IPv6-Only HTCondor Integration with the Open Science Pool: The Viper Cluster at GW/CAAREN

Research Technology Services, GWIT
The George Washington University

June 19, 2025

1 Introduction

In 2025, The George Washington University [1], in partnership with the Capital Area Advanced Research and Education Network (CAAREN) [2], successfully contributed a single-stack IPv6 HTCondor [3] cluster, Viper, to the Open Science Pool (OSPool) [4], a nationally distributed computing platform for data-intensive research. This demonstrates a path forward for institutions preparing for an IPv6-only future, reinforcing GW's leadership in research cyberinfrastructure and open science.

There is some legitimate concern that integrating IPv6-only computational resources into the OSPool may have adverse effects on effective job-throughput. These concerns break down into two broad categories,

1. Interoperability with OSPool and HTCondor
2. Interoperability with researcher jobs which are far less predictable

2 System Overview

Viper is a 12-node HTCondor cluster managed with Warewulf 4.6.1 [5] running Rocky Linux 9.5 and supported by a dedicated network-attached ZFS pool for backups and administrative workflows. Each node in the cluster is a Dell PowerEdge R730 with the following hardware specifications:

Component	**Specification**
Cores	2x14 Intel Xeon E5-2680 v4 @ 2.40GHz
RAM	8x16 GB DIMM ECC 2400 MT/s
Scratch Storage	1 TB SSD

There are two master nodes serving as redundant central-managers in an active-passive relation, one submit node, and nine execute points. Because Viper uses Warewulf's stateless provisioning, nodes can be quickly repurposed or recovered after failure.

Viper is configured to advertise 4.5 GB of RAM per core to the Open Science Pool, allowing it to efficiently integrate with workloads of varying shapes. All Viper hosts have globally routable IPv6 addresses through CAAREN.

3 About CAAREN

The Capital Area Advanced Research and Education Network (CAAREN) provides high-performance networking to support education, research, and innovation in the Washington, D.C. area. As a regional optical network and member of Internet2, CAAREN connects GW to other research institutions and to global science collaborations.

CAAREN has been a leader in routing security, being among the first US higher education institutions to implement Route Origin Authorization (ROAs). CAAREN implemented the first know production deployment of TCP-AO for advanced BGP session security [6].

Also available on CAAREN is an advanced content distribution test environment using Tree Distribution Networking (TreeDN) [7]. Employing a combination of native multicast and overlay tunneling technology (via Automatic Multicast Tunneling - AMT), multicast content can be delivered to any endpoint.

CAAREN's core network includes:

- Juniper MX480
- DWDM with optical protection to connect to Intenet2
- Supplemental 10 Gbps access via secondary switches

Additional CAAREN services include:

– Cloud Connect: Dedicated links via Internet2's footprint to AWS, Azure, Google Cloud, and OCI, offering low-latency, secure access to cloud resources.

– The Things Network: A collaborative IoT sensor network initiative using LoRaWAN for low-power, long-range device communication.

These services position CAAREN as a leader for data-intensive science, advanced networking research, and the transition to secure, scalable IPv6-only architectures.

4 Network Architecture

– **CAAREN Connectivity:** Global IPv6 /64 allocations routed through CAAREN. All Condor daemon communication (Collector, Negotiator, StartD, Schedd) operates exclusively over native IPv6.

– **Firewalls:** Host-based firewall rules allowing limited IPv6 originating from both GW and OSG and unrestricted outbound. No other firewalls or access control lists.

5 HTCondor Configuration and IPv6 Deployment

All HTCondor components are configured to operate over IPv6 only. The following key values were set on each node via overlay.

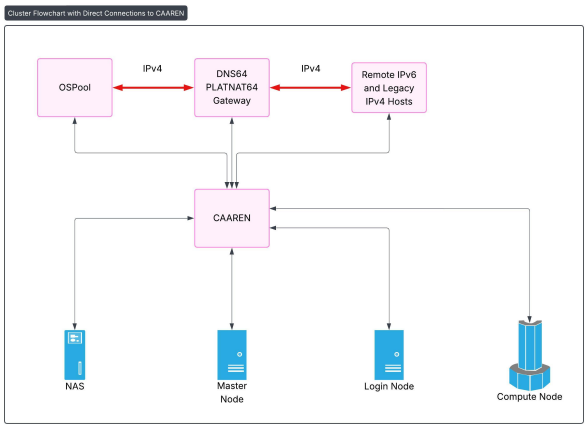

Figure 1: Essential network architecture of the Viper IPv6-only implementation.

Access Point and Central-Manager IPs

Note that the addresses below are reserved for documentation and used for illustrative purposes; they will need to be updated for site-specific networks.

Role	IPv6 Address
Central-Manager (CONDOR_HOST)	2001:db8:0:1::200
Submit/Access Point	2001:db8:0:1::100

/etc/condor/config.d/10_ipv6.conf

```
# submit/access point
UID_DOMAIN = 2001:db8:0:1::100

# central-manager
CONDOR_HOST = 2001:db8:0:1::200

NETWORK_INTERFACE = eno1
ENABLE_IPV6 = true
PREFER_IPV6 = true
ENABLE_IPV4 = false

ALLOW_READ = 2001:db8:0:1::/64
ALLOW_WRITE = 2001:db8:0:1::/64
ALLOW_NEGOTIATOR = 2001:db8:0:1::/64
ALLOW_ADVERTISE_STARTD = 2001:db8:0:1::/64
```

/etc/condor/config.d/20_role.conf

```
# central-manager
COLLECTOR_HOST = [2001:db8:0:1::200]:9618
```

6 DNS64, NAT64, and the Role of CLAT

Viper operates in an IPv6-only environment so some measures are required to enable connectivity to legacy IPv4 hosts and applications. This is achieved through a combination of DNS64, NAT64, and CLAT, an implementation of 464XLAT [8].

DNS64 and NAT64 are required for accessing upstream OSG services, Git repositories, and software mirrors that remain IPv4-only.

DNS64 and NAT64

- **DNS64:** When an IPv6-only host makes a DNS request for an A record (IPv4), a DNS64-enabled resolver synthesizes an AAAA record using a configured NAT64 prefix, often `64:ff9b::/96`, but in the case of Viper an external NAT64 gateway was used.
- **NAT64:** The NAT64 gateway intercepts IPv6 packets destined for these synthesized addresses and translates them to the corresponding IPv4 address, enabling IPv4 interoperability.

CLAT for Application Compatibility

Some applications do not support IPv6, including those that bind to literal, or hardcoded IPv4 addresses. To address this limitation, Viper utilizes the following:

- **CLAT (Customer-side Translator):** An open source implementation [9] of 464XLAT [8], `CLATD` allows IPv4-only applications to operate in Viper's IPv6-only environment. `CLATD` orchestrates the translation of IPv4 socket calls into IPv6, which are then handled by Viper's provider-side translator (PLAT)—the NAT64 gateway responsible for bridging to legacy IPv4 services.
- **Implementation:** On Viper, `CLATD` runs on each node and provides its own instance of `TAYGA`, an out-of-kernel stateless NAT64 daemon. `CLATD` (and `TAYGA`) are integrated at boot through systemd units within overlays.

Note, in a dual-stack environment (e.g., while transitioning from legacy to IPv6) it is important that HTCondor not attempt to bind to the virtual `clat` interface for intra-daemon communication. It is typically sufficient to specify the appropriate interface by setting `NETWORK_INTERFACE` and redirecting all other IPv4-bound traffic to the `clat` interface.

Together, DNS64, NAT64, and CLAT allow IPv6-only nodes to maintain operational capability with legacy IP hosts and applications.

7 Results

- Viper has been integrated successfully into the OSPool.
- Despite signaling ENABLE_IPV4 = false and PREF

8 Conclusions

Appendix: Site-Specific Notes

8.1 Installing CLATD

Due to the stateless nature of Viper's execute images, installing CLATD into an image presents an issue. The CLATD installation procedure doesn't handle TAYGA installation correctly if it cannot start or run daemons, as is the case with modifying an image. For installation on Viper, the CLATD Makefile was modified and TAYGA installed separately. Viper's minimal CLATD Makefile is,

```
DESTDIR=
PREFIX=/usr
SYSCONFDIR=/etc

DNF_OR_YUM=/usr/bin/dnf
SYSTEMCTL=/usr/bin/systemctl

all:

install:
	install -D -m0755 clatd $(DESTDIR)$(PREFIX)/sbin/clatd
	pod2man --name clatd --center "clatd - a CLAT implementation for Linux" --section
	    8 README.pod $(DESTDIR)$(PREFIX)/share/man/man8/clatd.8 && gzip -f9 $(DESTDIR)
	    $(PREFIX)/share/man/man8/clatd.8 || echo "pod2man is required to generate
	    manual page"
	if test -d "$(DESTDIR)$(SYSCONFDIR)/systemd/system"; then install -m0644 scripts/
	    clatd.systemd $(DESTDIR)$(SYSCONFDIR)/systemd/system/clatd.service ; fi
	if test -d $(DESTDIR)$(SYSCONFDIR)/NetworkManager/dispatcher.d; then install -
	    m0755 scripts/clatd.networkmanager $(DESTDIR)$(SYSCONFDIR)/NetworkManager/
	    dispatcher.d/50-clatd; fi

installdeps:
	if test -x "$(DNF_OR_YUM)"; then $(DNF_OR_YUM) -y install perl perl-IPC-Cmd perl-
	    Net-IP perl-Net-DNS perl-File-Temp perl-JSON iproute nftables; fi
```

8.2 HTCondor Configuration

Issues may arise from using short hostnames derived from /etc/hosts which we avoided by using IPv6 literals for CONDOR_HOST, COLLECTOR_HOST, and UID_DOMAIN.

References

[1] The George Washington University. *Research Technology Services – GW IT*. https://it.gwu.edu/research-technology-services. Accessed June 2025.

[2] Capital Area Advanced Research and Education Network (CAAREN). https://caaren.org. Accessed June 2025.

[3] Thain, D., Tannenbaum, T., and Livny, M. (2005). *Distributed computing in practice: the Condor experience*. Concurrency and Computation: Practice and Experience, 17(2-4), 323–356.

[4] Open Science Pool (OSPool). https://osg-htc.org/services/open_science_pool.html. Accessed June 2025.

[5] Warewulf Project. *Warewulf – Scalable Systems Management for HPC*. https://warewulf.org. Accessed June 2025.

[6] Gallo, A., Bonica, R., and Aelmans, M. (2022). *Production Deployment of TCP Authentication Option*. RIPE Labs. https://labs.ripe.net/author/andrew-gallo/production-deployment-of-tcp-authentication-option/

[7] Giuliano, L., Lenart, C., and Adam, R. (2025). *TreeDN: Tree-Based Content Delivery Network (CDN) for Live Streaming to Mass Audiences*. RFC 9706. https://tools.ietf.org/html/rfc9706

[8] Anderson, T., Byrne, C., and Bush, R. (2013). *464XLAT: Combination of Stateful and Stateless Translation*. RFC 6877. https://tools.ietf.org/html/rfc6877

[9] Anderson, T. *clatd – A 464XLAT CLAT implementation for Linux*. GitHub repository. https://github.com/toreanderson/clatd. Accessed June 2025.