



SDSC SAN DIEGO
SUPERCOMPUTER CENTER

Selkies

Qi QUALCOMM
INSTITUTE

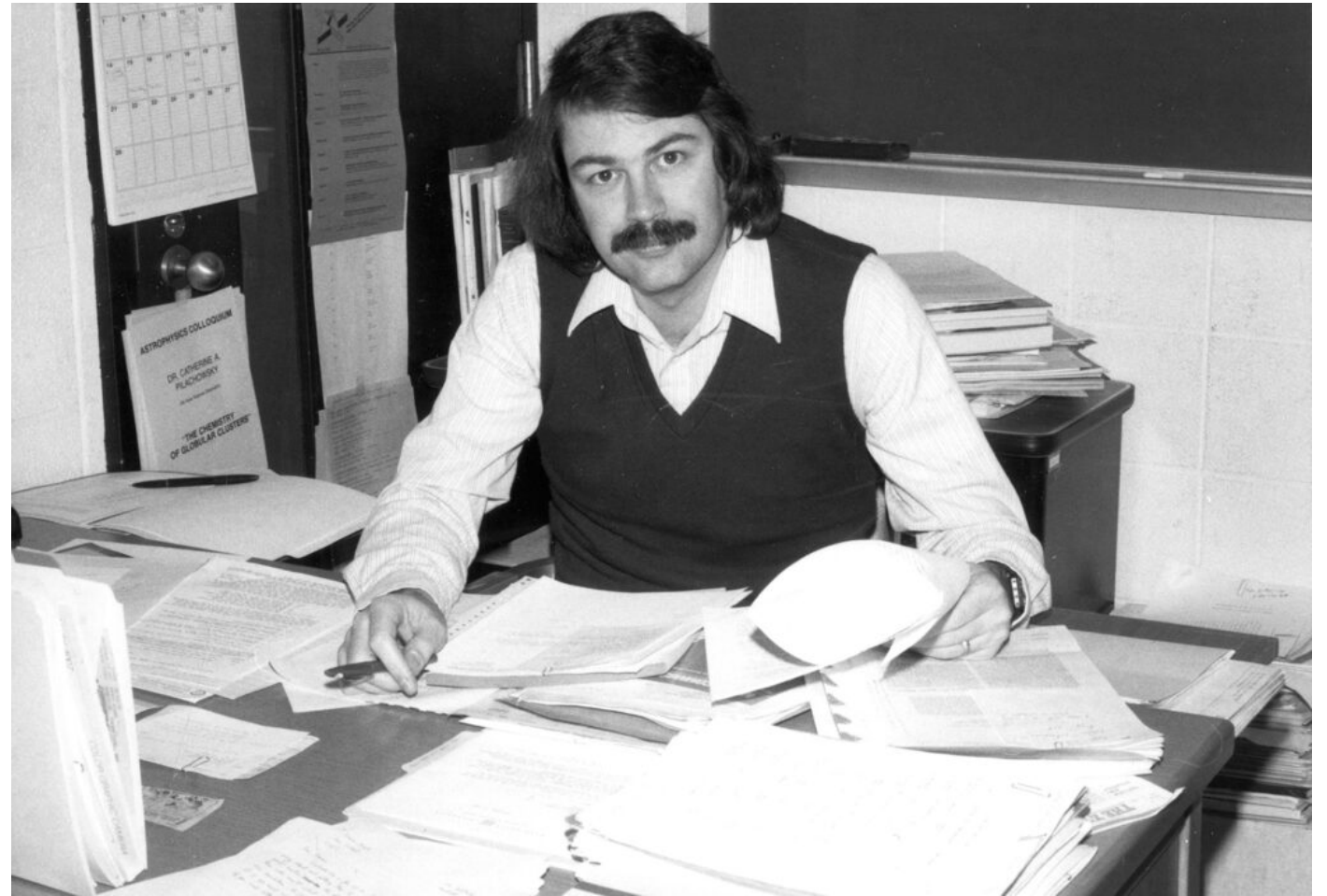
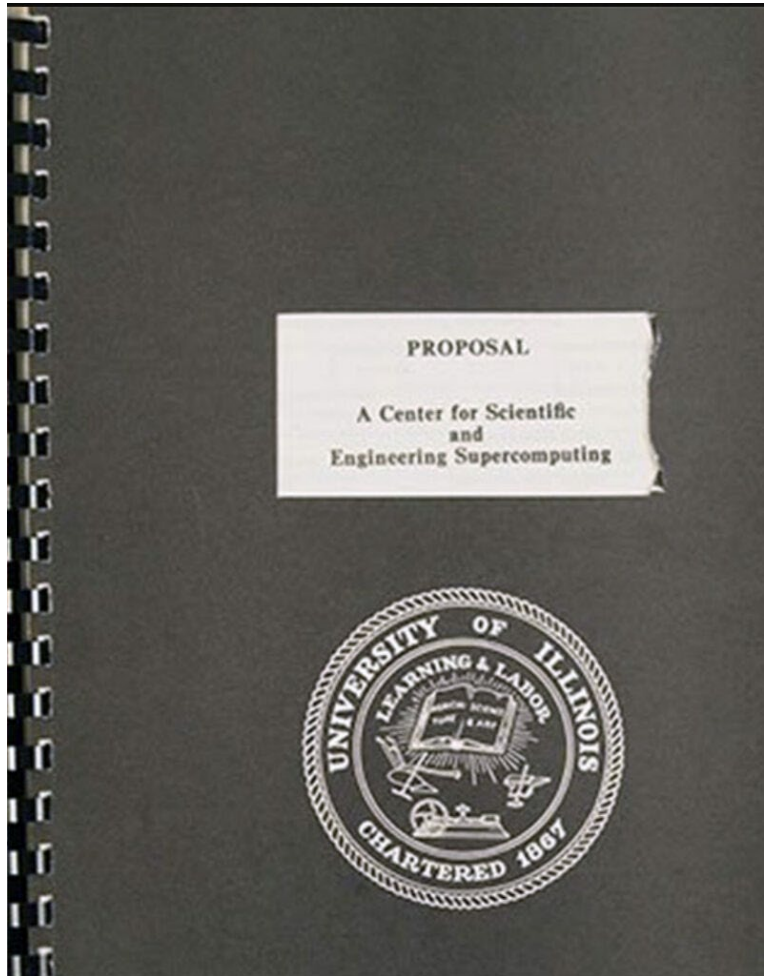
Bringing the governance of remote HPC back to the community

Seungmin Kim

College of Medicine, Yonsei University

San Diego Supercomputer Center, University of California, San Diego

The story starts...

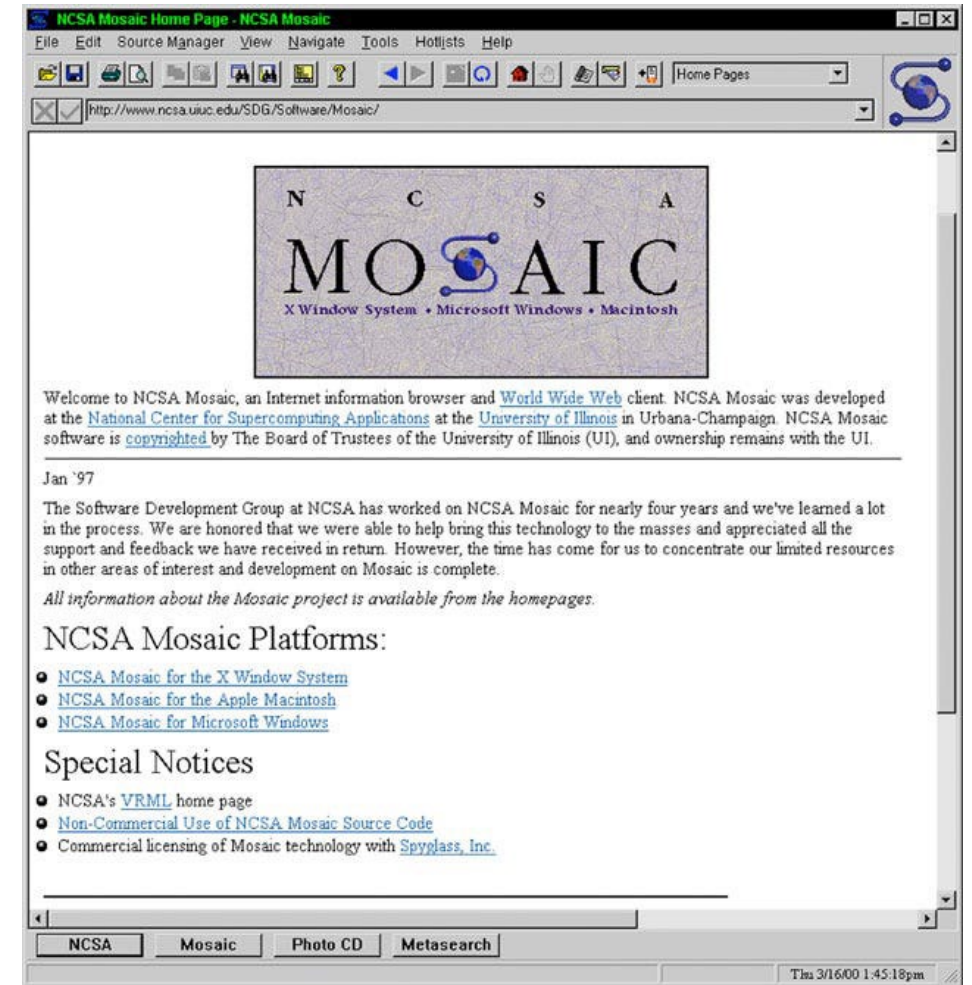


Dr. Larry Smarr, Ph.D., and the first five supercomputers

Now, when did everyone use computers at home?



Tandy 1000 SL, 1986-88

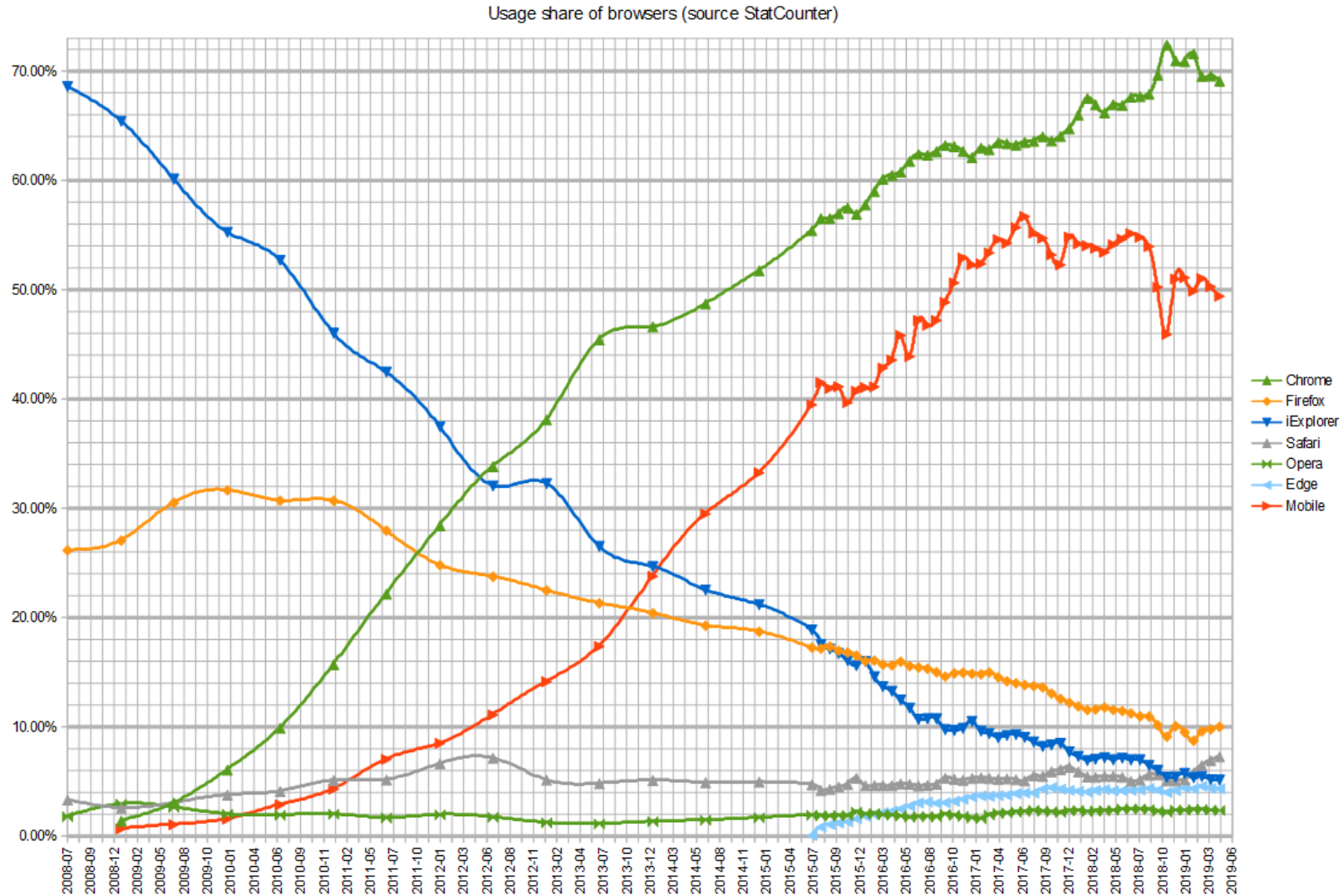


NCSA Mosaic, 1993

Netscape and Internet Explorer

- Mosaic was arguably the moment the computer became a commodity from an enthusiast machine
- The internet was a key traction in making computers an essential home appliance from a limited user pool
- The GUI was always the essential component for user accessibility
- But Netscape and Microsoft made the Internet proprietary by dumping non-standard extensions until Google Chrome overtook IE in market share
- The internet decomposed centralized clusters into individual hands

The Drift of the Proprietary Internet



Age of Clusters and Supercomputers... Again



1985-8, NSF Office of Advanced Cyberinfrastructure



2023, NSF Office of Advanced Cyberinfrastructure
National Artificial Intelligence Research Resource Pilot

AI Revives Linux & Clustered Computing

- In the 2010s, the cloud era arrived, decomposing physical servers into logical abstractions, with Linux as its champion
- Large Language Models and other deep learning paradigms reward massive scale; not a personal workstation, a whole cluster
- Containers (Docker/Kubernetes/Apptainer) became the AI workload abstraction platform of choice
- AI workloads, therefore, are now commonly distributed from a central source such as ACCESS for individuals to access remotely
- HPC and supercomputers, with Linux, are now again, seriously relevant to everyone's lives

AI Revives Clustered Computing



Frontier Supercomputer, Oak Ridge National Laboratory

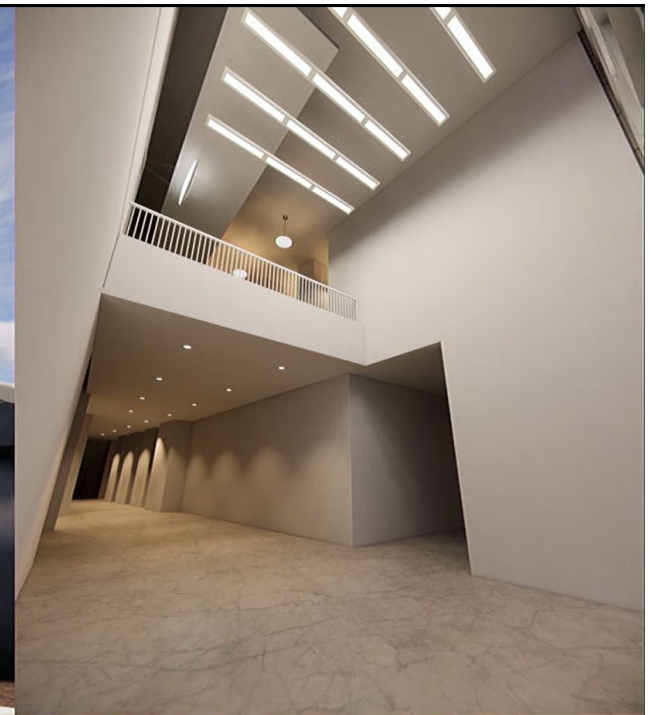


Eos NVIDIA DGX SuperPOD, NVIDIA Corporation

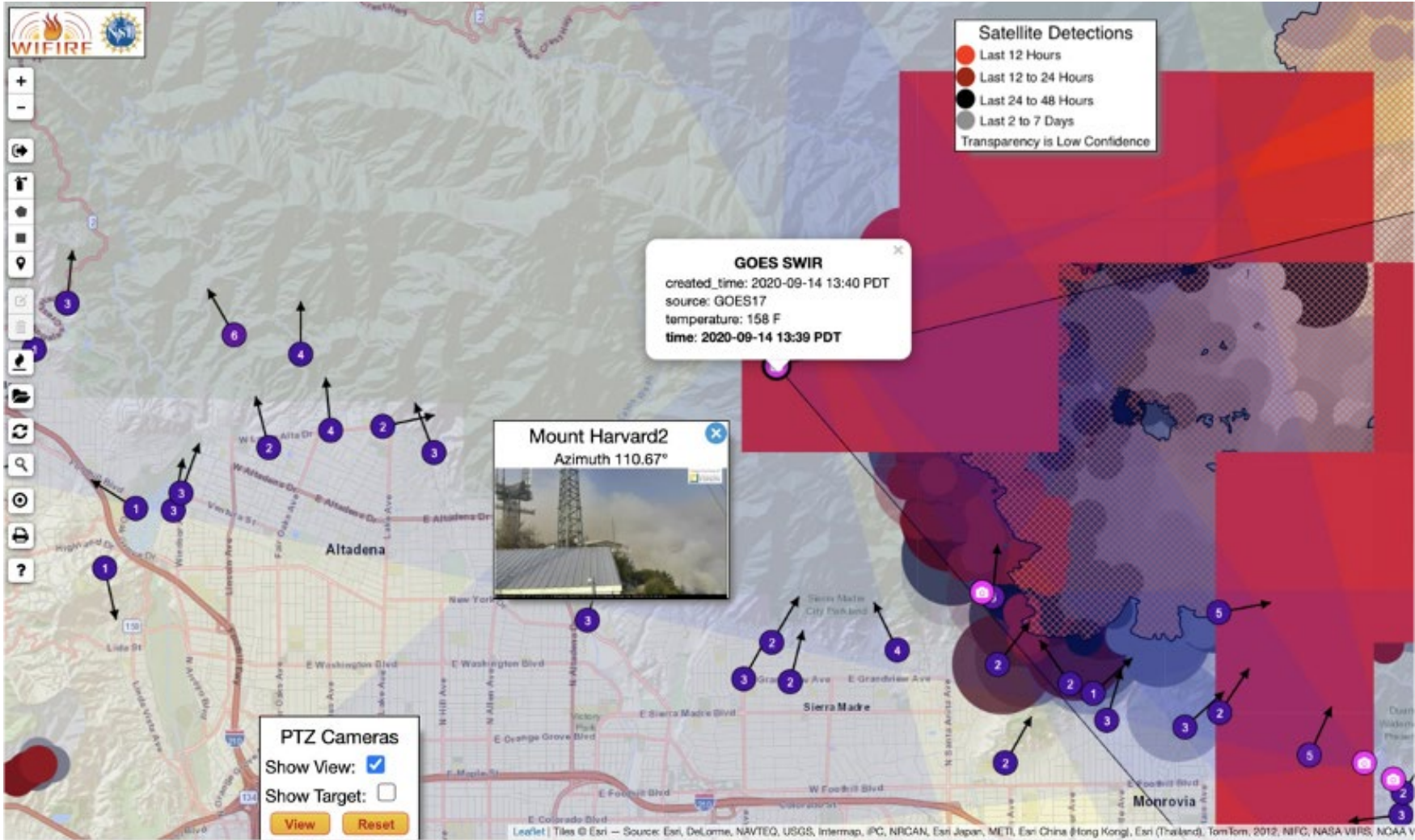
Massive Emerging 3D Demand for Linux

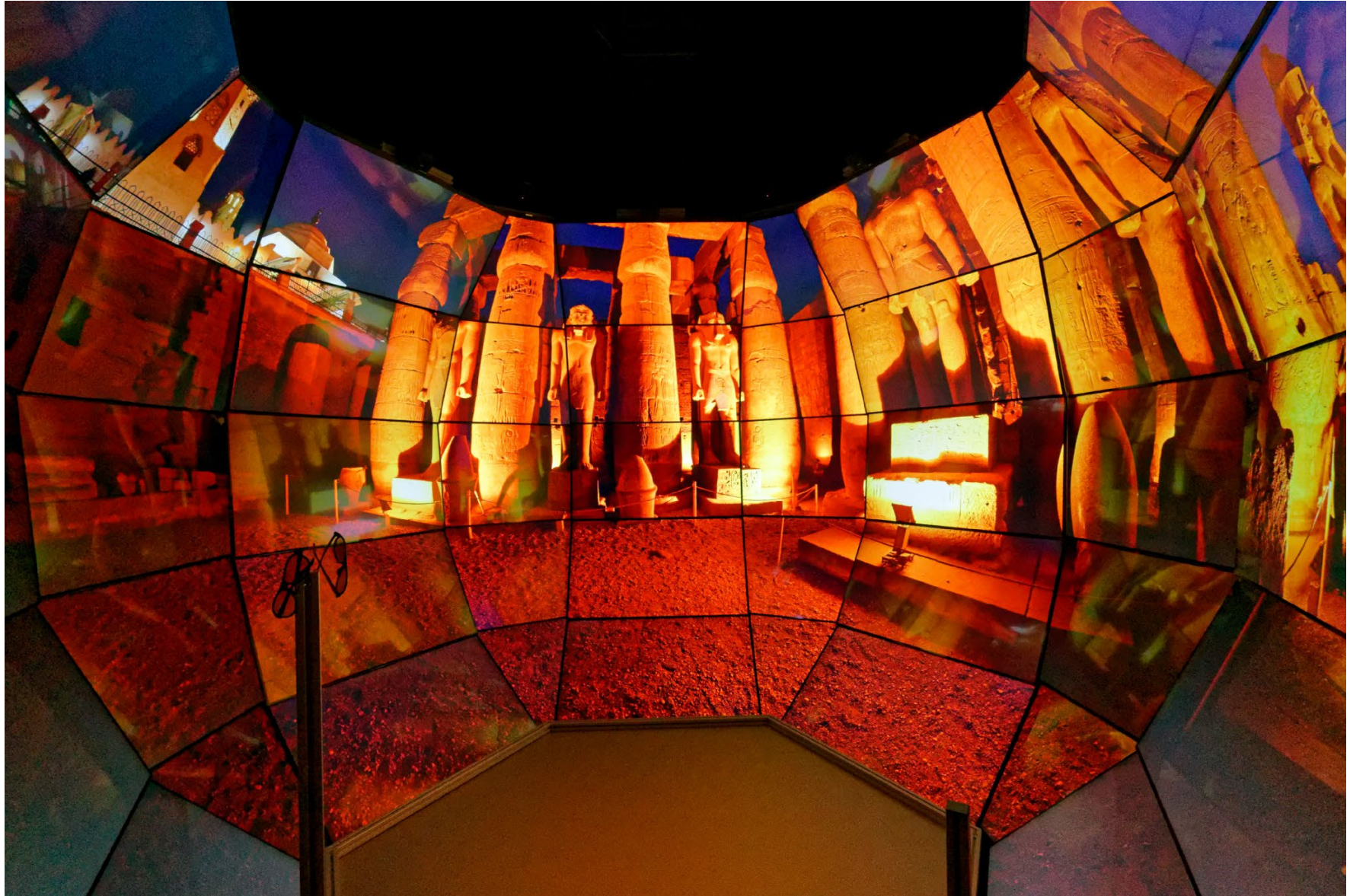
- Robotics Simulators with ROS/Gazebo
- Computer Vision with OpenCV, Autonomous Driving with CARLA
- Fluid Dynamics and CAD simulations with ANSYS, Solidworks
- FPGA Development Environments with Xilinx Vivado and Vitis
- 3D Modeling with Blender & Unreal Editor
- Drug Discovery & Structural Biology with VMD, PyMOL, ChRIS
- Geographic Information System (GIS) with QGIS/ArcGIS/OSGeo/Cesium
- Drone Mapping with OpenDroneMap
- The Collaboration for Astronomy Signal Processing and Electronics Research (CASPER) with GnuRadio
- Atmospheric Sciences, Archaeology, Geology, Medical Imaging, Oceanography, and other sciences
- Early wildfire detection with WIFIRE (Urgent HPC)
- Other interactive decisions and tasks requiring the power of HPC clusters

Digital Twins - Geolocated accurate 3-D representations of real-world structures in situ.



<https://www.youtube.com/watch?v=G4P6cc3vq7I>





Remote Desktop Landscape

- People wanted to access mainframes from their house.
- Started from command-line UIs (Telnet, SSH, etc.).
- X11 Forwarding / Microsoft Remote Desktop Protocol (send primitive CopyRect draw requests directly to client)
- Virtual Network Computing (VNC) (mix CopyRect/HexTile with Zlib and JPEG, nowadays primarily JPEG)
- libjpeg-turbo enables live-speed JPEG framebuffer encoding
- Xpra (similar to JPEG, evolved to utilize WEBP/H.264 and hardware encoding)

The Usefulness of H.264 Encoding in a VNC Environment

- 2014, D. R. Commander (<https://turbovnc.org/About/H264>)
- Author of VirtualGL, TurboVNC, libjpeg-turbo
- There are still only a handful of datasets for which H.264 can be shown to compress better than the TurboVNC encoder, ..., the improvement is now very compelling. The x264 library still takes up too much CPU time, however...
- This research did show that H.264 can be beneficial for certain types of apps...that generate somewhat game-like or video-like image sequences. X servers won't be around forever.
- Apps will increasingly stop using X primitives in favor of X RENDER and other image-based drawing methods...will become more H.264-friendly as time progresses.

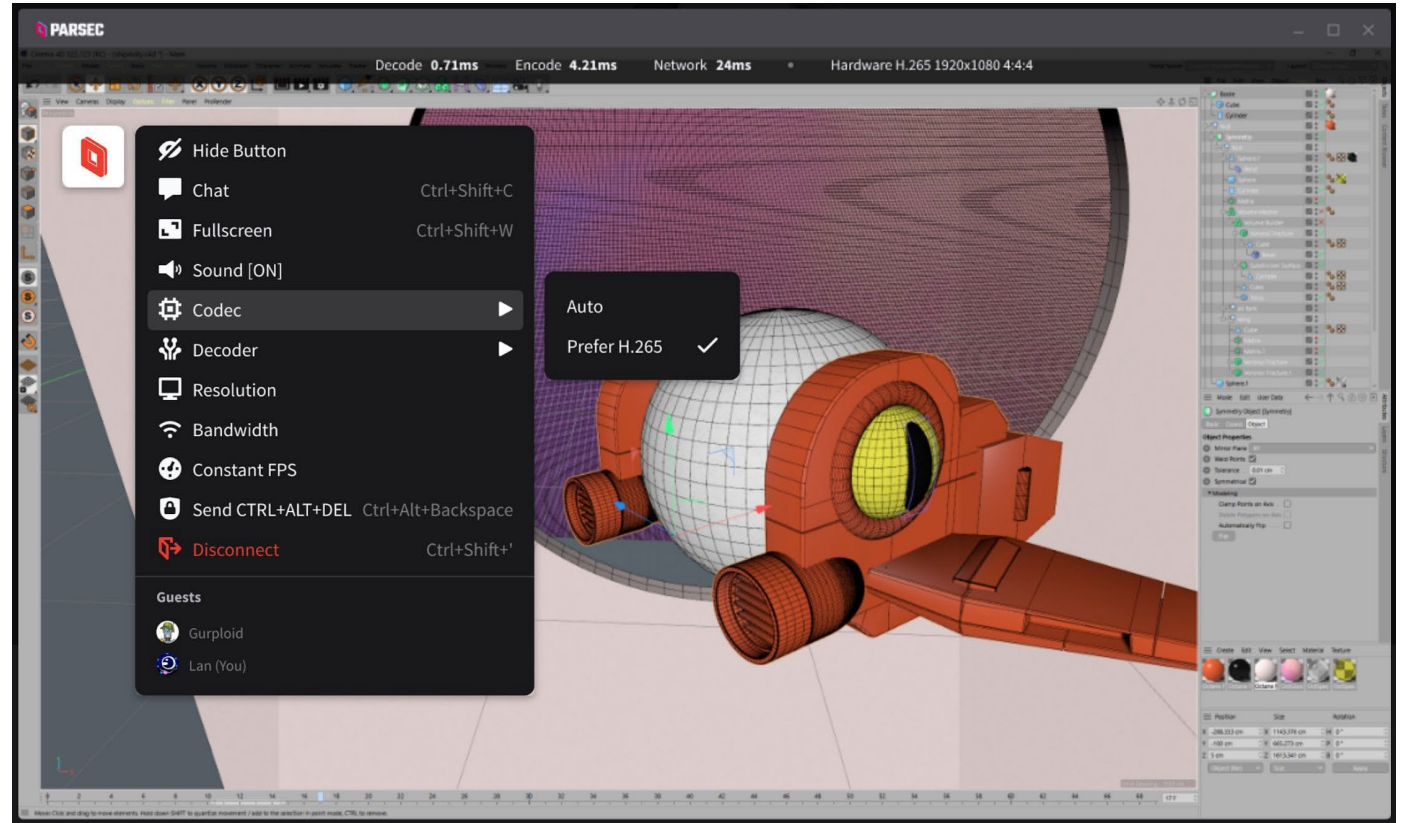
The Remote Game Streaming Scene

- OnLive (2009), GamingAnywhere (2013), Parsec (2016)
- H.264 video encoding in both hardware (GPU) and software (CPU) overtook the efficiency of JPEG and X11 CopyRect
- Microsoft RDP RemoteFX Media Streaming (H.264) in 2016
- Google Stadia implemented capabilities for low-latency interactive computing for Chrome, later in other browsers
- NVIDIA GeForce Now, Microsoft Xbox Cloud, Amazon Luna (2018-2020)
- Subsequent adaptation to professional graphics & visualization during COVID-19
- Rise of the scene just when I hated TigerVNC (back in 2017-18) for managing my high school GPU workstation for Molecular Dynamics

The Remote Game Streaming Scene

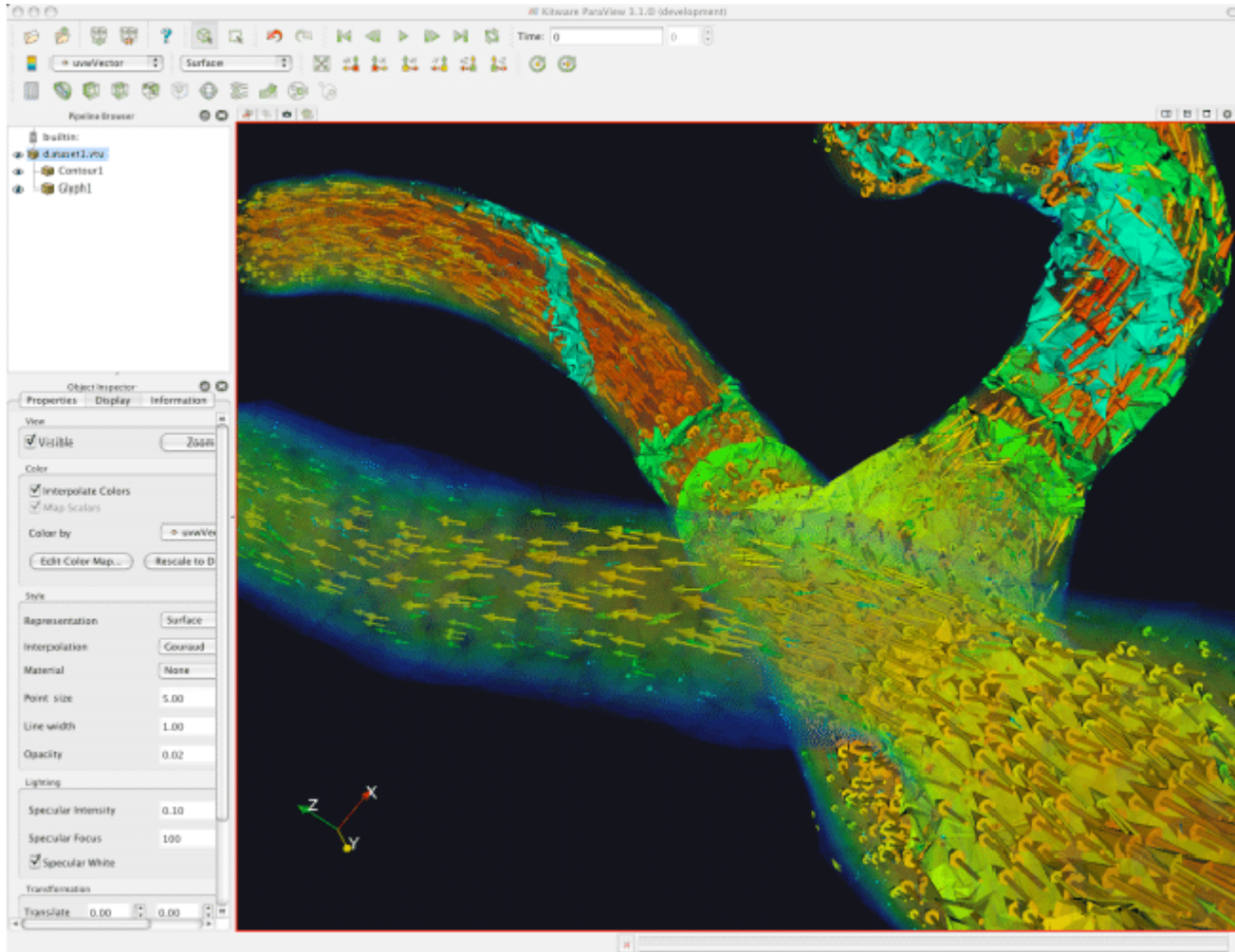


OnLive, 2009

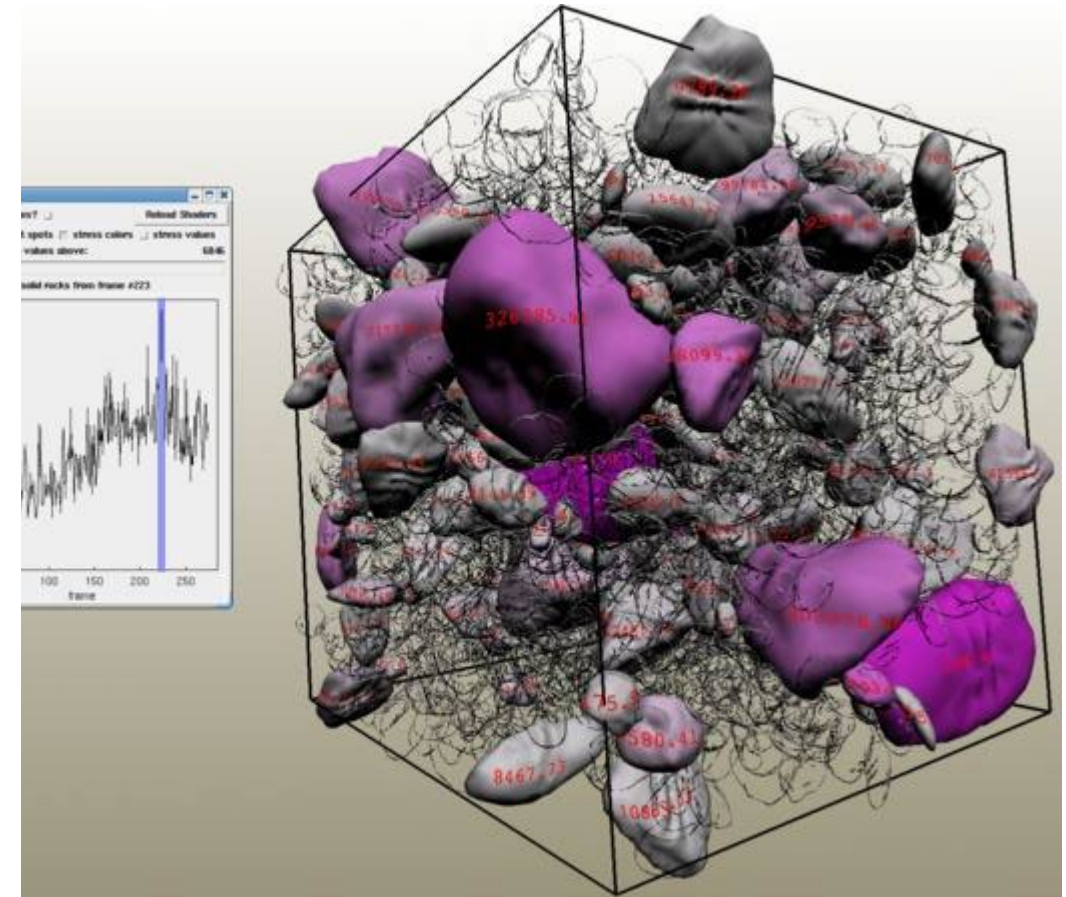


Parsec, 2016

Rise of Complex Graphics for Cloud/HPC



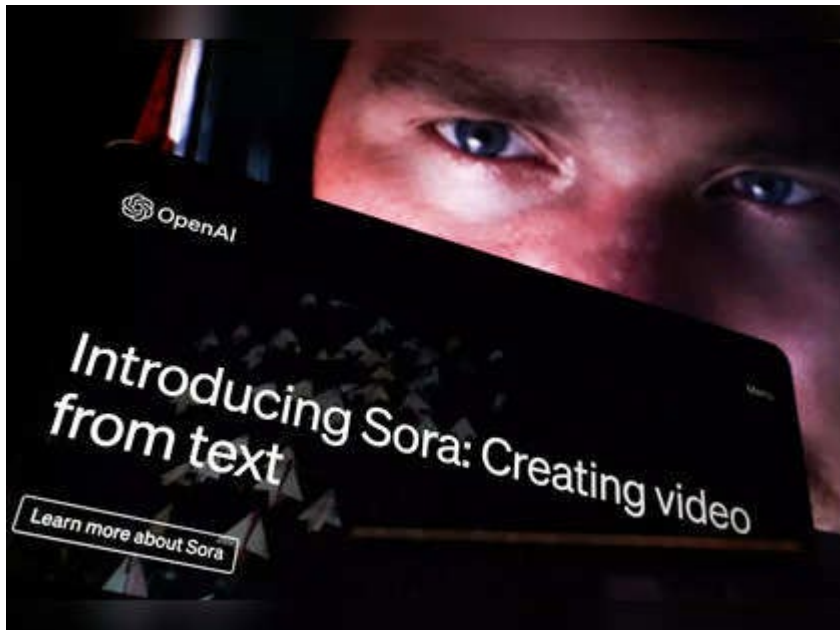
ParaView



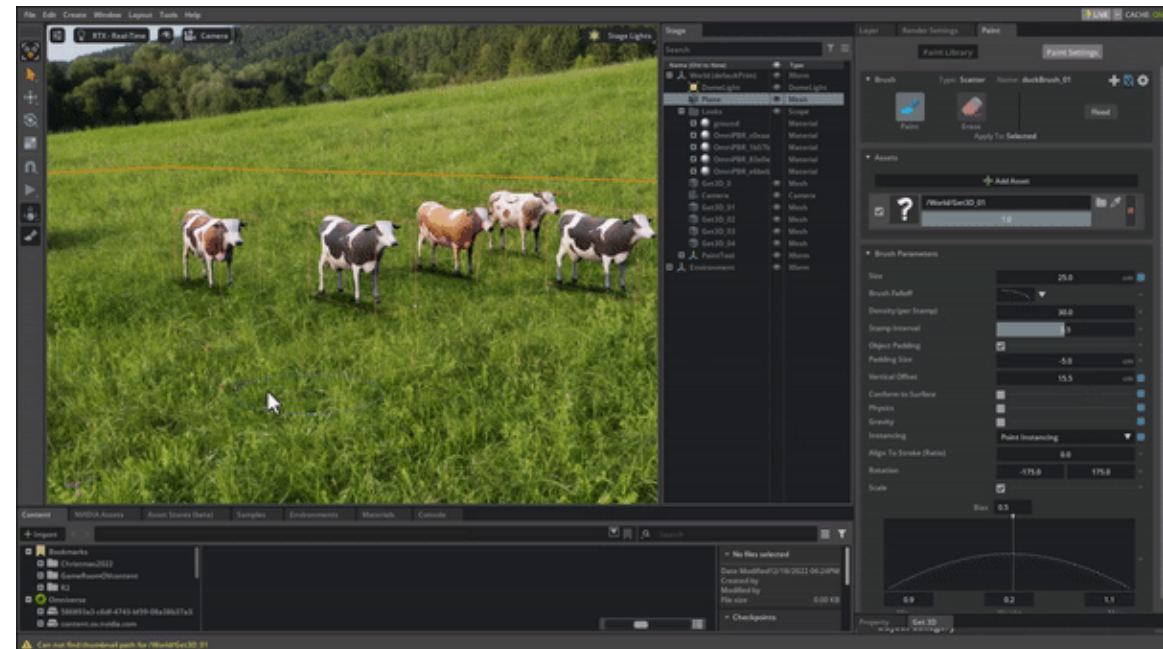
NIST

Rise of Multimodal & Spatio-Visual AI

- AI used to be just numbers; input numbers and predict the output (Regression, Classification)
- Then, text came along (ChatGPT)
- Now, AI outputs are interactive and spatio-visual, together with Linux



OpenAI Sora



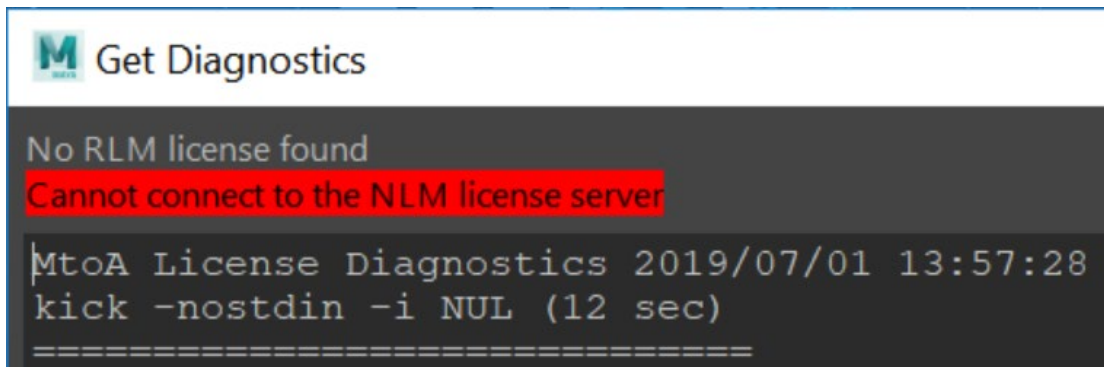
NVIDIA Omniverse

Current State of Linux Remote Desktops

- FOSS: RealVNC, rdesktop, TigerVNC, TurboVNC, Xpra, Xrdp, RustDesk...
- Either rely on VNC-like protocols or are meant for TeamViewer-like remote user support
- No FOSS solution in Linux maintaining server-client architecture to deploy on massive clusters AND providing modern performance, except for VNC – RFB is nearly 30 years old
- Paid: NoMachine, NICE DCV, Teradici, Reemo, Citrix...
- All have H.264 implemented for Linux 3D graphics
- Alternative: Moonlight + Sunshine – FOSS but designed for individual game streaming from desktops

Issue with Proprietary Remote Desktops

- License Servers and Seats! MATLAB is already painful enough
- The issue here is flexibility, not the fees
- Cannot bundle into a portable environment (Snapshot/Container)
- Cannot customize deployment to various environments; forces a certain environment configuration the program wants
- OS and architecture support is also inflexible



M Get Diagnostics

No RLM license found
Cannot connect to the NLM license server

```
MtoA License Diagnostics 2019/07/01 13:57:28  
kick -nostdin -i NUL (12 sec)  
=====
```

Licenses - Reemo Studio - Reemo Private Cloud

We offer 2 licensing models:

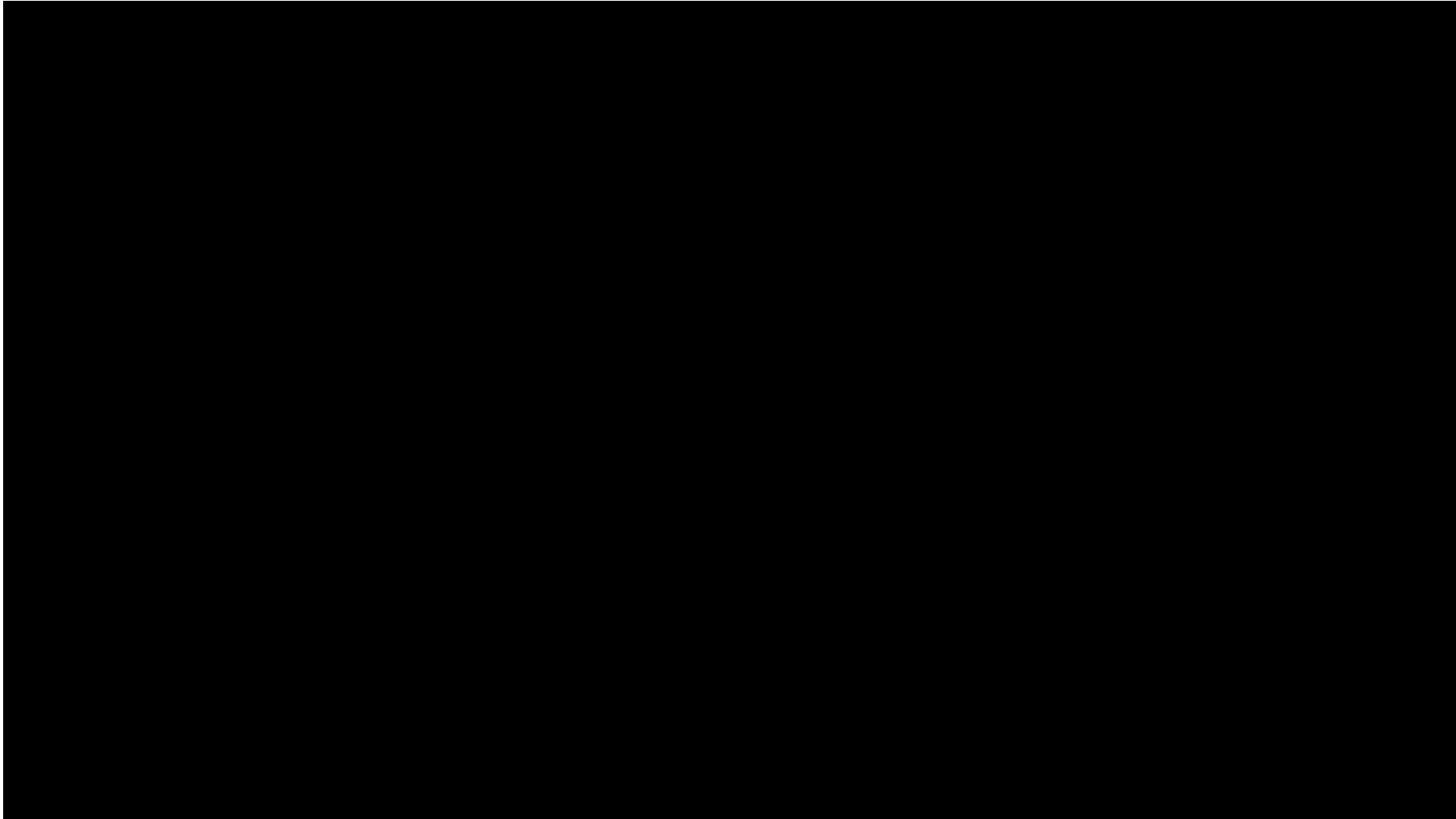
"Per User" if you don't know how many users will use the licenses at the same time.

"Per Session" if you want to control the maximum number of licenses used at the same time.

Billed Yearly

PRODUCT	MODE	PRICING	QUANTITY	TOTAL
Reemo Remote Desktop - Reemo Private Cloud Licenses used to access your computers remotely.	<input type="radio"/> Per User <input type="radio"/> Per Session	50-99 419.00 € / year 100+ 379.00 € / year	<input type="text" value="12"/>	5028.00 € / year
Reemo Browser Isolation - Reemo Private Cloud Licenses used to provision and access your secure browsers.	<input type="radio"/> Per User <input type="radio"/> Per Session	100-299 192.00 € / year 300-499 168.00 € / year 500+ 160.00 € / year	<input type="text" value="12"/>	2304.00 € / year
Total Excl. Tax:				7332.00 € / year
Tax:				0.00 € / year
Total Incl. Tax:				7332.00 € / year

Current State of Linux Remote Desktops



Design of Selkies

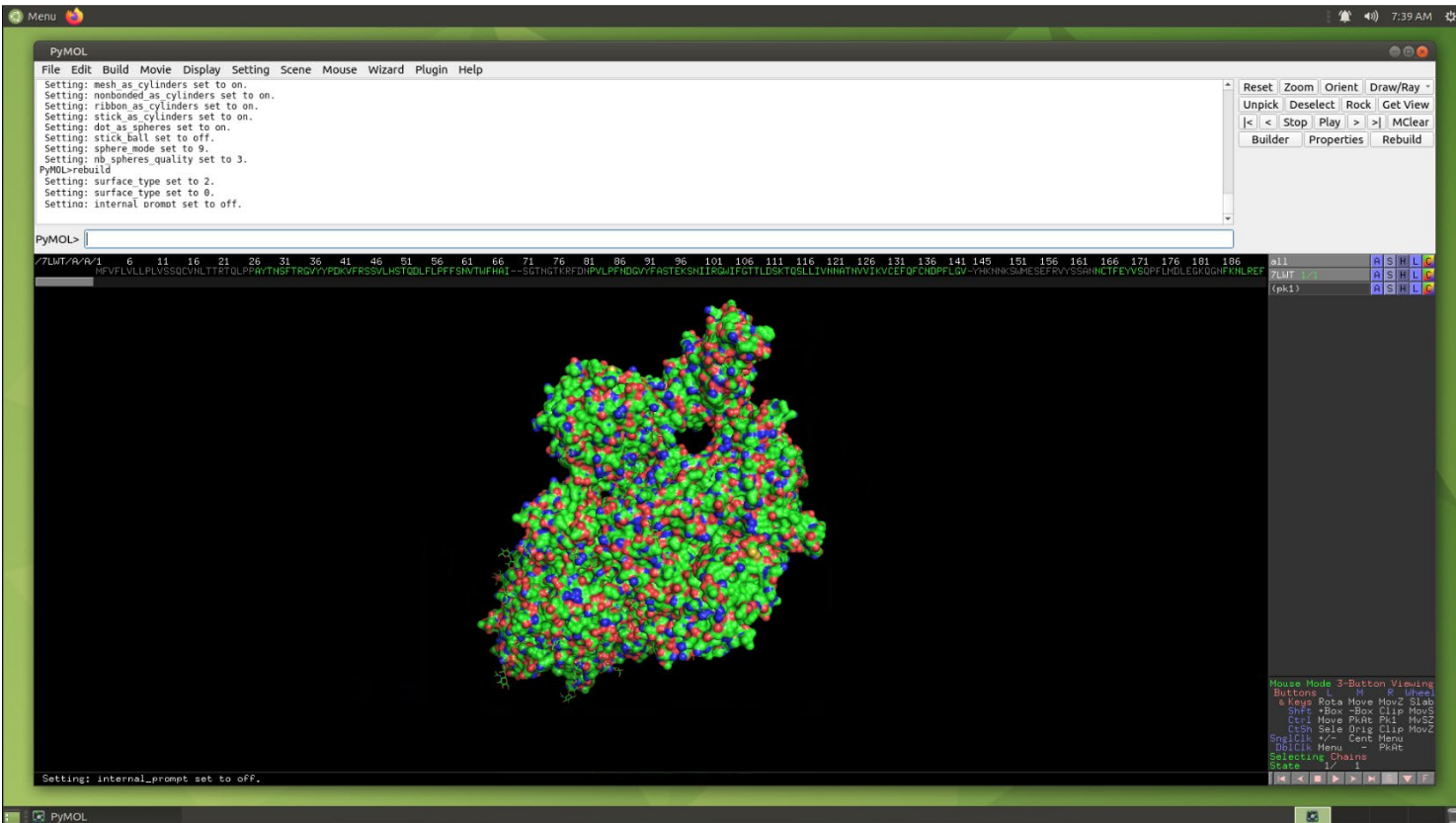
- Usage of HTML5 and WebRTC; anywhere Chromium, Firefox, Safari exists is a client (including embedded platforms)
- Utilizes the same techniques Google Stadia, NVIDIA GeForce Now, etc. used, obtaining equal or better latency and quality to commercial products
- Modularized: Web Client, Server, and Media Processor (GStreamer) are in separate components, compile everything anywhere
- Designed for Containers and HPC clusters, with minimum intervention from administrators (Kubernetes, (rootless) Docker/Podman, Apptainer)
- Open-Source License: MPLv2 allows flexibility to customize all code, but with a weak copyleft license
- Hardware GPU acceleration, both for the H.264 stream and the workload

Selkies Offerings

- docker-nvidia-egl-desktop (uses Xvfb + VirtualGL for HW acceleration of workloads), docker-nvidia-glx-desktop (X.Org server with NVIDIA X11 drivers inside the container)
- Portable distribution, compatible with any Linux OS glibc 2.17 or higher (based on the Conda build toolchain for distribution compatibility, but the user does not use Conda)
- Method to compile and/or install the Python Server, Web, and GStreamer Media Framework separately in any architecture
- Web interface is compatible with all major web servers and reverse proxies (HAProxy, NGINX, Jupyter, etc.), where noVNC is originally used

Selkies in modern Linux

- 10 years later from D. R. Commander, H.264, even in software x264 encoding, is capable of 60 FPS in Full HD resolution with only 130% CPU utilization
- NVIDIA and AMD/Intel GPUs are capable of live encoding QHD or 4K
- VNC (JPEG) struggles to encode 40-45 FPS with > 2000% CPU utilization with highly active or 3D visualization workloads
- Solves the underappreciation of GUI & Visualization in Linux, where multimodal AI is accelerating a new demand
- X11 is used primarily, Wayland has issues with NVIDIA drivers (needs to enable optional kernel modules)



3.04 60 14 10 0.45

Video bitrate

16 mbps

Dynamic bitrate selection for host video encoder

Video framerate

60 fps

Framerate selection for host video encoder

Audio bitrate

128 kb/s

Streaming bitrate selection for host audio encoder

- Peer connection state: **connected**
- Peer connection type: **relay**
- Packets received: **23576**
- Packets lost: **0**
- Bytes received: **23.70 MBytes**
- Bytes sent: **0.62 MBytes**

Video Stats

- Latency: **8 ms**
- Video: **H264 1920x1080**
- Video encoder: **hardware (nvh264enc)**
- Window size: **2880x1678**
- Framerate: **60 fps**
- Bitrate: **3.04 mbps**
- Available receive bandwidth: **5.37 mbps**

Audio Stats

- Latency: **74 ms**
- Codec: **OPUS**
- Bitrate: **128.50 kbps**

Shortcuts

- Fullscreen: Ctrl + Shift + F or Fullscreen Button
- Remote (Game) Cursor Lock: Ctrl + Shift + LeftClick
- Open Side Menu: Ctrl + Shift + M or Side Button
- Please cite within your publication for academic usage

Benchmarking



Video bitrate

16 mbps

Dynamic bitrate selection for host video encoder

Video framerate

60 fps

Framerate selection for host video encoder

Audio bitrate

128 kb/s

Streaming bitrate selection for host audio encoder

- Peer connection state: **connected**
- Peer connection type: **relay**
- Packets received: **724378**
- Packets lost: **44**
- Bytes received: **782.03 MBytes**
- Bytes sent: **11.88 MBytes**

Video Stats

- Latency: **8 ms**
- Video: **H264 1920x1080**
- Video encoder: **hardware (nvh264enc)**
- Window size: **2880x1678**
- Framerate: **60 fps**
- Bitrate: **15.33 mbps**
- Available receive bandwidth: **23.72 mbps**

Audio Stats

- Latency: **37 ms**
- Codec: **OPUS**
- Bitrate: **128.76 kbps**

Shortcuts

- Fullscreen: Ctrl + Shift + F or Fullscreen Button
- Remote (Game) Cursor Lock: Ctrl + Shift + LeftClick
- Open Side Menu: Ctrl + Shift + M or Side Button
- Please cite within your publication for academic usage

Bringing Remote HPC Back to Everyone

- Freedom to bundle and customize to every environment
- Highly readable code for future contributors
- Sustained by consultancy contracts (for industrial contributors) and academic involvement
- Linux gamers are also our source of feedback in terms of usability and performance; Selkies is also the containerized, web-based choice
- Citations are the currency of academia, this is why primarily proprietary technologies can become open-sourced
- Future Potential: NSF Pathways to Enable Open-Source Ecosystems
- The way forward is to bring Selkies to a de-facto standard for Interactive and Urgent HPC across all of the states

The spirit of NCSA

- NCSA was the birthplace of the precursors of Netscape Navigator, Internet Explorer, Apache HTTPd, and many other tools we use
- Despite using modern technologies such as Docker, Kubernetes, Apptainer, Conda, etc., the core of our technologies and the hardest component is still tussling with X11, old-school to this day
- This leaves the door open for many senior-level HPC personnel with decades of experience to pitch their expertise
- Selkies must become a community-sustained project to survive; input from many people are always better than one

Thank You

This project has been developed and is supported in part by the National Research Platform (NRP) and the Cognitive Hardware and Software Ecosystem Community Infrastructure (CHASE-CI) at the University of California, San Diego, by funding from the National Science Foundation (NSF), with awards #1730158, #1540112, #1541349, #1826967, #2138811, #2112167, #2100237, and #2120019, as well as additional funding from community partners, infrastructure utilization from the Open Science Grid Consortium, supported by the National Science Foundation (NSF) awards #1836650 and #2030508, and infrastructure utilization from the Chameleon testbed, supported by the National Science Foundation (NSF) awards #1419152, #1743354, and #2027170. This project has also been funded by the Seok-San Yonsei Medical Scientist Training Program (MSTP) Song Yong-Sang Scholarship, College of Medicine, Yonsei University, the MD-PhD/Medical Scientist Training Program (MSTP) through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea, and the Student Research Bursary of Song-dang Institute for Cancer Research, College of Medicine, Yonsei University.

The TURN REST API Method

