



Proposal and Testing for ATLAS Distributed Analysis Monitoring System

Mariam John
UTA
03/30/2006

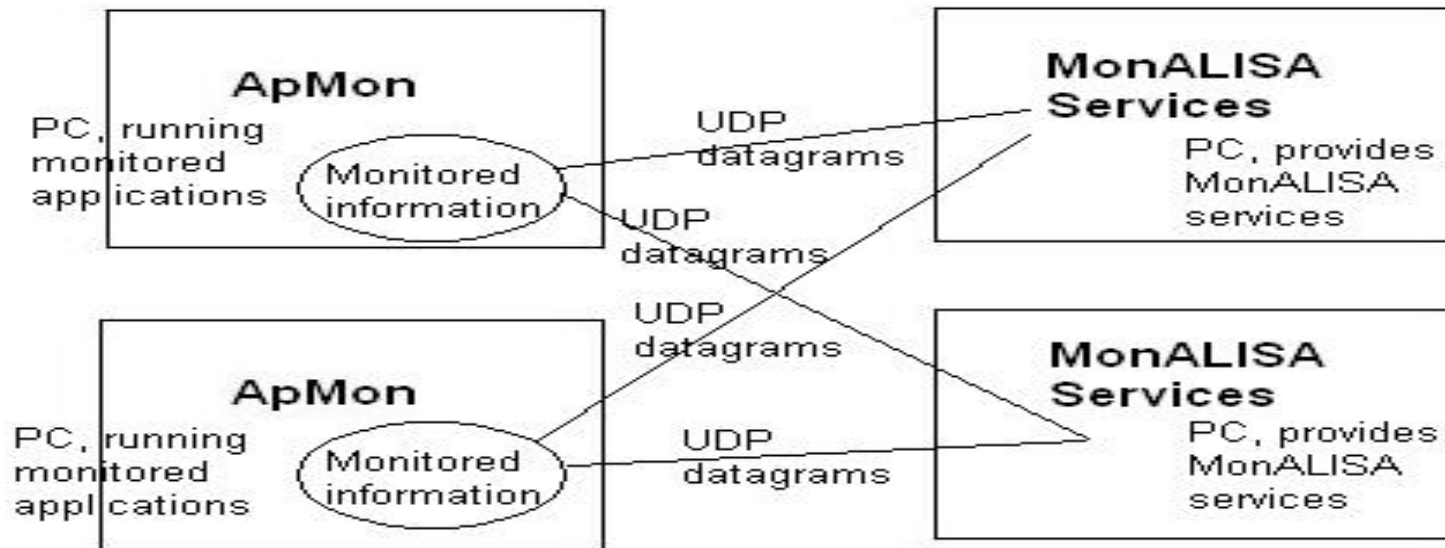


Introduction

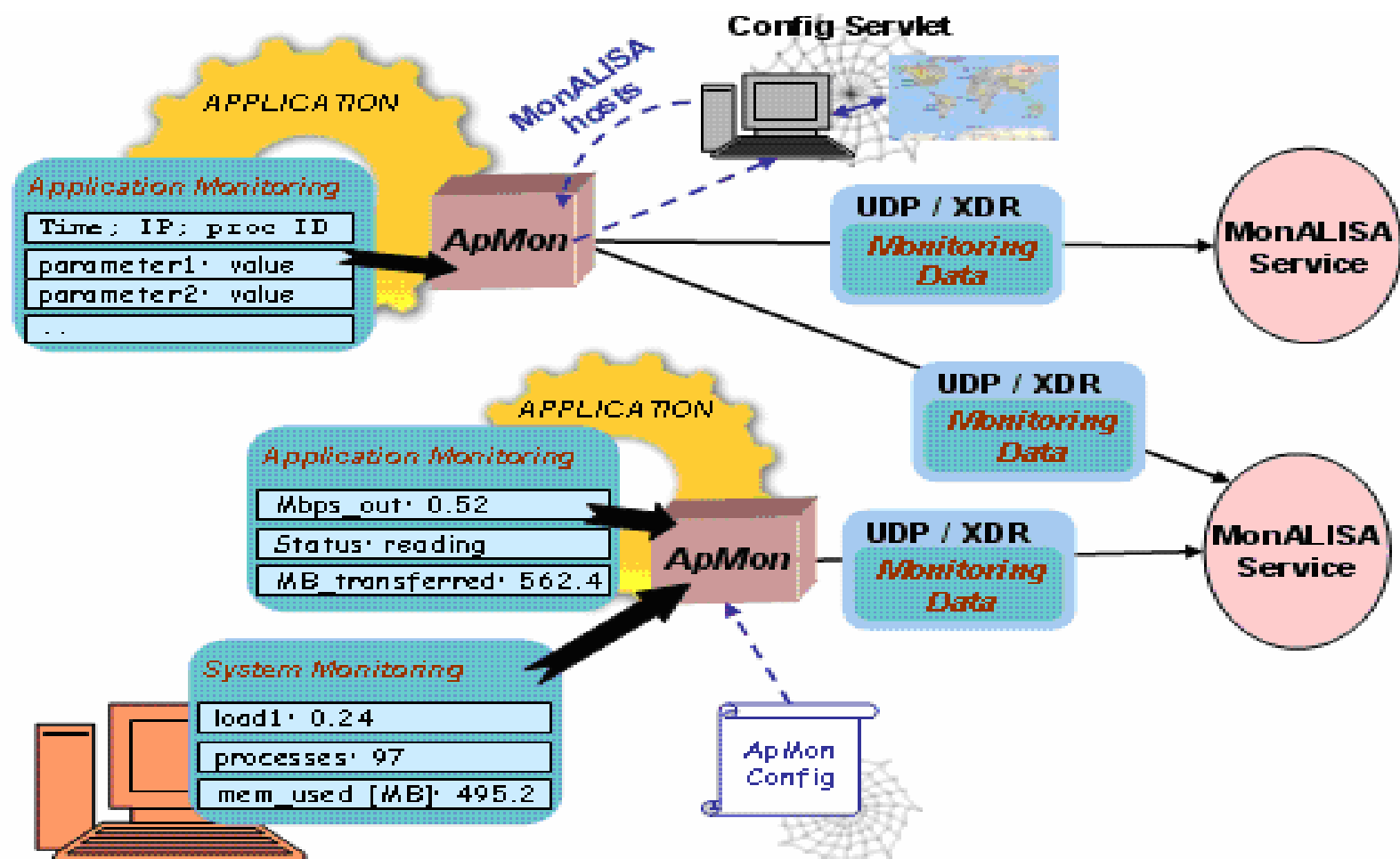
- Aim of distributed analysis is to enable ATLAS users to use grid resources for rapid analysis of data in physics and detector groups.
- Idea is to share as much backbone of the framework for distributed production and analysis.
- As a first step, we look into monitoring system for Distributed Analysis System.

Overview of MonALISA Scalability Test

- ApMon is a set of flexible API's used by applications to send monitoring information to MonALISA services.

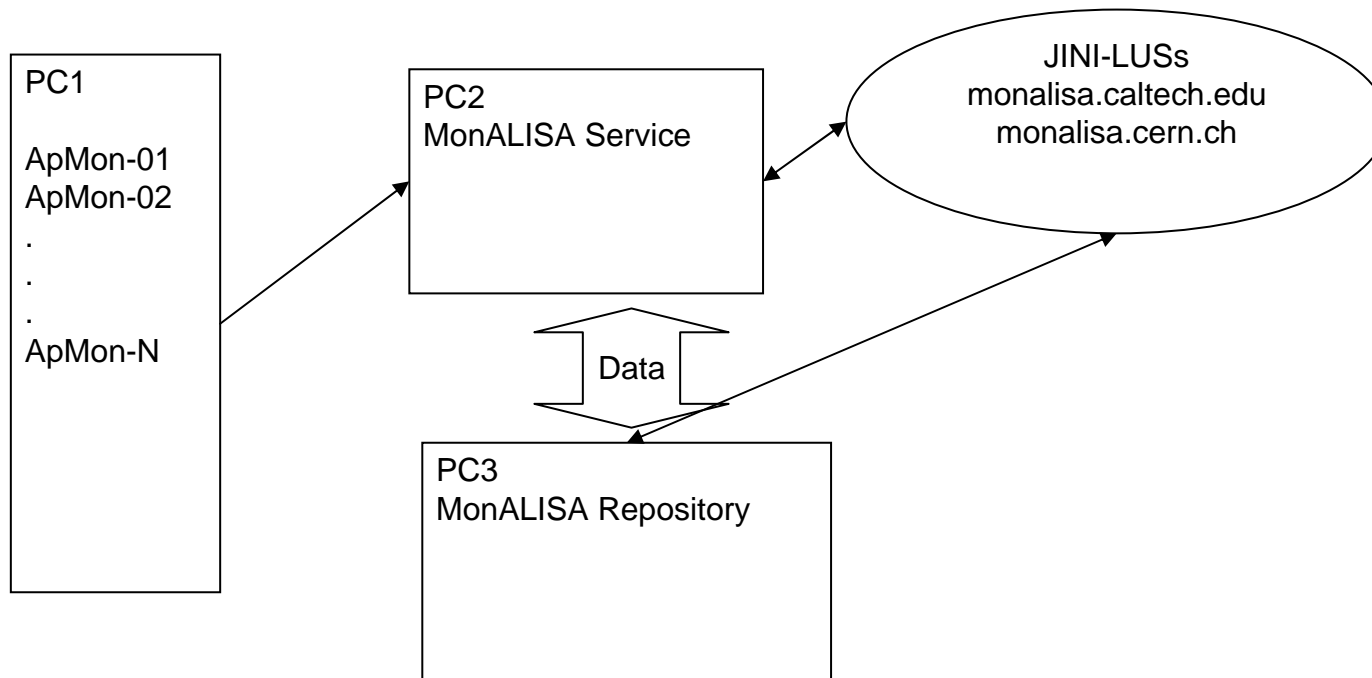


ApMon



Architecture

○ Basic Structure of the Test



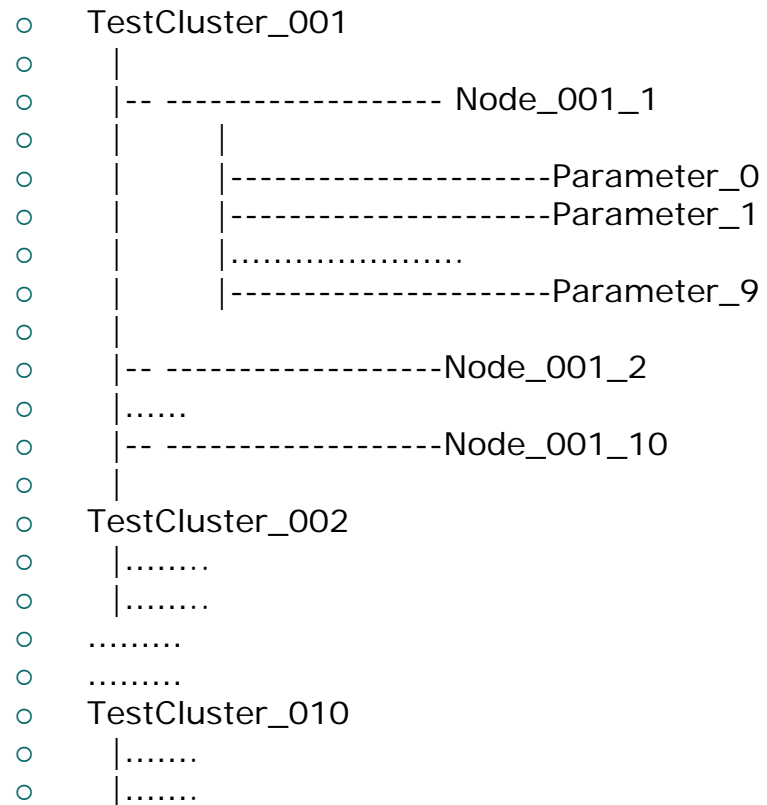


Architecture

- The basic architecture for this experiment defines three different machines:
 - PC1 – runs ApMon, which provides monitored data to the MonALISA services
 - PC2 – hosts a MonALISA service which gathers monitored information from ApMon.
 - PC3 – hosts MonALISA repository which stores the monitored data that the MonALISA service has gathered.
- In this experiment, ApMon simulates 1000 hits/sec of monitored data



Tree Structure Of ApMon programs





Mechanism

- 10 copies of the ApMon programs running at the same time.
- Each one simulates a cluster with 10 nodes.
- Each node generates 10 string parameters.
- There is a 10 millisecond interval between any two transfers of values of the parameters.
- Then, in 1 second, 1000 data grams will be sent.



Implementation of ML Repository

- Example:
 - If we want to see the values of Parameter_0 of Node_006_3 of TestCluster_006 in the past 1 minute
 - We can run:
 - `./run_client "heptest017" "TestCluster_006" "Node_006_3" "Parameter_0" -60000 0`
- Arguments (respectively):
 - farm name
 - cluster name
 - node name
 - parameter name
 - from time (in milliseconds): 60000 means from the past 60 sec
 - to time (in milliseconds): 0 means now

```
[heptest016] /home/mlrep/MLWSClients/WS-Clients/Java-Axis/getValues > ./run_client "heptest017" "TestCluster_006" "Node_006_3" "Parameter_0" -60000 0
```

Feb 7, 2006 11:57:20 AM org.apache.axis.utils.JavaUtils isAttachmentSupported

WARNING: Unable to find required classes (javax.activation.DataHandler and javax.mail.internet.MimeMultipart). Attachment support is disabled.

-----> Received values <-----

FarmName: heptest017
ClusterName: TestCluster_006
NodeName: Node_006_3
Parameters:
 Parameter_0 : 1.4642179284553853
Time: 1139334985862

FarmName: heptest017
ClusterName: TestCluster_006
NodeName: Node_006_3
Parameters:
 Parameter_0 : 1.7570228187377621
Time: 1139334985864

FarmName: heptest017
ClusterName: TestCluster_006
NodeName: Node_006_3
Parameters:
 Parameter_0 : 1.6394790068778262
Time: 1139334985864

FarmName: heptest017
ClusterName: TestCluster_006
NodeName: Node_006_3
Parameters:
 Parameter_0 : 1.0997952124786856



Conclusion

- This experiment successfully tested MonaLISA's ability to handle large number of hits to a certain extent.
- All machines in this test were within the same subnet.
- Future work would include testing MonALISA for large hits from ApMon programs running outside the subnet and also to get ApMon to monitor real jobs.