A Hierarchal Framework Offering Insights via Single view of HPC Systems Under NSM

(Objective, Architecture, APIs, Deliverables & Metrics)

सी-डैक
CDAC

Centre for Development of Advanced Computing

Contents

Glossary of Terms and Abbreviations

Terms	Definition
C-DAC	Centre for Development of Advanced Computing
NSM	National Supercomputing Mission
DR	Disaster Recovery
HPC	High Performance Computing
API	Application programming interface
HTTP	Hypertext Transfer Protocol
REST	Representational State Transfer
HA	High Availability
NKN	National Knowledge Network
ACL	Access Control List
IPMI	Intelligent Platform Management Interface
SNMP	Simple Network Management Protocol
BMC	Baseboard Management Controller

Introduction:

Under NSM (National Supercomputing Mission), national academic and R&D institutions will going to get over 70 HPC (High-Performance Computing) facilities.

"High Performance Computing most generally refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, or business."

It will be extremely important from the governing point of view as well as from the administration point of view that, how these large number of system behaving individually and collectively. To know the behavior of such systems, collection of data from (possibly) every system component and at very finer level (in terms of duration/period) will be necessarily important.

The proposed framework will be instrumental in pursuing the above goal. This framework will not only provide insight into the System workflow but also about the Job behavior on the systems. Outcome of this framework i.e large amount of normalize data will be crucial for decision making in defining policies and strategy for the future HPC systems in the country.

Purpose of the document:

The purpose of this document is to provide insight into our proposed framework so that reader could give comments, suggestions. It explains about the architecture, category of the systems / sub-systems, request pay load and response. As a result of reader's comment/suggestions, we will get inputs on existing features as well as get to know about the points that we have missed.

We will welcome comments and suggestions at mailing list: npsfhelp@cdac.in with the below given Subject line. We request readers to not deviate from the provided Subject Line for the e-mail.

Subject Line: ***A hierarchal framework offering insights via single view of HPC systems under NSM***

How This Document Is Organized

- Terminologies being used – This section describes about the terminologies being used in the HPC systems.
- Objective - Provides overall objective of this document
- Architecture – Explains about the proposed framework's 3 Tier architecture in detail
- Present Framework at NPSF – About the current data collection system for PARAM Yuva II at NPSF.
- Deliverables – Details about the tasks going to be accomplished in the project life cycle.
- Development Model- It gives detail about the request generated and what will be the response given by the adaptors with required payload.
- Annexure A - List of all the category and sub-category metrics which we are going to collect from the system.

Terminologies being used:

- **Cluster -** Clusters are basically many computers connected together with a network and centrally coordinated by some special software. Because the computers are usually physically very close together, the common term for a high performance computer today is a cluster.

  The size of an HPC cluster usually refers to how many processors, or how many cores, it has.

- **Job** - In computing, a job is a unit of work or unit of execution (that performs said work). A component of a job is called a task or a step (if sequential, as in a job stream). As a unit of execution, a job may be concretely identified with a single process, which may in turn have sub processes (child processes; the process corresponding to the job being the parent process) which perform the tasks or steps that comprise the work of the job; or with a process group; or with an abstract reference to a process or process group, as in Unix job control.

  Jobs can be started interactively, such as from a command line, or scheduled for non-interactive execution by a job scheduler, and then controlled via automatic or manual job control. When somebody types commands in a login shell and see a response displayed, then he/she is working interactively. To run a batch job, you put the commands into a text file instead of typing them at the prompt.

- **Batch System-**. You submit Jobs to the *batch system*, which will run it as soon as resources become available. It consists of a resource manager for ex: Torque and a scheduler for ex: Maui, Moab.

  Resource Manager - The resource manager is software, which communicates with users submitting jobs and all of the compute nodes on the system. It monitors memory usage and processor utilization for all jobs. The resource manager communicates this information to Scheduler.

  Scheduler - Scheduler is the job scheduling program and determines when and where jobs can get scheduled.

  The batch system allows users to submit jobs requesting the resources (nodes, processors, memory, GPUs) that they need. The jobs are queued and then run as resources become available. The scheduling policies in place on the system are an attempt to balance the desire for short queue waits against the need for efficient system utilization.

- **Queue –** In HPC systems, job queue (sometimes batch queue), is a data structure maintained by job scheduler software containing jobs to run.

Users submit their "jobs", to the queue for batch processing. The scheduler software maintains the queue as the pool of jobs available for it to run.

Multiple batch queues might be used by the scheduler to differentiate types of jobs depending on parameters such as:

- job priority
- estimated execution time
- resource requirements

- **Storage -**

- **E-Bank/Accounting** – It is an accounting system in HPC. It tracks resource usage on High Performance Computers and acts much like a bank, establishing accounts in order to pre-allocate user and project resource usage over specific nodes and timeframe. It also provides balance and usage feedback to users, managers, and system administrators.

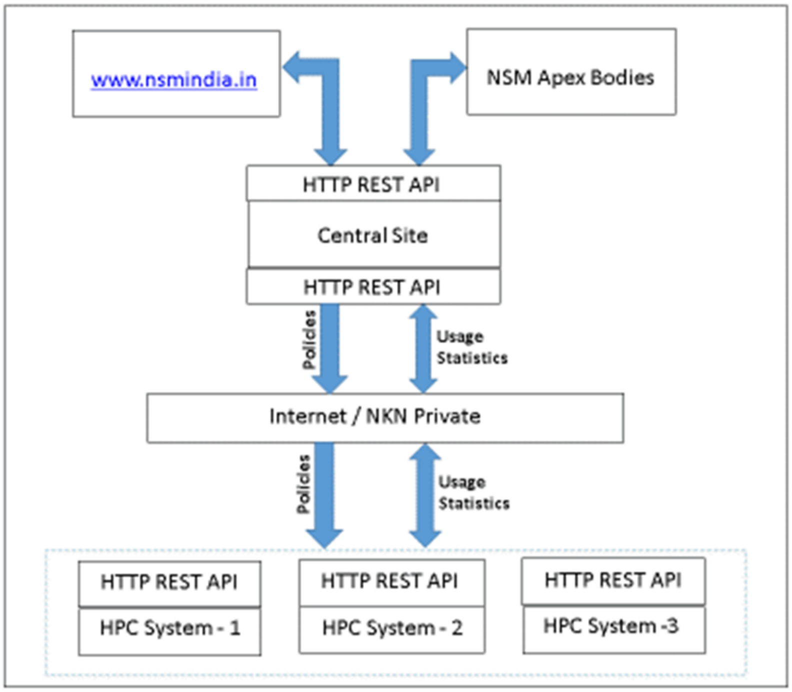
Objective:

The most prominent use case of the proposed framework will be that it will provide Graphical View of collected statistics using Dashboard at www.nsmIndia.in for public at large and ACL for restricted functions. Same can be provided with requisite data for peer review to apex bodies and likes.

As all the proposed NSM systems will be distributed across the Nation for academic and R&D institutions, system resources won’t be solely used by single organisation/ institutions, instead CPU time will be allocated in 60:40 ratio for the in-house project and outside projects as well. This framework will play an important role in Centralized and Proportionate allocation of CPU time for the projects and can simplify the approving process. Data coming from all the Tier 1 systems will be normalized at central location for quick access and presentation

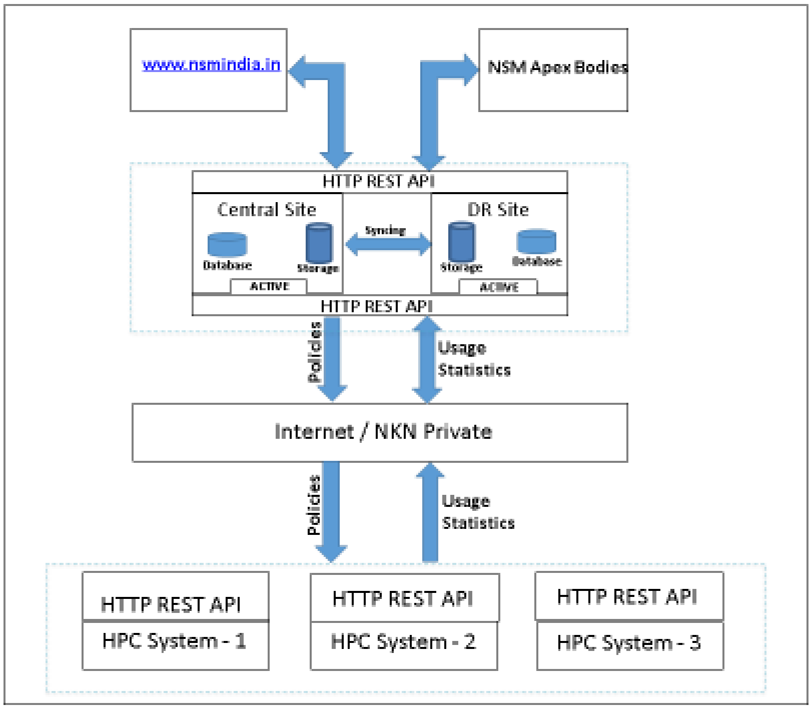
This framework will be useful in data acquisition for jobs, system utilization, other related parameters of interest, reporting capabilities. This data will be helpful in knowing about the System Availability statistics and its health.

Architecture:

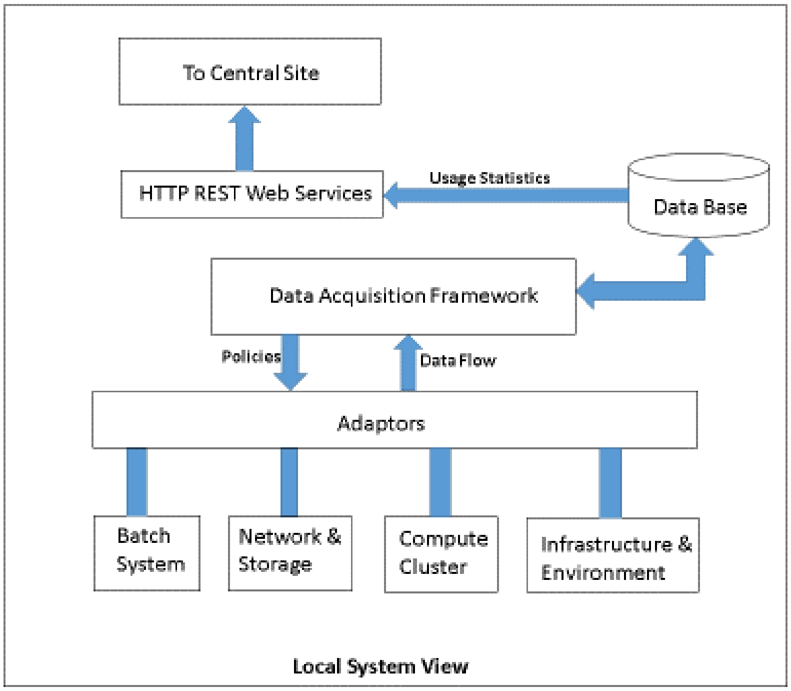


In the above architecture diagram, user/consumer who wants to see graphical representation of the statistical data will visit a website such as www.nsmindia.in. From there, our three tier architecture will come into action. Communication from www.nsmindia.in to “Central Site” shown in the image above will happen over HTTP protocol using REST API. Central Site is a site where the data for all the HPC systems (lowest tier) will be stored. This data from all HPC systems will be either pulled by the top tier systems or pushed from the underlying systems over a specific interval, this communication will again happen over HTTP using REST. Every communication happening between these tiers will be either over internet or over NKN.

NSM Apex Bodies as shown in the above diagram will be the policy and decision makers about the NSM systems and can govern those policies to all the systems using this framework. Like any other request for data, request for policy change will also traverse from top tier to bottom but it can be changed only by the system administrators of the local HPC systems.



In the above architecture diagram, the central site where collected data from all the Tier 1 systems will be store. This central will be backed by the DR site as shown in the diagram. Central site and DR site will always be in sync in terms of everything for ex. Database, service configuration etc.. And will be in Active-Active/HA (High Availability) mode.



In the above diagram, data from all the sub-systems like Batch, Network, Storage, Compute and Infrastructure will be pulled by a Framework component called as Adaptors. There will be one or more than one adaptors for each sub-systems. The job of these adaptors will be to pull the data for each sub-system components (Metrics, Can find complete details in Annexure A) and will be given to the Data Acquisition Framework. This framework will eventually store this data into Database in the defined/configured time interval or pattern. This data will be then consumed by the REST Web services at central location and store it for further use at above Tier.

Present Framework at NPSF:

There is a setup of Hadoop MapReduce cluster on 4 nodes of the PARAM Yuva II system. The Hadoop MapReduce framework is being used to achieve the data collection in nearly synchronized fashion. MapReduce jobs runs for data collection on those nodes, these jobs collects following data from each node of the cluster such as power supplies, temperature sensors, perform system accounting like load on the node, I/O access, memory usage, ***partition***, jobs running on the node, etc..

Partition is a logical grouping of resources such as nodes for better utilization of cluster resources. Using the present framework we are getting per partition statistics like number of nodes configured, number of nodes up/down, list of all nodes in that partition, etc...

Present framework also collects Torque (Resource Manager) and Maui (Scheduler) statistics like active jobs, blocked jobs, idle jobs, cluster utilization, total nodes, active nodes, free nodes, etc. With the use of Ganglia gmetrics with this framework allows system administrators to see these stats in the Ganglia Web Interface. This framework will leverage upon Baseboard Management Controller (BMC) IPMI, SNMP interfaces available on the system/nodes.

Deliverables:

- A hierarchal framework is required to collect various Parameters and stats and other data pertaining to Performance, monitoring and cluster utilization.
- The above framework can be utilized for Policy framing, Proportionate allocations of CPU time.
- In addition the framework can be used for Data analysis and report preparation.
- The above data can be the input for the R & D projects under NSM.
- Health of the system and individual cluster services
- Limited and controlled access to public at large

Development Model

Interface between Framework and Adapters

Data collection framework calls interfaces from adapters and it will respond with the requested data, this communication/data exchange will happens using XML. This section gives in detail of format of data exchange between the two.

This section is specifically written about the sub-category "Batch" in "Compute" Category. Find category and sub-category table at "Appendix A".

1.1 Batch Request

This message is sent from Data Collection Framework to get the data from batch system for obtaining the required data can be provided by Adaptors. The XML format and details of the attributes is as follows:

Batch	Version 0.1 <BatchReq ver="" ts="" txn=”” an=”” on=”” dn=”” pan=”” cup=”” cun=”” cupr=”” aj=”” ij=”” bj=”” bhp=”” bhc=””> </ BatchReq >

Element Name: BatchReq (Mandatory)

- Description: Root element of the input xml
- attributes: Table below

Sr No	Attribute	Long Name	Description
	ts	Timestamp	Timestamp when the response is generated (at KUA). This is in the format of “YYYY-MM-DDTHH:mm:ss.sss”
	txn	Transaction id	This is an alpha-numeric string of maximum length 50 .Only supported characters are A-Z, a-z, 0-9, period, comma, hyphen, backward & forward slash, left & right parenthesis, and colon. This will be generated by the framework.
	An	activeNodes	Nodes executing jobs including partially occupied ones. Value can be 1 or 0
	On	offlineNodes	Nodes up but not eligible for job execution. Value can be 1 or 0
	Dn	downNodes	Nodes configured but down. Value can be 1 or 0
	Pan	partialActiveNodes	Nodes running jobs but with free procs. Value can be 1 or 0
	tn	totalNodes	Total nodes in the HPC system. Value can be 1 or 0
	Cup	clusterUtilizationPecentage	Cluster utilization in percentage. Value can be 1 or 0

	Cun	clusterUtilizationNodes	Cluster utilization in terms of nodes running jobs. Value can be 1 or 0
	Cupr	clusterUtilizationProcs	Cluster utilization in terms of processors utilized for running jobs. Value can be 1 or 0
	Aj	activeJobs	Total number of jobs currently running on cluster. Value can be 1 or 0
	Ij	idleJobs	Total number of valid jobs waiting for execution. Value can be 1 or 0
	bj	blockedJobs	Total number of jobs ineligible for execution for various reasons. Value can be 1 or 0
	bhp	backlogHoursProcs	Total number of processors hours required to clear the Idle jobs. Value can be 1 or 0
	bhc	backlogHoursCluster	Total number of hours required given the current cluster state to clear the Idle jobs. Value can be 1 or 0

Note:-

Element Name: Batch (Mandatory)

- Description: Root element of the input xml
- attributes: Table below

1.2Batch Response

This message is sent as a response from Adapters to Data collection framework, in regard to data per request. The XML format and details of the attributes is as follows:

DCReqAdapters	**Version 0.1** <BatchResp ver="" ts="" txn=”” an=”” on=”” dn=”” pan=”” cup=”” cun=”” cupr=”” aj=”” ij=”” bj=”” bhp=”” bhc=””> </ BatchResp >

Element Name: BatchResp (mandatory)

- Description: Contains the data information.
- Attributes: Table below

Sr No	Attribute	Long Name	Description
	ts	timestamp	Timestamp when the response is generated (at KUA). This is in the format of "YYYY-MM-DDTHH:mm:ss.sss"
	txn	Transaction id	Agent specific transaction identifier. This is exactly the same value that is sent within the request.
	responseCode	responseCode	Failure/Success response code. If request fails/succeeds, this attribute provides codes, description is as follows. **Response Codes** BATCH-0100 Successful Response BATCH-0102 Aggregator not able to connect adapters BATCH-0103 Timestamp is not present BATCH-0104 Transaction ID is not present BATCH-0105 Attribute list is empty BATCH-0106 Invalid XML Request BATCH-0107 Partial Response Available BATCH-0108 Connection Timeout BATCH-0109 Timestamp not in sync with Aggregator BATCH-0110 Unknown Response
	an	activeNodes	Nodes executing jobs including partially occupied ones. Value -1 states Non Availability while 0 or more is actual data
	on	offlineNodes	Nodes up but not eligible for job execution. Value -1 states Non Availability while 0 or more is actual data
	dn	downNodes	Nodes configured but down. Value -1 states Non Availability while 0 or more is actual data

	pan	partialActiveNodes	Nodes running jobs but with free procs. Value -1 states Non Availability while 0 or more is actual data
	tn	totalNodes	Total nodes in the HPC system. Value -1 states Non Availability while 0 or more is actual data
	cup	clusterUtilizationPecentage	Cluster utilization in percentage. Value -1 states Non Availability while 0 or more is actual data
	cun	clusterUtilizationNodes	Cluster utilization in terms of nodes running jobs. Value -1 states Non Availability while 0 or more is actual data
	cupr	clusterUtilizationProcs	Cluster utilization in terms of processors utilized for running jobs. Value -1 states Non Availability while 0 or more is actual data
	aj	activeJobs	Total number of jobs currently running on cluster. Value -1 states Non Availability while 0 or more is actual data
	ij	idleJobs	Total number of valid jobs waiting for execution. Value -1 states Non Availability while 0 or more is actual data
	bj	blockedJobs	Total number of jobs ineligible for execution for various reasons. Value -1 states Non Availability while 0 or more is actual data
	bhp	backlogHoursProcs	Total number of processors hours required to clear the Idle jobs. Value -1 states Non Availability while 0 or more is actual data
	bhc	backlogHoursCluster	Total number of hours required given the current cluster state to clear the Idle jobs. Value -1 states Non Availability while 0 or more is actual data

Annexure A:

Category	Sub-Category
Compute	Batch
	Cluster
	Job
	Accounting
	System
Network	NA
Storage	NA
Infrastructure	Cooling
	Power

Category: Compute

Sub-Category: Batch

Short Name of the Metric	Description
activeNodes	Nodes executing jobs including partially occupied ones
offlineNodes	Nodes up but not eligible for job execution
downNodes	Nodes configured but down
partialActiveNodes	Nodes running jobs but with free procs
totalNodes	Total nodes in the HPC system
clusterUtilizationPecentage	Cluster utilization in percentage

clusterUtilizationNodes	Cluster utilization in terms of nodes running jobs
clusterUtilizationProcs	Cluster utilization in terms of processors utilized for running jobs
activeJobs	Total number of jobs currently running on cluster
idleJobs	Total number of valid jobs waiting for execution
blockedJobs	Total number of jobs ineligible for execution for various reasons
backlogHoursProcs	Total number of processors hours required to clear the Idle jobs
backlogHoursCluster	Total number of hours required given the current cluster state to clear the Idle jobs

Category: Compute

Sub-Category: Job

Short Name	**Description**
perProcessUserCPUUtilization	User CPU utilization for every process participating in a job, for every job executed in the past and currently running
perProcessSystemCPUUtilization	System CPU utilization for every process participating in a job, for every job executed in the past and currently running
perProcessWaitCPUUtilization	Wait CPU utilization for every process participating in a job, for every job executed in the past and currently running
perProcessCPUUtilization	CPU utilization for every process participating in a job, for every job executed in the past and currently running

perProcessMemoryUtilization	Memory utilization for every process participating in a job, for every job executed in the past and currently running
CPUUtilization	Aggregate CPU utilization for a job for every job executed in the past and currently running
memoryUtilization	Aggregate memory utilization for a job for every job executed in the past and currently running

Sub-Category: System

Short Name	**Description**
perNodeUserCPUUtilization	User CPU utilization for every compute node in the cluster identified by hostname
perNodeSystemCPUUtilization	System CPU utilization for every compute node in the cluster identified by hostname
perNodeWaitCPUUtilization	Wait CPU utilization for every compute node in the cluster identified by hostname
perNodeCPUUtilization	CPU utilization of compute node identified by hostname
perNodeFreeMemory	Free memory in compute node identified by hostname
perNodeSwapMemoryUtilization	Swap memory utilization in compute node identified by hostname
perNodeCachedMemoryUtilization	Cached memory utilization in compute node identified by hostname

perNodeCachedMemoryUtilization	Buffered memory utilization in compute node identified by hostname
perNodeMemoryUtilization	Memory utilization of compute node identified by hostname

Sub-Category: Cluster

Short Name	**Description**
userCPUUtilization	User CPU utilization for entire cluster aggregated from individual compute nodes
systemCPUUtilization	System CPU utilization for entire cluster aggregated from individual compute nodes
waitCPUUtilization	Wait CPU utilization for entire cluster aggregated from individual compute nodes
CPUUtilization	CPU utilization for entire compute cluster
freeMemory	Free memory for entire cluster aggregated from individual compute nodes
swapMemoryUtilization	Swap memory utilization for entire cluster aggregated from individual compute nodes
cachedMemoryUtilization	Cached memory utilization for entire cluster aggregated from individual compute nodes
cachedMemoryUtilization	Buffered memory utilization for entire cluster aggregated from individual compute nodes
memoryUtilization	Memory utilization for entire compute cluster

Category: Storage

Sub-Category: Storage

Short Name	Description
homeTotal	Total home storage available. Fixed
homeOccupancy	Used space on home filesystem
homeOccupancyPercentage	Used space on home filesystem in percentage
homeOccupancyUser	Used space on home filesystem for an user identified by username
homeQuota	Aggregate of all user quotas
homeQuotaPercentage	Aggregate of all user quotas in percentage against homeTotal
homeQuotaUser	Quota allocated on home filesystem for an user identified by username
scratchTotal	Total size of PFS available. Fixed
scratchOccupancy	Used space on PFS
scratchOccupancyUser	Used space on PFS for an user identified by username

Category: Infrastructure

Sub-Category: Power

Short Name	Description
nodePowerSupplyUtilization	Power in watts for each power supply in node
nodePowerUtilization	Aggregate power in watts for a node
clusterPowerUtilization	Aggregate power in watts for compute cluster