Adaptive Probing: A Monitoring-Based Probing Approach for Fault Localization in Networks

Himanshu Sharma[1], Sachin Chauhan[2]

[1]Poornima Institute of Engineering & Technology, ISI-2, Sitapura, 302022, Jaipur
himanshupiet030@poornima.org

[2]Poornima Institute of Engineering & Technology, ISI-2, Sitapura, 302022, Jaipur
sachinchauhan@poornima.org

Abstract: ***Previously for detecting faults in the networks, mainly two techniques are utilized which are monitoring and probing techniques. But both of the above techniques are used separately and advantages of both techniques are not utilized properly. In this paper, I am describing that by using both the techniques the process of fault localization can be performed easily and effectively. Using this we can have a technique named as Adaptive probing which is a combination of both the techniques. In this technique we will be having the advantages of both the above mentioned techniques i.e. monitoring and probing techniques. There are three main advantages of adaptive probing which are not present in any other technique. The main advantages are that probe traffic is reduced, localization accuracy is increased, and localization time is minimized. Further in next section, i will demonstrate this.***

Keywords: Fault Localization, Probing, Monitoring**.**

1. Introduction

The main demand of today's system is that they must be compact in size and the performance is high. For this we have to take care of its design and also ensure that there must no or negligible number of faults. But this is not possible as in decreasing size and increasing complexity increases the chance of having number of errors in the networks. So our main aim is to detect the fault or error, identify it and finally remove that fault to ensure proper working of the system.

Earlier there are two main types of method for monitoring the networks: (a) component level passive monitoring, and (b) end-to-end probing. Passive monitoring techniques includes use of monitors at each component to collect system metrics (e.g. CPU usage, memory consumption etc.) in regular interval of time. Probing-based techniques transmits test transactions (like ping, trace routes) via the network to examine the network status and faults. Passive monitoring type of techniques provide fine grained metrics, but these techniques fail to give end-to-end view of the system. Probing-based techniques, on the other hand, can give end-to-end metrics but adds an additional traffic in the system and also lacks in providing fine-grained results.

All the earlier techniques are mainly based on any of the above mentioned techniques, but the main fact is that both the techniques are not utilized with each other to have best results. When they are used individually we are not having best results like there may be large amount of time required of identifying the errors of the system. This results in lack of having a full proof network working with its maximum efficiency. In this paper, I will present the blueprint of a monitoring-based adaptive probing solution for fault identification, localizing it and finally its removal. However, the techniques used earlier which are based on probing have lack node-level view which limits its proper use. These techniques basically use the results collected through the probes. The main limitation which it has is that there is no proper coordination between probe traffic and error identification time. By using adaptive probing ,there are mainly three advantages which are (a) provide correct fault localization, (b) reduce fault localization time, and (c) reduce the additional probe traffic generated in the system. The adaptive-probing process mainly consists of two major steps. They are as follows:- 1. Firstly some probes are used at regular interval of time to check for any the presence of failure in the system. These probes just identify the existence of a failure but does not provide the exact location of faulty part of system. Our main aim in choosing the probes is that (a) probe traffic is reduced and (b) detection time is also reduced. 2. On identification of a error, some more probes are used to identify the exact position of faulty network part. In addition to above mentioned steps, the performance can be easily made better than earlier by utilizing the data collected through monitors. The monitors are present at each node. There main work is to collect metrics at each node. The data collected by the monitors which are present at each node can be utilized to figure out the performance of the system or the network. The data collected by the devices which are used as monitors is very useful in identifying the presence of error and locating it. This also optimize the probe traffic, localization time, and localization accuracy. In this process firstly, the information collected by monitors is used to get the exact working of system. Then the probes are utilized for identifying locating the error.

By following the above mentioned procedure fault identification time is made low and the performance of our system becomes high

2. Proposed Architecture

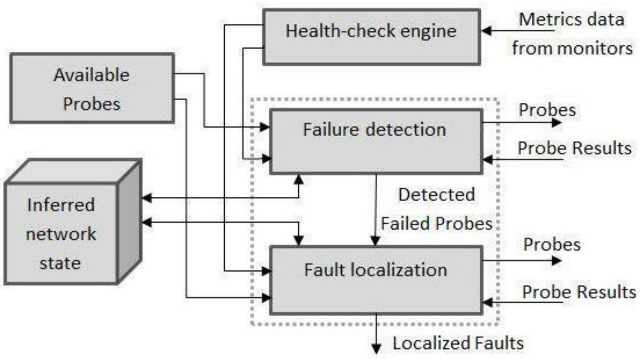

Figure 1: Architecture of Adaptive Probing

The architecture for the technique which we are introducing is given above. Basically in this method, we have two sets of probes which are used for checking the faults and their localization. Firstly we employ a small set for checking whether there is any fault or not. Here one set of probes is used for failure detection and its result is stored. Then another set of probes are used to localize the failure and then its result is also stored. Simultaneously, monitors are used on each node to have the performance check of each node. Monitors collects the system metrices and send it to the health check engine. Through all this process, the fault identification and its location is performed. Several design decisions need to be made at the health-check engine such as: (a) given a large number of metrics, how to choose the best representative set of metrics. (b) how to compute change in the steady state of a node? (c) how to compute abnormality of a node behavior?

3. Computing health measure of a node

This process is completed in three steps:-(1) Select the representative metrics, (2) Compute change in steady state of metrics, (3) Compute node abnormality.

4. Failure Detection

Probes are sent from probe stations. Probe stations are special nodes instrumented to send probes and receive probe results. Probes should be selected carefully because sending a large number of probes will create a huge burden on the already existing application traffic. The probes for failure detection are selected only to detect the presence of failure in the network and not to localize the cause of failure. The failure detection module uses the node health information to select probes and probe frequency for failure detection such that (a) any failure in the network can be detected, (b) probe traffic is minimized, and (c) failure detection time is minimized. A. Desired probing frequency for node and probe The desired probing frequency for a probe can be recommended based on the abnormality measure of its nodes. Thus, we propose to compute path abnormality as a function of abnormalities of its nodes.

Algorithm : Probe Selection for Fault Detection

Input: ProbeSet, Abnormality%, Abnormality to Freq. mapping of nodes.
Output: Fault Detection ProbeSet.

1. For each probe, calculate the following measures:
   - *Average Probing Frequency(APF)* = Average of the desired probing freq.(DPF) of all nodes in that probe.
   - *Additional Probing cost(APC)* = Sum of the additional probing costs of all nodes in that probe. It is APF – DPF if DPF<APF else 0.
   - *Monitoring loss(ML)* = Sum of the monitoring losses of all the nodes in the probe. It is DPF – APF if DPF>APF else 0.
   - *Coverage* = number of nodes in the probe which are not yet covered.
   - *ProbeScore* = α **APC* + β **ML* + γ **Coverage*

   where α, β, γ are constants whose typical values are -1, -2 and 3.
2. Include the probe which has maximum *ProbeScore* in the probeset.
3. Keep repeating Steps 1 and 2 till all nodes are covered.

Algorithm 1: Fault Detection Probe-set Selection

Figure 2: Alogorithm 1

Various functions can be used to compute path abnormality. A conservative approach is to choose maximum of the node abnormality values, while an aggressive approach is to choose minimum. For the following discussion, we consider path abnormality as an average of abnormality values of the nodes on path. Thus:

Abnormality(P)=Avg n {Nodes(P)}(Abnormality(n))

5. Fault localization

The fault localization component selects additional probes to be sent into the network to further analyze the nodes on the failed probe paths. Fault localization probes are invoked only in the event of presence of failure and should be designed to quickly and accurately localize the fault. The fault localization module uses the node health information to select probes such that (a) fault localization is accurate, and (b) localization time is minimized.

Algorithm : Probe Selection for Fault Localization

Input: ProbeSet, Abnormality%, Threshold, Suspected NodeSet(N)
Output: Fault Localization ProbeSet.

1. For each probe, compute the coverage as follows:
   Probe Coverage = $N_{healthy}$ -$N_{unhealthy}$
2. Cover Healthy Nodes.
   i. Select the probe that has largest *Probe Coverage*.
   ii. Keep repeating step 2.i till all healthy nodes are diagnosed.
3. Cover Unhealthy Nodes.
   i. Select the probe that has smallest *Probe Coverage*.
   ii. Keep repeating step 3.i till all unhealthy nodes are diagnosed.

Algorithm 2: Fault Localization Probe-set Selection

Figure 3: Alogorithm 2

Once a failure is detected by failure detection probes, additional probes are sent to localize the failed node(s). In this section, we demonstrate how probe selection for fault localization can be improved by using the monitoring information. We demonstrate this improvement using the example of Min and Max Search proposed as part of the adaptive probing solution in [3]. We refer to the nodes on the path of the failed failure detection probes as the suspected nodes. (a) Min-Search: Min Search selects a probe for each suspected node which passes through the least number of other suspected nodes. Here, a failed probe quickly localizes the failed node, but a successful probe does not significantly prune the suspected node set.

(b) Max-Search: Max Search selects probes that cover maximum number of suspected nodes. In this search, a failed probe needs additional probes to further localize the root-cause, but a successful probe significantly prunes the set of suspected nodes. We propose to use node health information to exploit the strengths of both Min and Max search. Probes for healthy and unhealthy nodes can be selected based on Max-Search and Min Search policy respectively. Thus, the probes selected using Max-Search policy are likely to succeed and hence will effectively prune the search space with minimal probes. The probes selected using Min-Search policy are likely to fail and will quickly localize the failed node(s).

6. Conclusion and future work

We presented an adaptive probing solution for fault localization in a network by adapting the probing policies using the information provided by monitoring agents. We presented initial ideas to infer information captured by monitoring agents, and used this information to select probes and their frequency for failure detection and fault localization. We presented the proof-of-concept through simulations. While the proposed solution demands deployment of both monitoring and probing agents, we argue that an adaptive approach can significantly decrease the probing and monitoring demands compared to cases of using either of them in isolation. As part of our going research, we are also working on intelligently selecting minimal nodes to deploy monitoring agents. We are also working on several other aspects of this solution such as (a) different ways to compute node health and path health, (b) automating the rule-book to map path-health to probe frequency, (c) extensive experimental evaluation of proposed solution in real set-ups and simulations.

References

[1] D.Jeswani, R.K.Ghosh and M.Monitoring: A Hybrid Approach for Monitoring HIPC 2010.

[2] S.-H.Han, M.-S.Kim, H.Ju, and J.Hong. The architecture of NG-MON: A passive network monitoring system for high-speed IP networks. In DSOM 2002, Montreal, Canada, 2002.

[3] M.Natu, A.Sethi, Application of Adaptive Probing for Fault Diagnosis in Computer Networks , In NOMS, Salvador, Brazil, 2008.

[4] I.Rish, M.Brodie, S.Ma, N.Odintsova, A.Beygelzimer, G. Grabarnik, and K. Hernandez. Adaptive diagnosis in distributed systems. IEEE Transactions on Neural Networks, 2005.

[5] P.Bahl , R.Chandra , A.Greenberg , S.Kandula , D. A. Maltz and M. Zhang "Towards highly reliable enterprise network services via inference of multi-level dependencies", Proc. ACM

Author Profile

Himanshu Sharma is currently pursuing his engineering in Electronics and Communication branch from Poornima Institute of Engineering and Technology, Jaipur and is in his final year(8th semester) right now.