / shrink the vacuum tubes
- **Dig** for signal in data stream

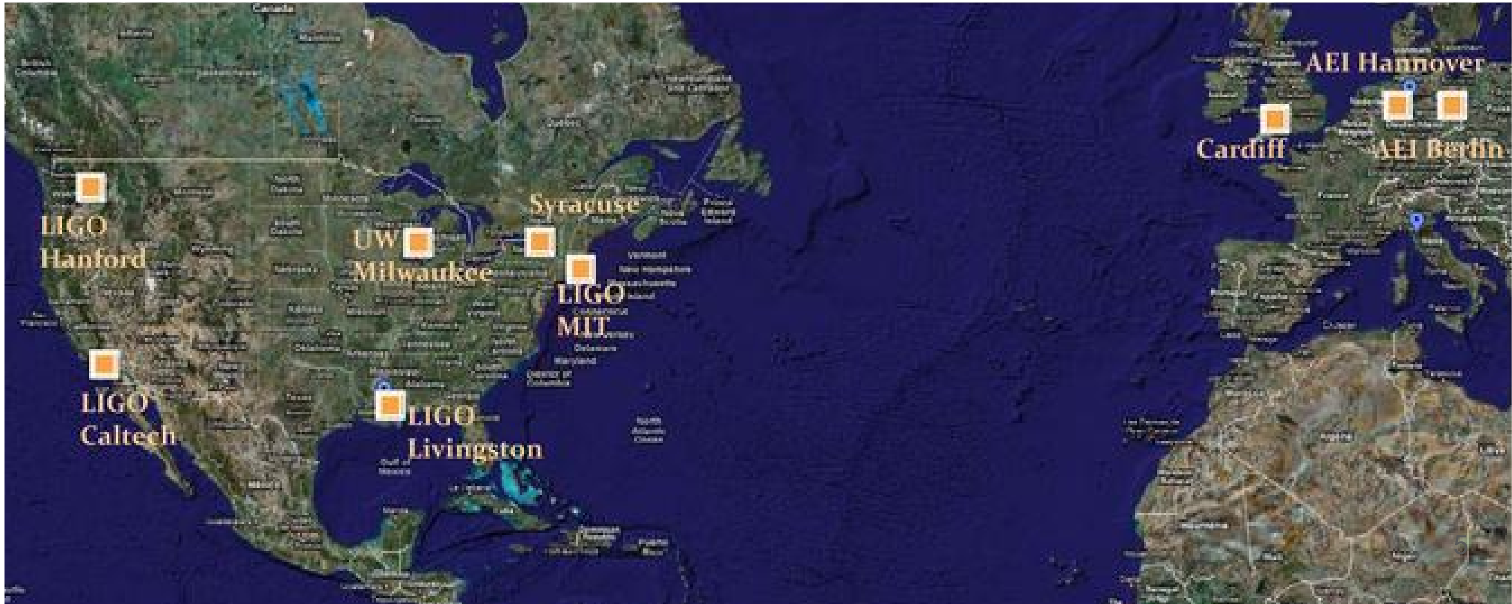
- lightweight data transfer
- computationally intensive analysis: 100s of TBs of data, hundreds of millions of CPU-hrs per discovery

Science case

Gravitational wave data analysis is very computationally intensive

Traditionally done using the LIGO Data Grid (LDG)

High Throughput Computing (HTC) centers in the US and Europe



Science case

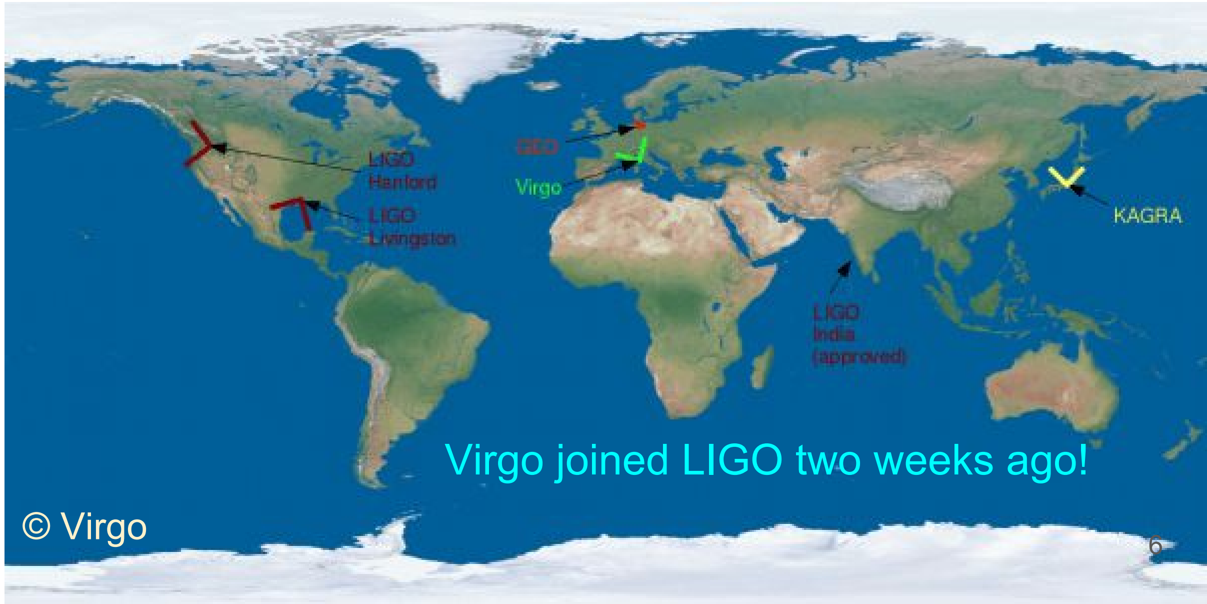
Gravitational wave data analysis is very computationally intensive

First discovery campaign (O1) required over 90M SUs (1SU=1 Xeon E5-2670 CPU-core-hour)

Demand for computational resources is increasing:

more detectors coming online, requiring more computational power

longer detection campaigns: O2 will consume 3x O1's SUs



Virgo joined LIGO two weeks ago!

© Virgo

LIGO software stack

- Set of reference operating systems that are officially supported
  - Scientific Linux 7
  - Debian 7 / 8 (Wheezy / Jessie)
- LIGO software stack
  - Primarily RPM and Debian packages
  - Compiling full stack from hand is hard
  - HTC workflow / HTCondor + Pegasus
  - 1.5e6 lines of C99 code in base package
- Analysis code software requirements
  - Pegasus requires Java
  - PyCBC requires many python packages
  - Dominant cost is a **fast Fourier transform**
- Typically provided by LDG centers but not HPC centers

LIGO DATA GRID meets the OPEN SCIENCE GRID

Running a typical LDG center requires ongoing sysadmin effort at the minimum level of a full FTE

Open Science Grid (OSG): a means to connect LSC scientists with disparate computing resources outside the LDG

Additional resources are always welcome!

What are the requirements to use the OSG?

LIGO DATA GRID meets the OPEN SCIENCE GRID

LIGO workflows need to be compatible with OSG requirements

Software requirements: single threaded, less than 2GB in memory, each invocation should run for 1-12hrs

Jobs should handle multiple re-starts, binaries should be statically linked, no special python modules

Input and output data for each should be less than 10Gb

No MPI communication, no shared file system, no complex or licensed software

PyCBC satisfies these requirements with a mature workflow manager: Pegasus

LIGO DATA GRID meets the OPEN SCIENCE GRID

OSG resources contributed about 20% of computational resources during O1: more than any US LDG site!!

OSG was used in 2015 to connect the LDG with XSEDE sites.

Seamless connection since XSEDE machines and LDG use similar OS

XSEDE experts re-wrote LIGO code to make it efficient in HPC environments

The frontier: connect the LDG to Blue Waters via OSG

LDG has several Tiers:

Tier-1: large center, available to all LIGO Scientists

ATLAS	LIGO	UNIVERSITY of WISCONSIN UWMILWAUKEE	IUCAA, India
31000 cores	18780 cores	3392 cores	2520 cores

Tier-2/3: small data center, available to whitelisted users

LDG Tier-3 center used in 2015 to connect the LDG to XSEDE

The frontier: connect the LDG to Blue Waters via OSG

LDG has several Tiers:

Tier-1: large center, available to all LIGO Scientists

ATLAS	LIGO	UNIVERSITY of WISCONSIN UWMILWAUKEE	IUCAA, India
31000 cores	18780 cores	3392 cores	2520 cores

Tier-2/3: small data center, available to whitelisted users

LDG Tier-3 center used in 2015 to connect the LDG to XSEDE

BLUE WATERS
724480 cores

Dwarfs total of all existing LIGO resources

18,828,441,574

The frontier: connect the LDG to Blue Waters via OSG

LDG Tier-3 center used in 2015 to connect the LDG to XSEDE was private system

Needs to be re-implemented for Blue Waters use

NCSA and LIGO Lab scientists configured a Tier-1 LDG center with OSG submission capabilities

Configure environment to handle heterogeneous workflows

Configure PyCBC to optimally exploit resources at this LDG Tier-1 OSG center

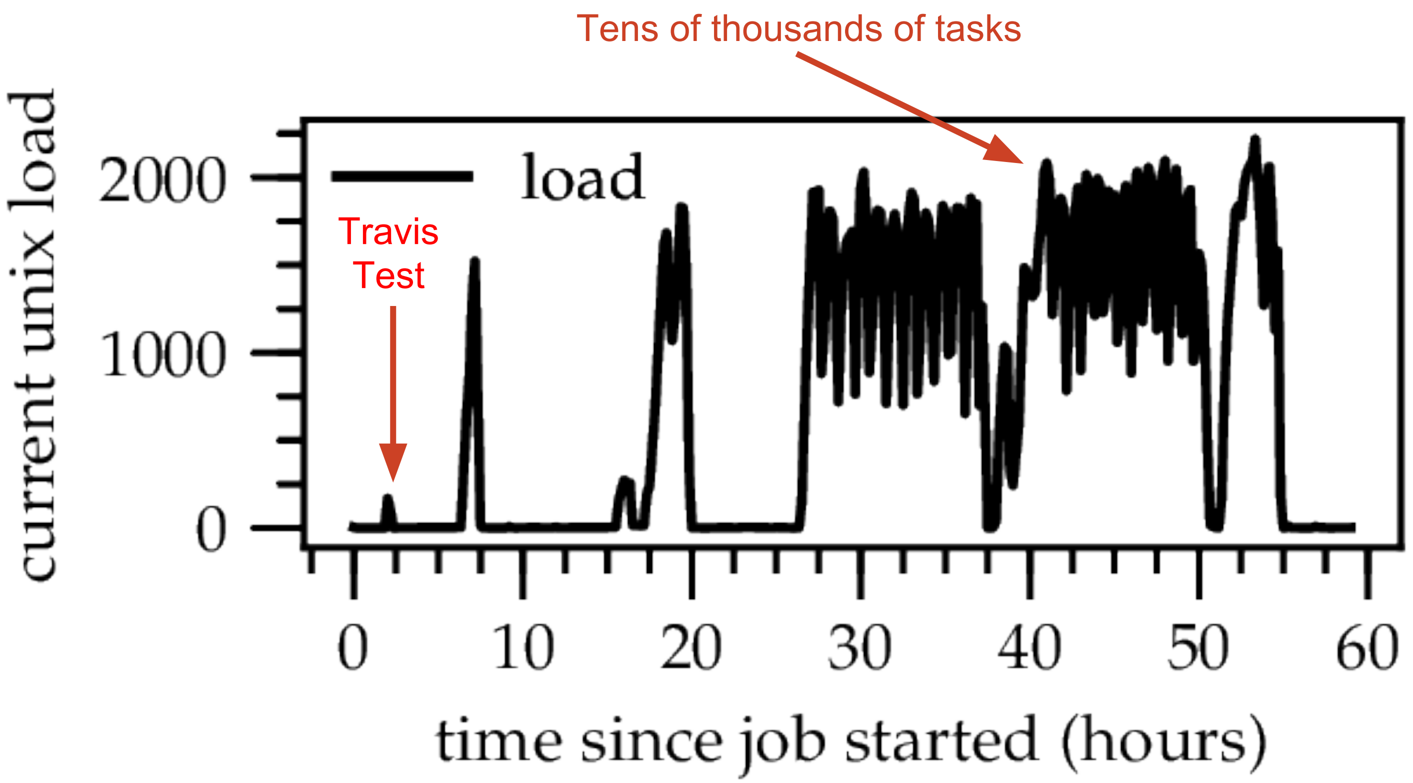
NCSA

Caltech

ABILENE CHRISTIAN UNIVERSITY 1906

The frontier: connect the LDG to Blue Waters via OSG

Connection between LDG Tier-1 to OSG done!

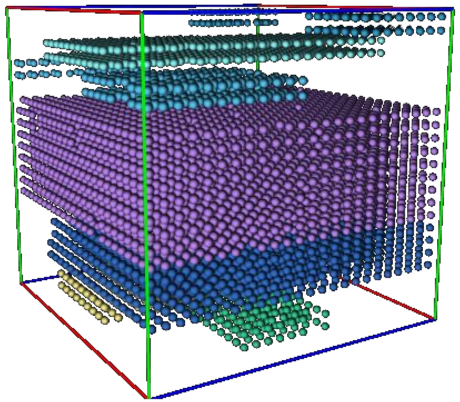


HTCondor
High Throughput Computing

While this was happening in HTC land…

transformative developments were happening at NCSA to connect Blue Waters to the LDG

Connecting Blue Waters to the LIGO DATA GRID



- Blue Waters is a classic HPC system
  - Large, shared file system
  - Fast interconnect
  - Designed to handle **few, large jobs**
  - Connection to the LDG is harder than for XSEDE
  - Roads you do not want to explore:
    - Install LIGO software cluster-wide
    - Maintain rapidly evolving code: it is already hard to do this at LDG sites!!

Configure Blue Waters as an OSG compute element

Run LIGO workflows on Blue Waters without installing LIGO software natively

LIGO Data Grid

Open Science Grid

SHIFTER

BLUE WATERS

Software developed to run LIGO workflows in XSEDE systems was adapted to use it in Blue Waters

We did not have to reinvent the wheel, just tune it to work on Cray Systems (We like the HPC community!)

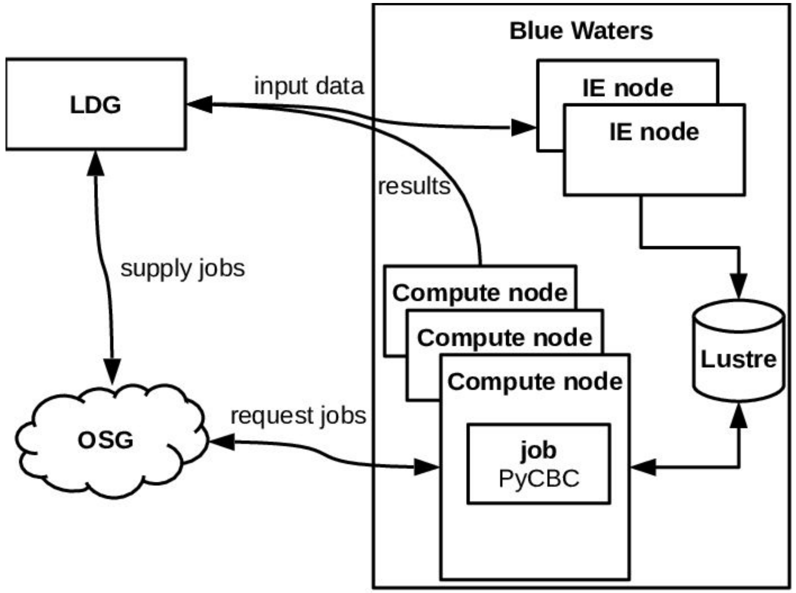
Great team work, several research powerhouses brought together to tackle this problem

NCSA

SDSC

Caltech

ABILENE CHRISTIAN UNIVERSITY 1906



- Condor scheduler runs on LDG site at Caltech
- Blue Waters configured as OSG site for Condor
- Glidein pilot jobs submitted on Blue Waters contact scheduler at Caltech, ask for work and report results
- Data transferred from Caltech to Blue Waters and back via gridftp

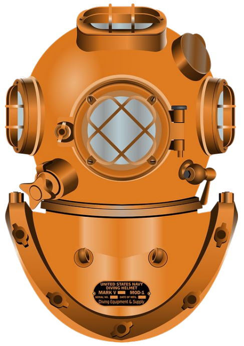
How Shifter is like docker...

- Shifter deploys containers in HPC environments
  - Supports docker image files
  - Native MPI + Gemini interconnect accessible
  - Supported by the scheduler (and SEAS)

BLUE WATERS

SHIFTER

docker



How Shifter is almost, but not quite, entirely unlike docker...

BLUE WATERS

SHIFTER

- Shifter deploys containers in HPC environments
  - Supports docker image files
  - Native MPI + Gemini interconnect accessible
  - Supported by the scheduler (and SEAS)

but

  - Images are ext3 disk images, not layered file systems
  - Shifter inserts/removes files e. g. in /etc/passwd
  - Images are **read-only**, no output to /var/log, no local disk!
  - /home is hidden by BW's native /home
  - Programs run as native user account, cannot be changed
  - Only in Cluster-Compatible-Mode does a Cray look like a white-box cluster

Maybe not your cup of tea … but it's definitely workable

Glidein WMS

- OSG software is hosted in CVMFS which is build on top of FUSE
  - BW hosts a copy of ATLAS CVMFS in /scratch (or use Parrot), FUSE is not available
- PyCBC pulls data files from OSG via globus-url-copy
  - Currently uses RSIP to compute nodes which limits speed
  - Currently pulls from Caltech LIGO cluster which is slow, should pull from primary server in Nebraska
  - Want to use 3rd party transfers through BW IE nodes to stage files outside of jobs
- aprun ties threads to cores, unexpected when used to OSG
  - mustn't forget "`-d 32`" option to `aprun` otherwise performance is very poor
- BW native file systems are slow to handle many small files
  - store some files in /dev/shm in containers

Worked out be Edgar Fajardo (SDSC) see his OSG all hands talk 2017

LIGO Data Grid

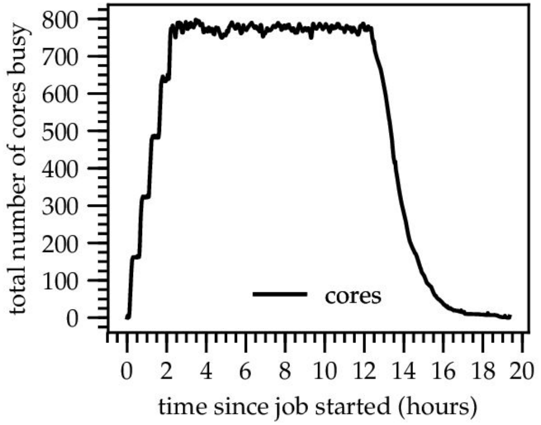
Open Science Grid

SHIFTER

BLUE WATERS

Blue Waters is a **big gun,** careful what you point it at

- “Small” BW jobs involve hundreds of nodes
- All nodes become available at the same time
- Each PyCBC job downloads ~400MB of data
- Startup can easily overwhelm the remote system
- Need to stagger pilots or limit startup rate in Condor / Pegasus



LIGO Data Grid

Open Science Grid

SHIFTER

BLUE WATERS

Connected LIGO Data Grid to Blue Waters through OSG

Showed that we can run PyCBC to completion using OSG+Blue Waters

Results agree with LIGO’s published results

Frameworks is not limited to LIGO; any OSG workflow can be used

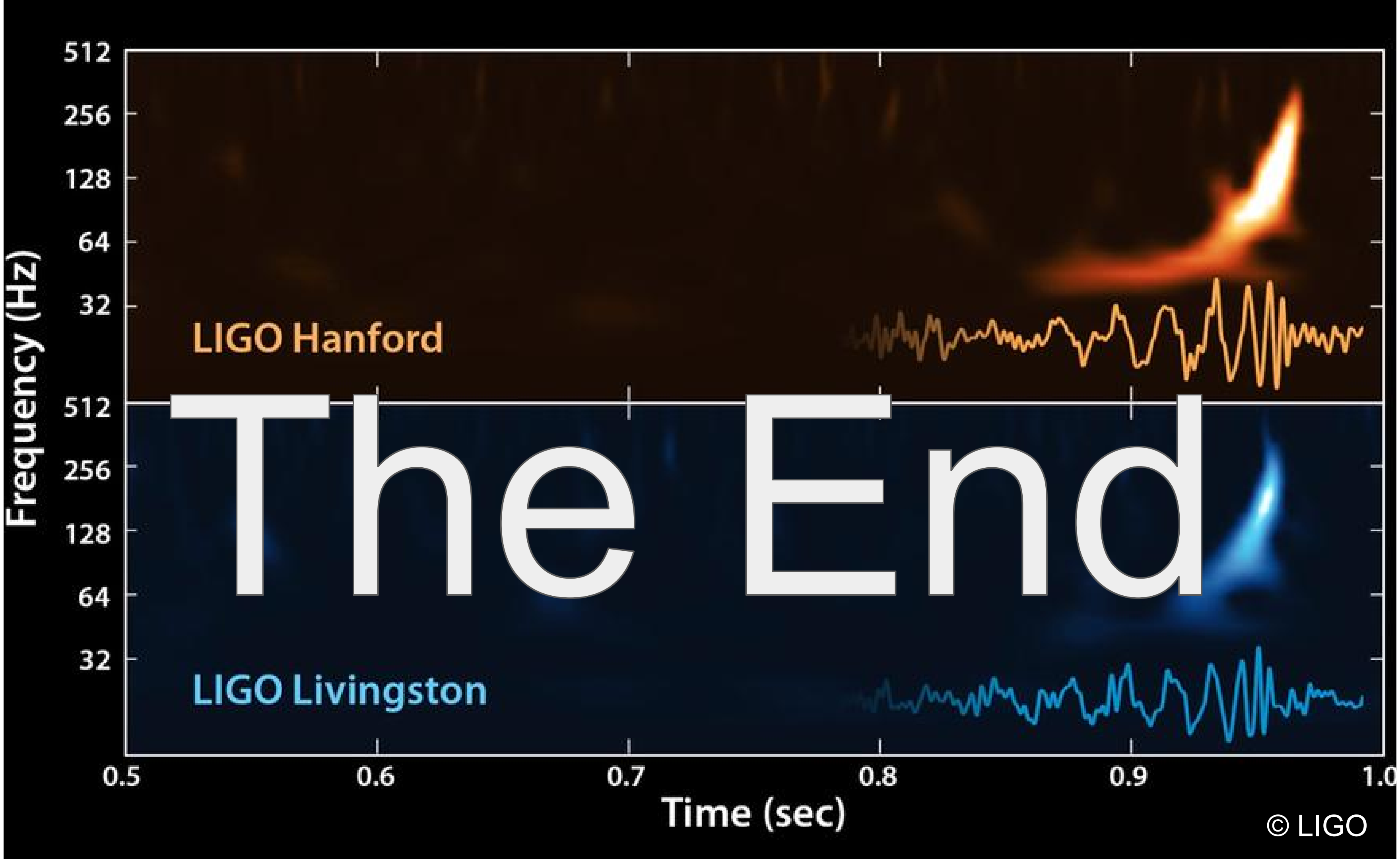
First time Blue Waters, OSG and Shifter are connected to address a science domain problem, and the first time this is done for gravitational wave science

NCSA

SDSC

Caltech

ABILENE CHRISTIAN UNIVERSITY 1906



512
256
128
64
32
LIGO Hanford
512
256
128
64
32
LIGO Livingston
Frequency (Hz)
0.5
0.6
0.7
0.8
0.9
1.0
Time (sec)

The End

© LIGO