SCALING A NUIX REST CLUSTER TO ONE TERABYTE PER HOUR

Ultra-fast processing of real-world data using a Nuix REST Cluster running in Amazon Web Services

OBJECTIVE

Increasing volume and complexity of data continue to challenge the ability to defensibly process vast amounts of unstructured content for eDiscovery, investigations, regulatory compliance and other use cases. To address the need for increasing throughput, Nuix has developed a new Nuix Core Engine REST clustering capability.

In benchmarking, we found a Nuix REST Cluster running in Amazon Web Services (AWS) exceeded 1 terabyte (TB) per hour processing throughput on complex real-world data for nominal AWS infrastructure costs. While 1 TB per hour is an arbitrary milestone, it effectively demonstrates that the Nuix cluster can scale to meet the most demanding organizational needs.

Here is a summary of our benchmarking setup and results:

- **Test 1 (Ubuntu):** The cluster deployed 180 Nuix workers, each assigned 11GB of memory, across 10 Nuix REST instances, each with its own Nuix case. It achieved a **sustained processing rate of 1086.66 GB/h** (or 6.037GB/h per worker) on Ubuntu without HCL Notes (formerly IBM Lotus Notes) data. This represents a 929% increase in total throughput and a 67% reduction in per-GB infrastructure costs as compared to a single instance.

- **Test 2 (Windows):** The cluster deployed 198 Nuix workers, each assigned 11 GB of memory, across 11 Nuix REST instances, each with its own Nuix case. It achieved a **sustained processing rate of 1038.29 GB/h** or 5.2439 GB/h per worker on Windows with HCL Notes data. This represents an 890% increase in total throughput and a 74% reduction in per-GB infrastructure costs as compared to a single instance.

NUIX CORE ENGINE REST CLUSTER

Nuix introduced clustering into the Core Engine REST service as a result of conversations with customers and partners. We learned that multiple organizations had already built or were in the process of implementing their own orchestration layers around the Nuix Core Engine to achieve greater scale and more efficient use of computing resources. Customers and partners sought to manage work across multiple virtual or physical machines to optimize tasks, increase automation, and pursue an always-on model appropriately sized to their needs.

Given these insights, we designed the Nuix REST Cluster to provide:

- Horizontal scalability
- Optimization of computing resources
- Reduction of bespoke code under maintenance.

The solution involved creating a single point of interaction for the cluster, introducing a layer of abstraction that ensured each function was performed on an appropriate member node without the invoker needing to know which node was doing the work (see Figure 1).

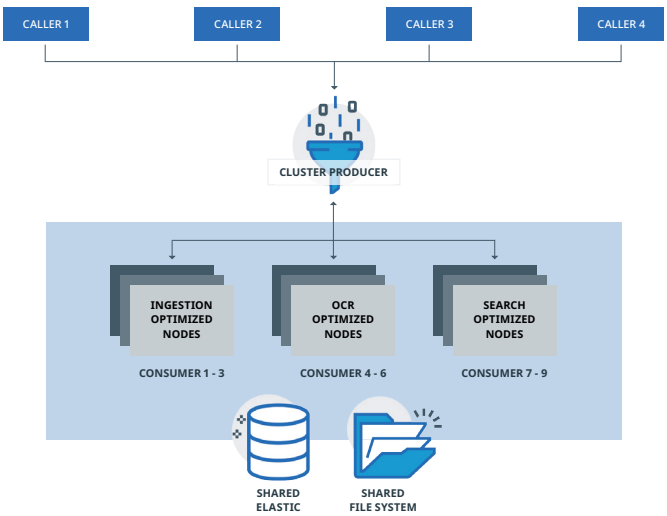

Figure 1: High-level overview of the Nuix REST Cluster

TASK PROCESSING USING PRODUCER AND CONSUMER NODES

Nuix designed the REST Cluster to address horizontal scalability concerns by reducing resource contention over local hardware. Nodes that participate in the REST Cluster support this goal and are assigned one of two roles: producer or consumer.

PRODUCER NODES

Producer nodes can respond to synchronous tasks, such as returning function status, and are responsible for placing asynchronous worker-based operations in the cluster task queue. The API checks the queue status, including queue capacity, size and the status of tasks in the data buffer. Producer nodes do not perform asynchronous tasks or worker-based operations such as processing, optical character recognition (OCR), or export.

CONSUMER NODES

Consumer nodes pick up and execute asynchronous or worker-based operations. Using a filter chain, these nodes poll the task queue and take on the tasks they are capable of processing. If all conditions pass the filters, then the consumer can lock and claim the task. This results in a natural load-balancing effect.

The following describes the filters that are currently implemented.

Consumer node filter
Case exists filter
Ensures the consumer node has local access to the case the task is running against
Task status filter
Ensures the consumer node will only claim a task in the NOT_STARTED state
Capacity filter
Ensures the consumer node will only claim a task if the task executor on the node has capacity
Node tag filter
If node steering has been applied to a task, ensures the consumer node will only claim a task if the task has been directed to this node
Worker based task filter
Ensures the consumer node will only claim a task provided it has enough workers to complete the task

Consumer nodes that are stressed or do not have the resources to complete a task cannot pick up a task for execution.

Task management within the cluster is shown below in Figure 2.

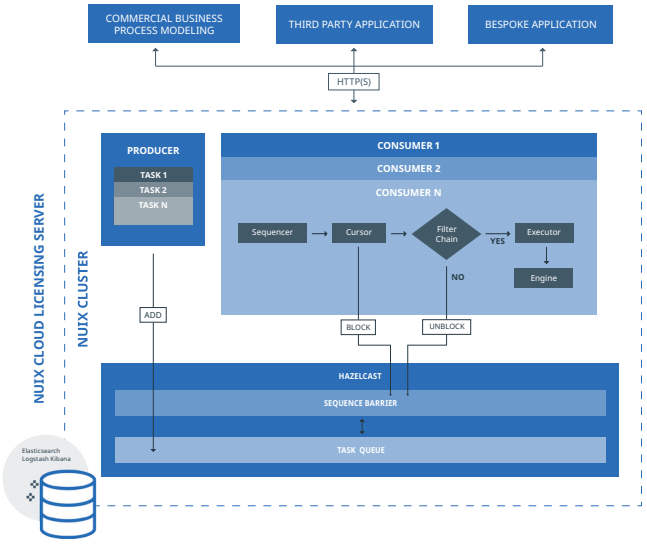

Figure 2: Task management framework in the Nuix REST Cluster

The Nuix Core Engine REST service is not intended to be a workflow or business process management (BPM) tool. For those needs, we recommend leveraging Nuix Automation, a partner application such as Rampiva Automate, or an enterprise BPM such as Microsoft Flow or Camunda.

The REST service's ability to cluster is intended to support:

- Implementation of bespoke applications
- Integration of best of breed systems
- OEM use cases
- Automating processes using an enterprise BPM, where separating the concerns of workflow and scale is a consideration.

For more information on common error scenarios that may occur when using the cluster, including fault tolerance, recovery and exception management, see the Cluster Error Scenarios section of the available documentation.

TEST COMPONENTS

We ran a series of benchmarking tests on AWS infrastructure using the Nuix Core Engine REST service and Elasticsearch as well as required dependencies.

TEST CONFIGURATION

The following table describes our test configuration.

Test 1 (Ubuntu):	Test 2 (Windows):
11 r5.8xlarge instances for the Nuix Core Engine REST Cluster • 1 producer node • 10 consumer nodes	12 r5.8xlarge instances for the Nuix Core Engine REST Cluster • 1 producer node • 11 consumer nodes

Both tests used the same 21 m5.2xlarge instances for the Elasticsearch back end, specifically:

- 3 master nodes
- 3 coordinator nodes
- 15 data nodes

We used Terraform to deploy the instances and Ansible to install and configure the Nuix Core Engine REST service. For more information on the configuration of the test environments, please download the extended version of this paper.

TEST DATA

Using meaningful real-world data for this benchmark was a priority. The Nuix Solutions Consulting team assembled 146 GB of diverse, complex and containerized data to ensure the processing throughput rates in this paper were applicable to real world scenarios. We replicated this data to create a corpus exceeding 1 TB. Before testing, we manually uploaded the data to an AWS S3 bucket, then downloaded to a shared Elastic File System (EFS) disk for the Ubuntu test and an Elastic Block Store (EBS) disk for the Windows test. For the Windows test, the EBS file system was exported as a network share, using Microsoft's Server Message Block (SMB) protocol to all REST instances.

To see a summary of processed test data by item kind from the Windows test, which included HCL Notes test data, please download the extended version of this paper.

The following table provides a summary of processed data by item kind from the Windows test.

Item kind	Item count	Total audited count*	Total audited size** (GB)	Total digest input size (GB)
calendar	5,830	5,830	0.060	0.140
chat-conversation	11	11	0.000	0.000
chat-message	104	104	0.000	0.000
contact	824	824	0.000	0.000
container	90,068	0	0.000	131.570
database	1,341	1,250	0.710	0.980
document	65,329	65,292	25.870	25.870
drawing	246,142	504	0.170	65.110
email	113,095	113,095	1.670	42.350
image	590,706	35,342	2.720	72.310
log	328	4	0.000	0.310
multimedia	1,266	1,266	0.720	0.720
no-data	2,210	2,196	0.000	0.000
other-document	76,885	75,107	0.550	0.550
presentation	36,191	36,191	80.010	80.010
spreadsheet	83,999	83,997	35.250	35.250
system	87,881	83,251	14.150	14.150
unrecognised***	136,253	135,384	86.860	86.870

TEST PROCEDURE

To perform these tests, we used LoadRunner 2020 to issue concurrent REST calls over HTTP. We applied processing job settings – including case evidence, data processing, parallel processing and other settings – using Nuix processing profiles. To access these processing profiles, please download the extended version of this paper.

While tests were in progress, we monitored server resources using CloudWatch and Kibana.

To successfully process all the emails in the test corpus, it was necessary to exclude evidence, Nuix case and worker temp directories from Windows Defender antivirus real-time monitoring because we intentionally included known malware (W32/MyDoom-O) in the data set.

* "Audited count" is a Nuix classification that identifies the number of material items in the test data set. Audited count excludes objects that are considered noise or immaterial to the goals of an investigator or reviewer.

* "Audited size" is a Nuix classification that identifies the size of material items in the test data set. Audited size excludes objects that are considered noise or immaterial to the goals of an investigator or reviewer.

*** "Unrecognised" is a Nuix data classification that identifies miscellaneous data, including plain text and binary files, in the test data set. As the test data contained disk images, a large number of binary (that is, application/octet-stream) items were present in the set.

To view item kind summaries for both tests – as well as specific breakdowns of MIME types and item types – download the extended version of this paper.

TEST RESULTS

Results were consistent with expected values for Ubuntu and Windows tests given the variance in operating systems and test data.

SCALING AND UTILIZATION

EFS and EBS utilization, CPU utilization of REST nodes and CPU utilization of Elasticsearch nodes performed as expected. This demonstrates that given a well-resourced Elasticsearch cluster, ingestion speed can scale linearly from one consumer node to the number of consumer nodes needed to achieve the 1 TB/h milestone while member nodes maintained full utilization.

A review of the resource usage charts (Figure 3 and Figure 4) show that our testing met this expectation.

Windows CPU utilization charts for the Nuix Core Engine REST Cluster and Elastic cluster are available in an extended version of this whitepaper. To access these charts, please download the extended version of this paper.

As we increased the number of consumer nodes within the cluster, we found performance scaled linearly, resulting in the desired efficiencies.

CPU UTILIZATION (%)

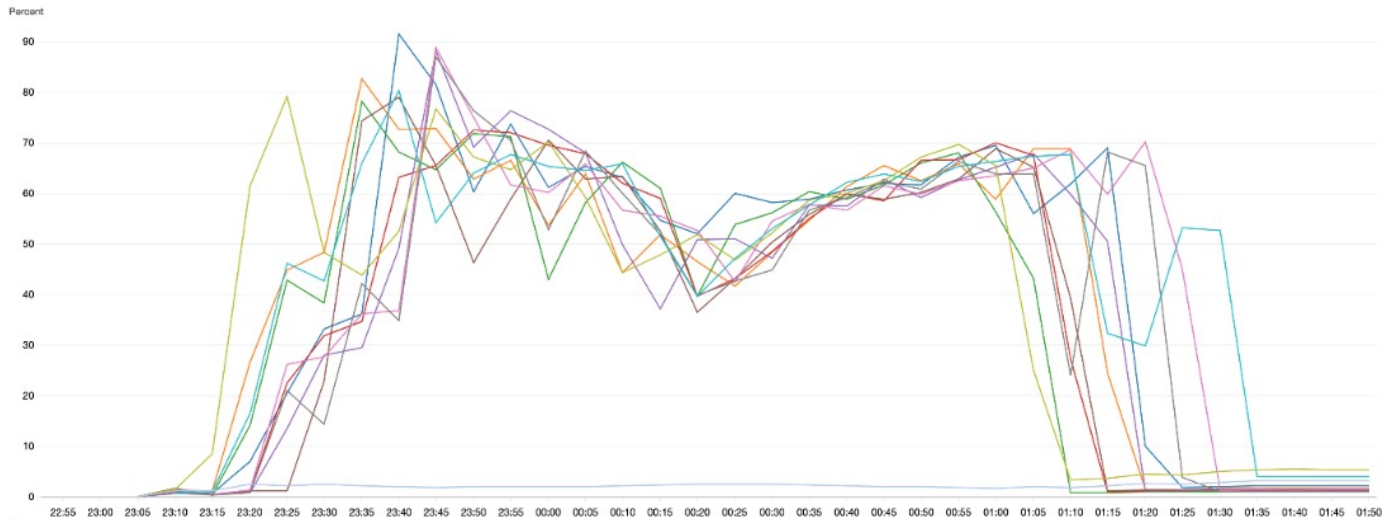

Figure 3: Nuix Core Engine REST CPU utilization (Ubuntu)

CPU UTILIZATION (%)

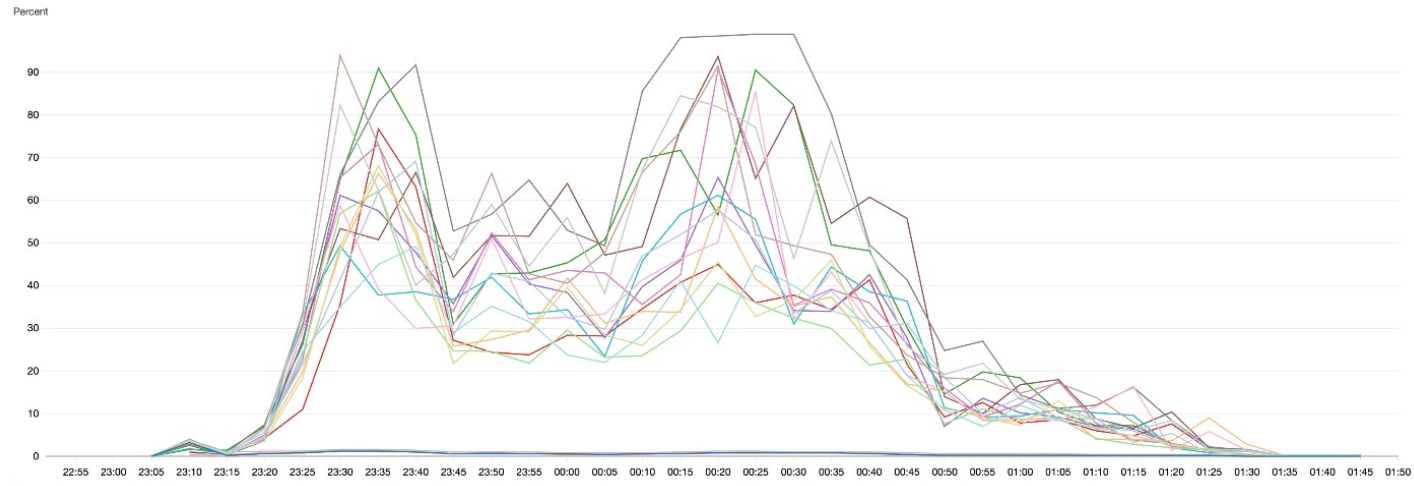

Figure 4: Elasticsearch Nodes CPU utilization (Ubuntu)

INFRASTRUCTURE COSTS

The following tables show the AWS infrastructure costs related to each test in its specific environment. They do not include software licensing costs and should not be interpreted as "per Gig" pricing. They show that increasing the number of nodes delivers near-linear scalability of throughput and achieves economies of scale of more than two-thirds in the infrastructure costs per GB of data processed.

WINDOWS TEST

AWS infrastructure costs for the Windows test

Number of consumer nodes	Throughput rate (GB per hour)	Throughput % increase	Total cluster cost per hour	Cluster cost per GB	Reduction in per-GB infrastructure costs
1	113.26	-	$18.25	$0.161	-
5	503.56	444%	$33.06	$0.066	59.0%
11	1052.23	929%	$55.27	$0.053	67.1%

UBUNTU TEST

AWS infrastructure costs for the Ubuntu test

Number of consumer nodes	Throughput rate (GB per hour)	Throughput % increase	Total cluster cost per hour	Cluster cost per GB	Reduction in per-GB infrastructure costs
1	122.10	-	$15.51	$0.127	-
5	573.17	469%	$24.43	$0.042	66.9%
10	1086.62	890%	$35.57	$0.032	74.8%

More information on AWS infrastructure costs is in an extended version of this whitepaper. To access that version, please download the extended version of this paper.

CONCLUSION

This testing effort focused on processing data – or normalizing heterogeneous unstructured data – from real-world use cases into a Nuix index. The results met expectations by achieving a throughput rate of over a terabyte per hour while efficiently utilizing the cluster's resources. Following this exercise, we identified opportunities for further study, including:

- Adding greater volume and more complex data types to the test data set
- Publication of the data set used for testing, following removal of all personally identifying information
- Expanding the benchmark to include post-processing steps such as deduplication, de-NISTing, date filtering, keyword searching, and promotion to review, to benchmark a representative "time to review" workflow.

What's next for the Nuix Core Engine REST Cluster? Nuix will keep investing in this area to make complex data processing jobs faster, cheaper and simpler. To achieve this, we will focus on:

- Introducing auto-provisioning for AWS, Microsoft Azure, and on-premises installations allowing the Nuix REST Cluster to run as a single instance while unutilized and spin up instances as needed
- Introducing the ability to assign all CPU, memory, and other resources of a cluster to a single, very large job across many instances using Nuix's existing brokers and agents
- Simplifying logging, debugging, and diagnostics retrieval to reduce the complexity of troubleshooting.

FEEDBACK

We welcome feedback and are interested in improving our benchmarking studies in the future. If you have questions, comments or suggestions, please reach out to sdk@nuix.com. We also encourage continued conversation through the Nuix Community Forums.

CONTRIBUTORS

- Cristina Aragon
- John Bargiel
- Daniel Berry
- Emilie Bertolino
- Bruce Brown
- Karen Burkardsmaier
- Erin Decker
- Nicholas DiPasquale
- Sean Dougherty
- Syed Hussainy
- Frank Marrone
- Josh Mehlman
- Kyle Pfueller
- Patrick Rice
- David Sitsky
- Stephen Stewart
- Phillip Weber
- Ryan White
- Dana Wilson

DISCLAIMERS

Last updated on: March 22, 2022

This publication is intended for informational purposes only. The information contained herein is provided “as-is” and is subject to change without notice. Although reasonable care has been taken to ensure that the facts stated in this publication are accurate, no representation or warranty, expressed or implied, is made as to the fairness, accuracy, or completeness of the information. The use, reproduction, and/or distribution of any Nuix software described in this publication requires an applicable software license. Nuix (and any other Nuix trademarks used) are trademarks of Nuix Ltd. and/or its subsidiaries, as applicable.

nuix

Nuix (www.nuix.com, ASX:NXL) creates innovative software that empowers organizations to simply and quickly find the truth from any data in a digital world. We are a passionate and talented team, delighting our customers with software that transforms data into actionable intelligence and helps them overcome the challenges of litigation, investigation, governance, risk and compliance.

APAC
Australia: +61 2 8320 9444

EMEA
UK: +44 203 934 1600

NORTH AMERICA
USA: +1 877 470 6849