# Designing Interactive Audio Experiences Using HIVE: A Distributed Cloud Supercomputer for Real-Time Multimedia Interactions


Yubei Tang[1], Jesse Allison[2,3], Frederick Ostrenko[2,4], Hardika Patel[5]

1. Rutgers University, New Brunswick 08901
2. Center for Computation and Technology, Louisiana State University, Baton Rouge, 70803
3. School of Music, Louisiana State University, Baton Rouge 70803
4. College of Art and Design, Louisiana State University, Baton Rouge 70803
5. Department of Computer Science, Baton Rouge Community College, Baton Rouge, 70806



## Abstract

Cloud-based high-performance computing presents a model for user-interactive multimedia performances on a more ambitious scale than ever. The high-performance computer HIVE is a distributed computing system: resources and tasks are allocated between virtual machines that pass messages to each other in a network. These machines are stored on the cloud and can be configured remotely. HIVE's capacity to call up the power of as many virtual computers (VMs) as needed lends itself to constructing multimedia installations that let a scalable number of participants interact with audiovisual data in real time, as emphasized by its name, High-performance Interactive Visualization and Electroacoustics. In a demonstration, an application in Max/MSP receives audio data from a web browser. A website allows users to interact with the performed audio in real-time.


## Inside HIVE

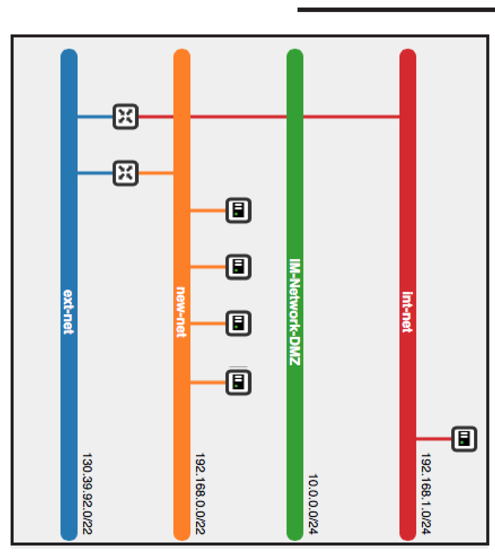
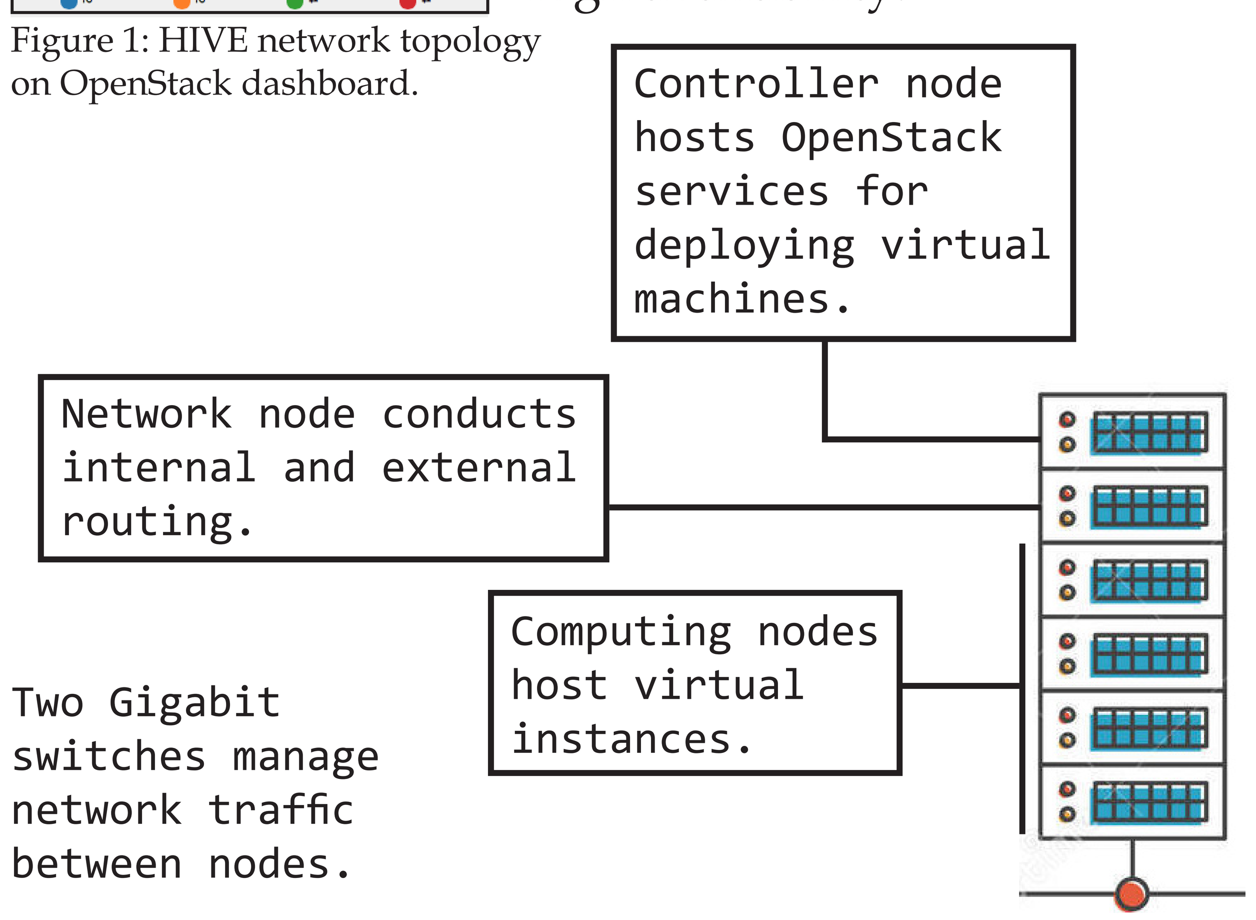
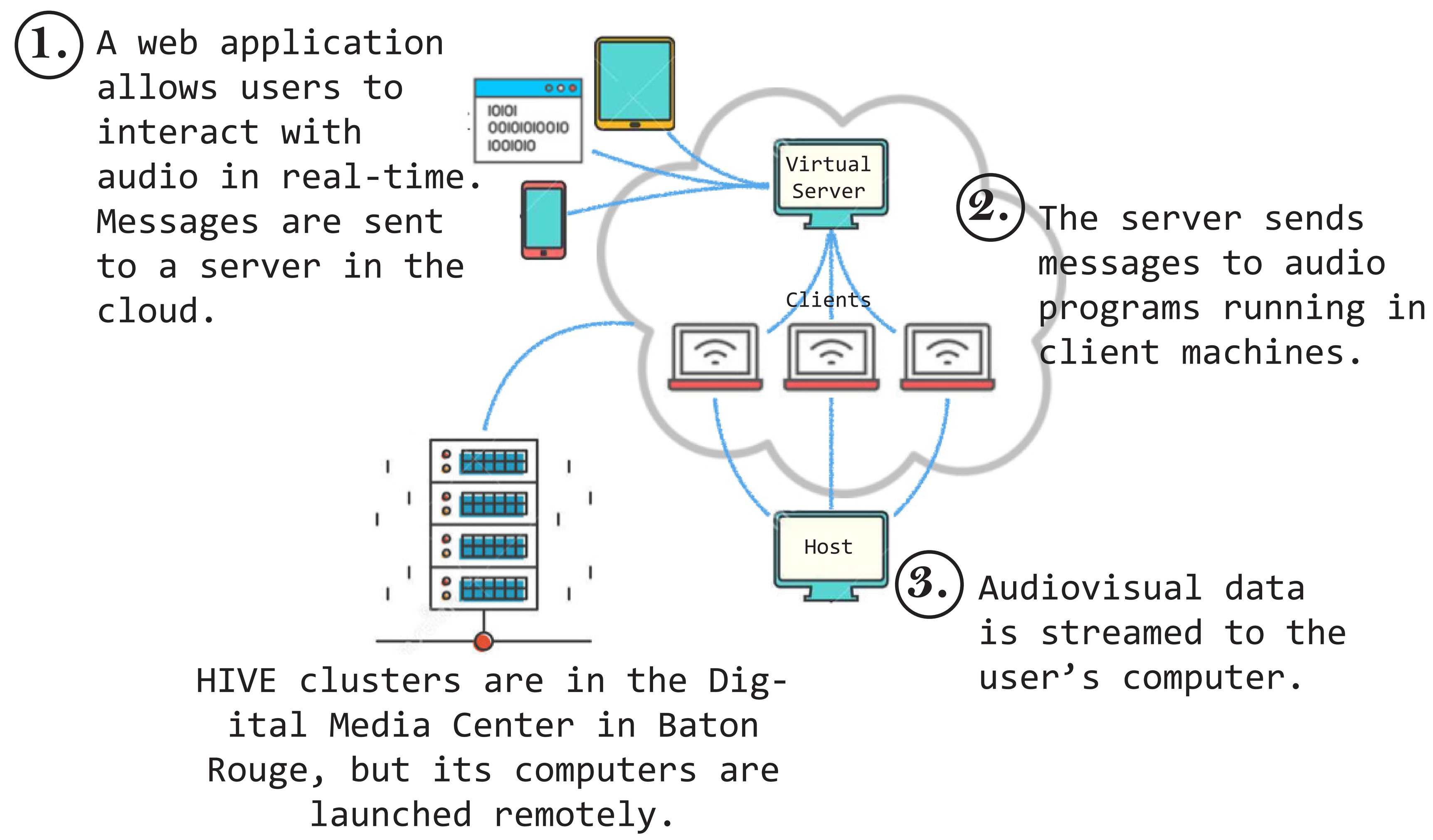


Figure 1: HIVE network topology on OpenStack dashboard.

HIVE is a 448-core cloud supercomputer built on OpenStack. It comprises one HP controller node, two 48-port Gigabit switches, and 30 Dell and 7 HP computing nodes. This system enables horizontal scaling and high availability.




## Your Head in the Cloud




Processing audio data in real-time for an extensible set of users requires enormous computing power, but HIVE's flexibility in launching virtual machines means it accomodates scalable number of participants.

## Latency Results

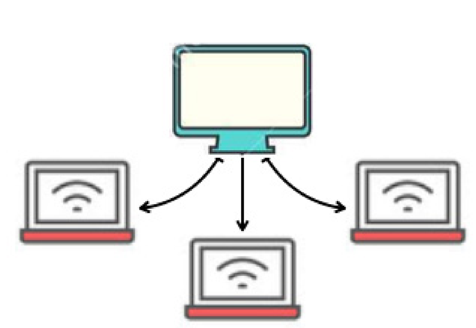


Figure 2: Parallel Set-up

In parallel set-up, a server sends a message to clients all at once. When all clients respond, it sends the next message.

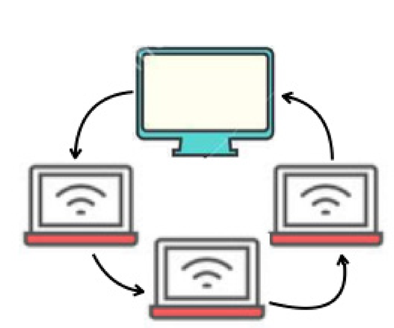


Figure 3: Round-Robin

A server sends a message in round-robin which is passed down a line of clients. When the last client responds, the server sends the next message.

**Table 1: 1000 Iterations of UDP Messaging with Max/MSP in HIVE VMs (seconds)**

| No. of VMs | Parallel | Round Robin |
|---|---|---|
| 2 | 38.0 | 11.2 |
| 3 | 34.1 | 18.8 |
| 4 | 36.3 | 28.6 |

Table 1 shows a start-up delay in messaging between two machines, but that adding machines to the network do not multiply latencies.

## Comparison with Physical Machines

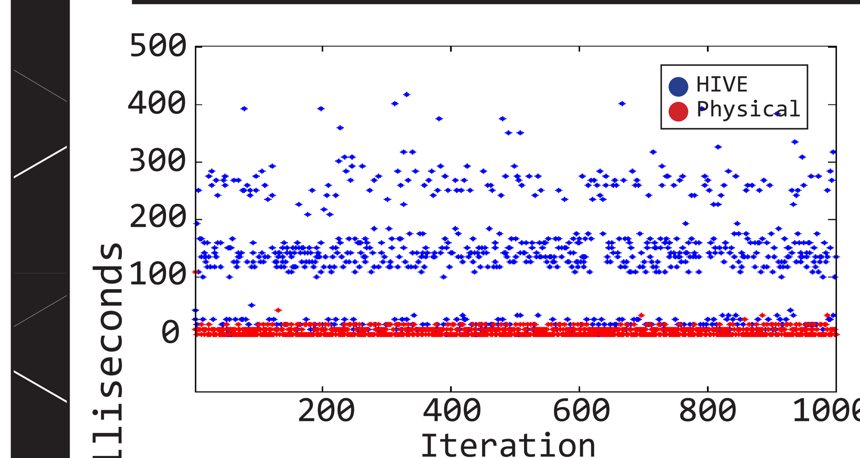



Figure 4: UDP over public Wifi in Max/MSP

The physical set-up uses an iMac server and three Macbook clients. The virtual set-up is a server and three clients in HIVE. The parallel sequence in Figure 2 is tested.

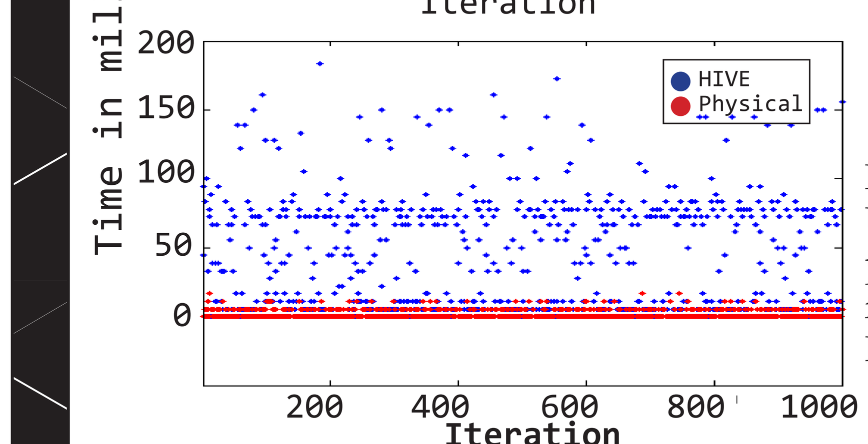
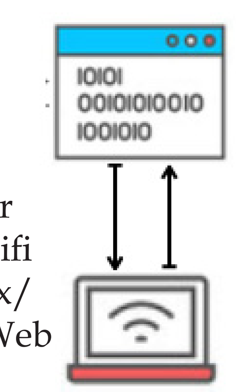



Figure 5: UDP over public Wifi from Max/MSP to Web



In the physical set-up, an iMac server running Max/MSP sends a message to a web browser on a second iMac. In virtual set-up, a HIVE instance messages a web browser on an iMac.

## Discussion

Tests in HIVE show that adding virtual machines to networks do not create additional latencies. Rather, the fluctuating timings and success rates of UDP messaging between two HIVE machines contribute most heavily to overall latencies. In order to locate sources of latencies, we intend to measure performances of VMs hosted on Dell nodes against HP nodes. Additionally, we wish to minimize the paths between the network node and each computing node. Traditional models recommend pooling resources on fewer VMs instead of distributing tasks too thinly for the reason that "latency between processes is less than latency between computers"[1]. However, our tests show that HIVE indeed has the horizontal scaling capacity for mediating web interactions that support thousands of users.

## References


1. Hindle, Abram. (2015, June 2). *Orchestrating Your Cloud-orchestra.* Paper presented at New Interfaces for Music and Electroacoustics, Baton Rouge.
2. Lopez-Lezcano, Fernando. (2012, April.) *From Jack to UPD Packets to Sound, and Back.* Paper presented at Linux Audio Conference, Palo Alto.


## Acknowledgments


This material is based upon work supported by the National Science Foundation under award OCI-1263236 with additional support from the Center for Computation & Technology at Louisiana State University. Huge thanks to Bety Rodriguez-Milla, Brandon Wheeler, Danny Holmes, Nick Hwang, and Zach Berkowitz for their immeasurable help and expertise.


RUTGERS THE STATE UNIVERSITY OF NEW JERSEY

LSU

batonrouge COMMUNITY COLLEGE

NSF