Global Knowledge®

Expert Reference Series of White Papers

Ensuring the Availability of Your IBM Tivoli Monitoring Infrastructure

Ensuring the Availability of Your IBM Tivoli Monitoring Infrastructure

David Biondi, Global Knowledge Instructor, IBM Advanced Deployment Professional

Introduction

If you've deployed IBM Tivoli Monitoring (ITM), you are probably well aware of how great it is at finding out when things go wrong in an enterprise's IT environment. But when a component of ITM fails, it can leave your organization in the dark. Luckily, there are ways to prepare for and prevent ITM failures. Understanding the different components of ITM and how they interact can make all the difference in terms of keeping an ITM system up and running strong.

This paper will address four different components of the Tivoli monitoring infrastructure and contains some tips for ensuring that those crucial ITM components stay up and running. The four components are:

1. Tivoli Enterprise Monitoring Server (TEMS)
2. Tivoli Enterprise Portal Server (TEPS)
3. Remote Tivoli Enterprise Monitoring Server (RTEMS)
4. Tivoli Enterprise Monitoring Agent (TEMA)

Your strategy for ensuring the availability of each of these systems is based on the same principles: creating a high availability environment and providing redundancy and failover capacity. Although the principles are the same for each of these systems, the methods for adhering to these principles are different for each of them. This paper will address each of these four systems individually and provide examples of ways to ensure their availability.

TEMS—The Queen Bee of Tivoli Monitoring Infrastructure

Tivoli Enterprise Monitoring Server (TEMS) is the most important component of an effective Tivoli monitoring system. TEMS could be considered the check-in point for the entire Tivoli monitoring system: if it goes down, your monitoring system simply won't work. Because of this, measures have to be taken to ensure its availability. These measures can include monitoring as well as creating high availability by providing redundancy/fail over capacity.

How to Build Redundancy into the TEMS

There are two main ways to provide redundancy for the TEMS. One is to use a commercial clustering technology, such HACMP software, to failover components to a new system. In this scenario, there would need to be a secondary server configured identically to the primary server. All of the monitored systems would have to connect up to a virtual IP and this IP address would only be active only on the currently running monitoring system.

Creating redundancy by using clustering software has the advantage of providing a seamless transition to the failover system, thus allowing uninterrupted availability of the monitored system. When using clustering software, managed systems that are being monitored should be blissfully unaware of any problems: the transition to the

failover system (ideally) occurs so quickly and smoothly that it barely disrupts the functioning of the system. Although this redundancy solution sounds almost effortless, keep in mind that it requires the purchase of commercial clustering technology as well as implementation of that purchased solution.

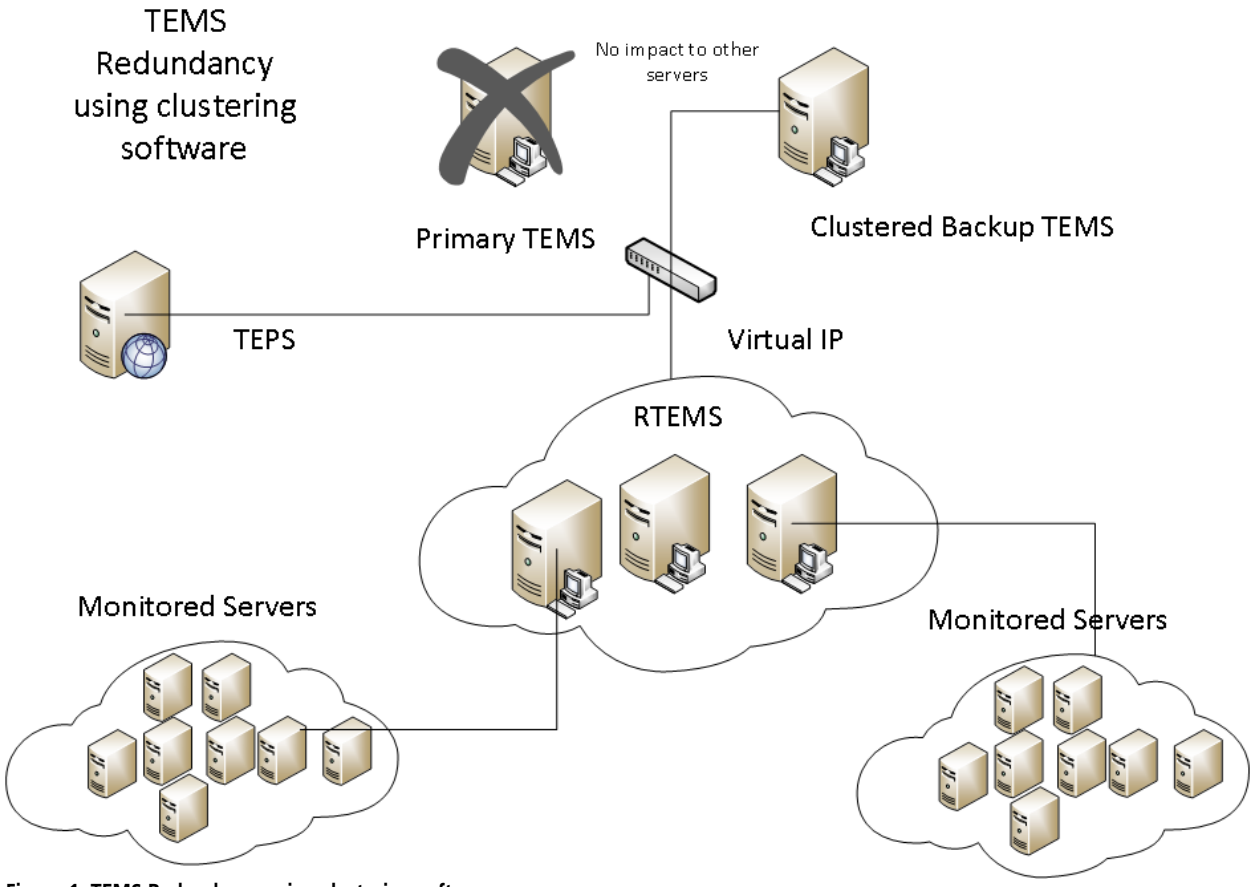


Figure 1: TEMS Redundancy using clustering software

In addition, for this solution to be most effective, it should have a dedicated standby server to failover to, thus adding additional expense to its implementation. This best practice recommendation can be circumvented by failing over to your TEPS or RTEMS systems if they have the capacity to accommodate the failed system. However, using this money-saving shortcut will reduce your system's overall performance during a failure.

A second method of providing redundancy for the TEMS is to use a hot-standby TEMS. Implementing this solution does not require the purchase of software, though it will require dedicated hardware. This solution relies on the use of internal ITM architecture to create failover capacity on a second TEMS server, which runs constantly (is "hot") and acts as a backup to the primary TEMS. The system will be configured to allow the RTEMS and TEMA components to know where the backup TEMS. That way, if the primary TEMS fails or systems are unable to connect to it, they will be able to report to the backup TEMS instead.

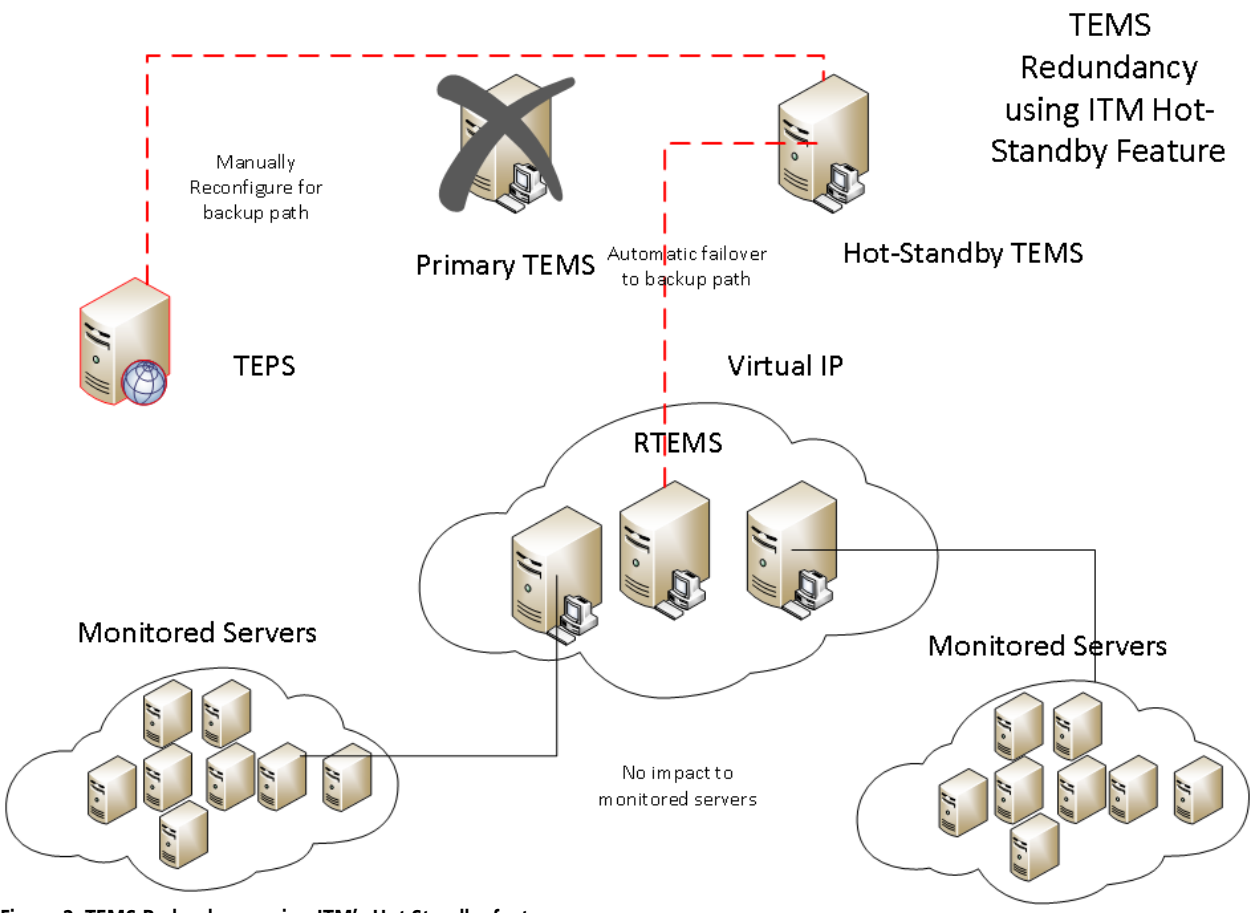


Figure 2: TEMS Redundancy using ITM's Hot-Standby feature

One of the major drawbacks of this solution is that it is not fully automated. When the primary TEMS returns to service, the RTEMS will continue to use the backup TEMS until they are manually switched back over to the primary TEMS. Also, the TEPS will require manual reconfiguring in order to start communicating with the primary TEMS again.

The TEPS—A Window into the ITM Hive

The Tivoli Enterprise Portal Server, often referred to as the TEPS, is the user interface into an enterprise's Tivoli monitoring environment. The TEPS could be described as a "window" into ITM: usually, it is the only way that administrators or operators can see what is going on in a monitoring system. The TEPS is the portal that they connect to in order to access monitoring systems. Through the TEPS, administrators and operators can see and influence activity in a monitoring system: the TEPS allows them to see any alerts that have been triggered and is also the portal through which they can configure the monitoring environment.

If the TEPS isn't running, there won't be a practical way for administrators and operators to see what's going on with the monitoring system or to configure any aspect of the system. Therefore, it's important for the TEPS to always be up and running.

Ensuring High Availability for the TEPS

If clustering technology has been implemented for an environment's TEMS failover, it can be easily integrated into the TEPS system of that environment. So, if an environment does have clustering software installed in the TEMS, it's likely that it will also be the best solution for providing high availability for the TEPS in that environment .

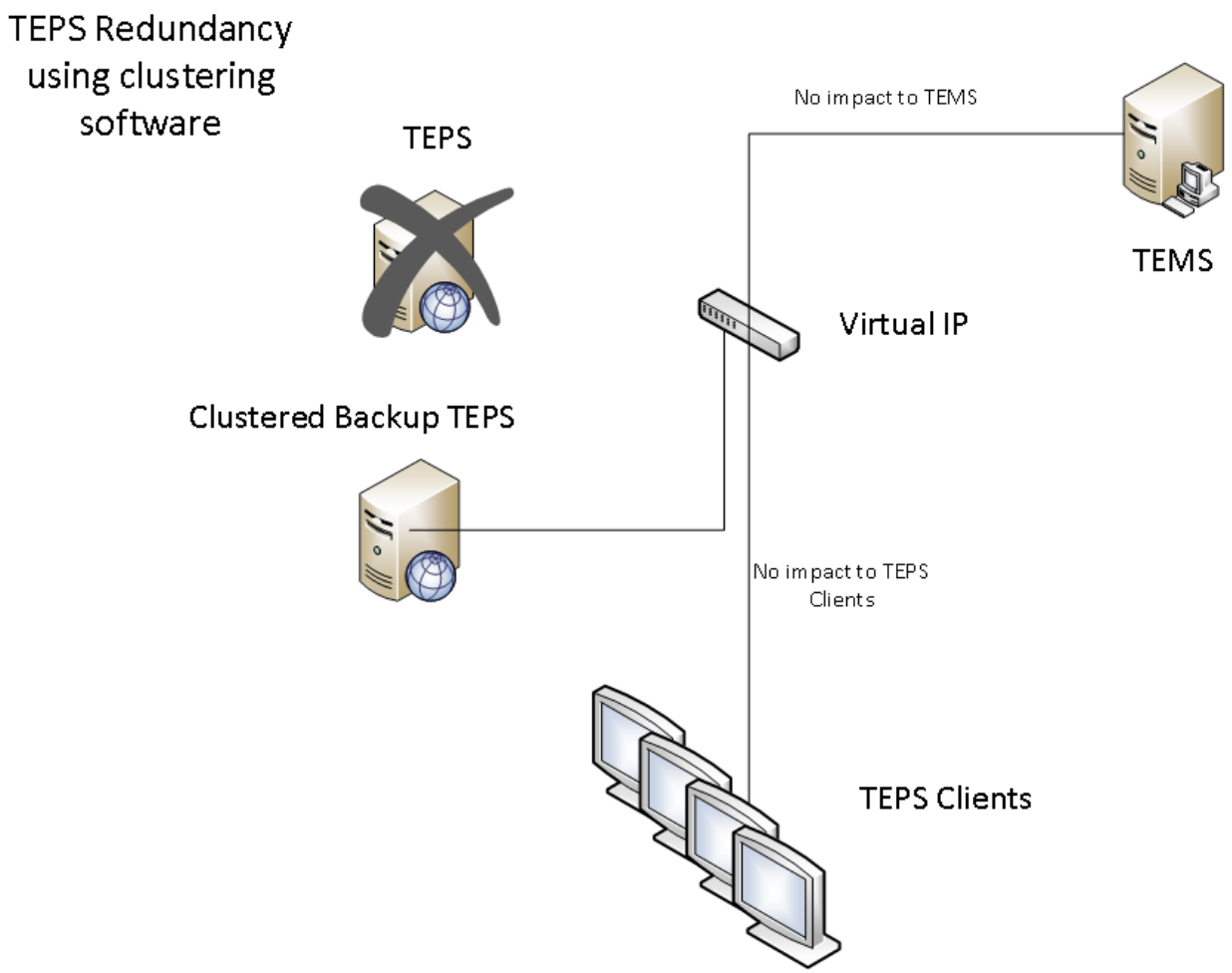


Figure 3: TEPS Redundancy using clustering software

There is no hot-standby capability for the TEPS built into ITM. However, standby capability can be built into a system by running multiple TEPS in an ITM environment. Deploying a multiple TEPS environment has the same benefits of a backup system in the event of a failure and has an added benefit of providing additional capacity to support increased numbers of portal users.

If multiple TEPS are going to run in a given ITM environment, it's important to consider whether the backup TEPS should be configured to communicate with the system's primary TEMS or with the hot-standby backup TEMS. Here's why: In the event of a TEMS failure that triggers a switch to the hot-standby, a TEPS that communicates with that system's primary TEMS will need to be manually repointed to the backup TEMS whenever it takes over for a non-functional primary TEMS.

If your backup TEPS is already configured to communicate with the backup TEMS, it will eliminate the need to manually reconfigure the TEPS when the TEMS goes down. If you aren't going to be using clustering software, having the backup TEPS configured to communicate with your backup TEMS is probably the way to go, because a TEMS failover is a really bad time to lose your TEPS. However, it is important to keep in mind that if your system is configured in that way, your TEPS will not have a backup unless you also have a dedicated backup TEPS system.

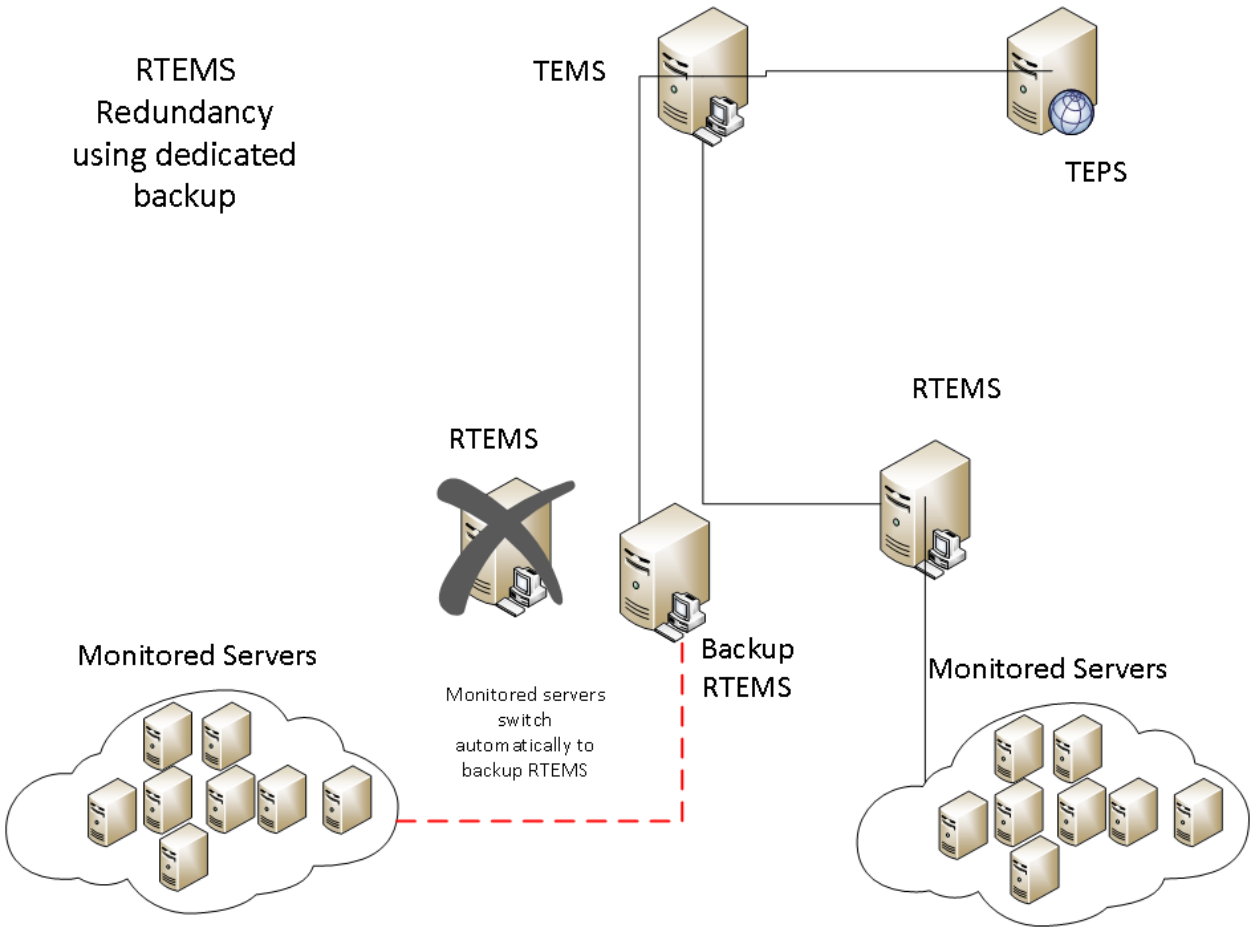


Figure 3: TEPS Redundancy using a backup TEPS server

Here's another point about installing your backup TEPS: If your hardware resources are limited, it is OK to install a backup TEPS on the same server as a hot-standby TEMS, even if the primary systems are split onto different machines. When implementing your backup TEPS solution in this manner, there will be a reduction in the performance of the backup systems compared to what the performance would be if the backup systems were installed on individual servers. As long as running at a reduced capacity is acceptable during an outage, having the backup TEMS and TEPS on the same server is a viable option for environments that need to implement a lower-cost solution.

RTEMS—The Messenger Bee

Think of the Remote TEMS (RTEMS) as a messenger that keeps the TEMS from being overloaded with chatter or low-level communications. The RTEMS is a second tier of infrastructure that serves as a proxy and communicates directly with a system's agents (TEMAs). The RTEMS thus allows a large number of monitoring agents to be supported without overloading the TEMS. The RTEMS chats with the ITM monitoring agents frequently. The communications occur both through a regularly scheduled heartbeat process as well as on an as-needed basis when a monitor triggers, a configuration change is made, or historical data is rolled up.

The RTEMS will sift through the information that it gleans during its communications with the TEMAs and will only relay certain information such as change of status, monitoring events, or batches of historical data to the TEMA.

Since the RTEMS acts primarily as a messenger, the failure of a single RTEMS does not impact the overall health of your environment. However, when an RTEMS fails, any monitoring agent that reports to that RTEMS will be disconnected from the monitoring infrastructure. To protect against the failure of an RTEMS, each of the monitoring agents in a system can be configured with the location of a backup RTEMS. When the system is configured in that way, a monitoring agent can simply connect to a backup RTEMS if they lose the connection to their primary RTEMS. Once connected to the backup RTEMS, the agent will continue to use that RTEMS until the agent is manually restarted.

The process of creating an environment with RTEMS redundancy consists of one straightforward step: simply provide the location of a backup RTEMS to your agents. However, there are some complex decisions to be made in terms of how to choose a backup RTEMS.

There are two ways to assign a backup RTEMS:

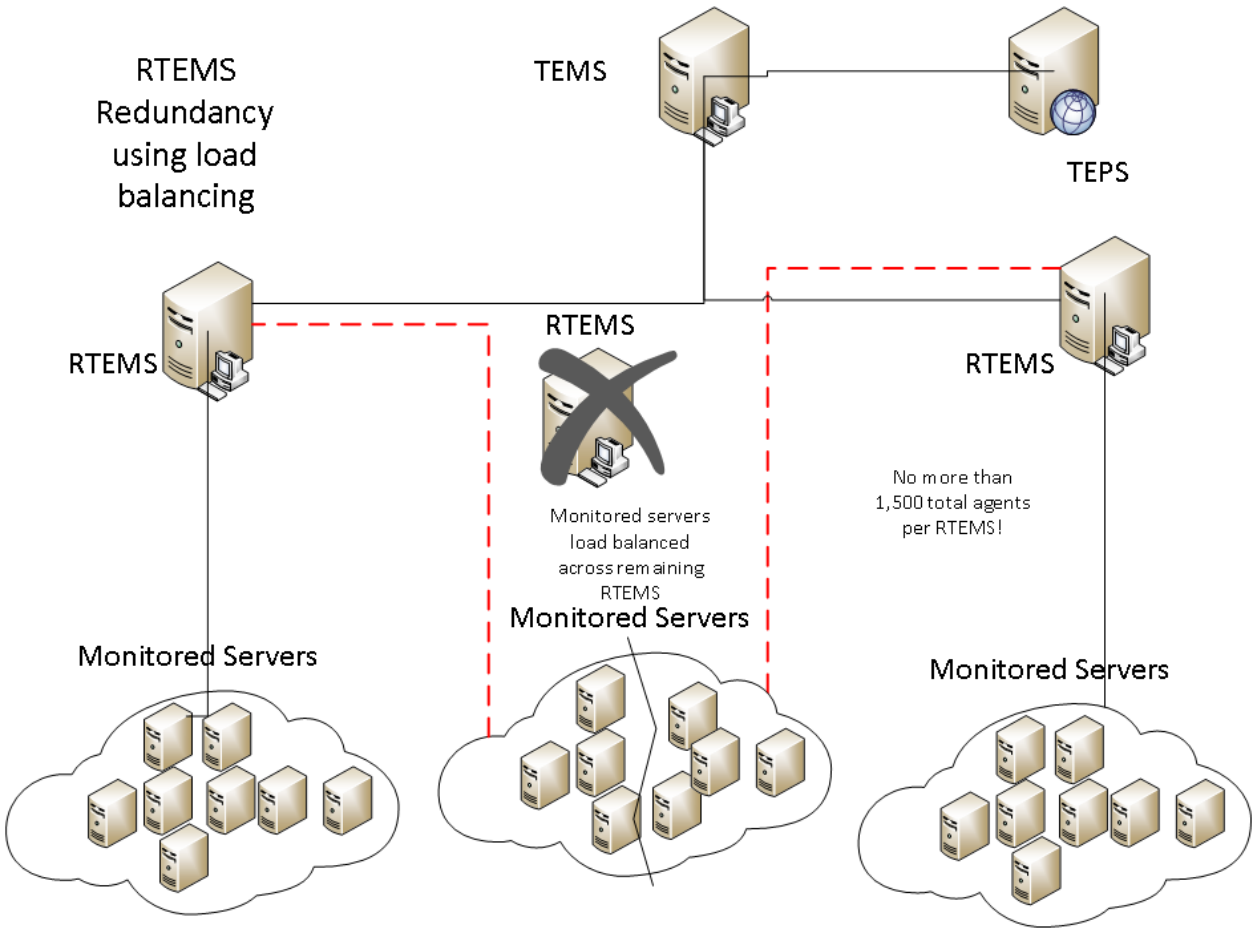
1. Deploy a dedicated backup RTEMS
2. Use current production RTEMS to back each other up.

It is highly preferable in terms of effectiveness and efficiency to design an environment with a dedicated RTEMS. However, doing so is initially more expensive than using production RTEMS as backups for each other and thus may get the thumbs down in an enterprise with budget constraints. If option number two is selected, and the environment will not have a dedicated RTEMS backup system, it is important to analyze the number of agents that are connected to each RTEMS. This analysis will be the first step toward designing an effective (though cumbersome) backup system using only the enterprise's current production machines.

Once the analysis is complete, you will have to decide how the agents will be distributed during an RTEMS failure. A single RTEMS should never connect to more than 1,500 agents. Thus, if you have more than 750 agents per RTEMS during normal operations, the load from a failed system will have to be split across multiple backup systems if an outage occurs.

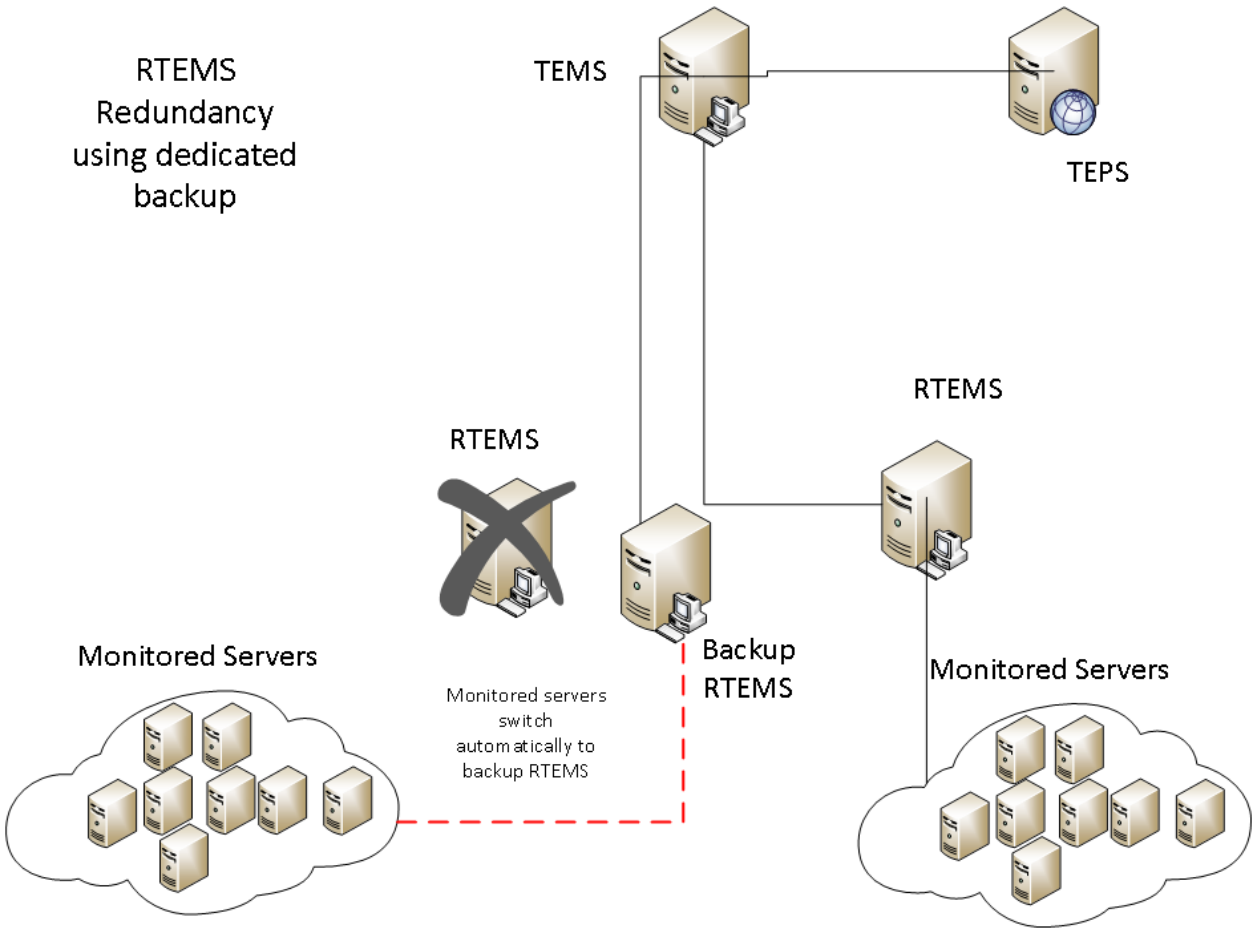
Here's a more specific look at this type of scenario, which outlines some common pitfalls that occur when there is no dedicated RTEMS backup system:

Suppose that there are four RTEMS with 900 agents each. When one of them fails, 300 agents should be sent to each of the three remaining systems. Here's the catch: this allocation must be pre-configured at each agent, so managing this configuration can become very complicated in a large environment. However, it is a process that should not be overlooked.



In an environment with no backup RTEMS system, it is absolutely critical that each agent be pre-configured to know the location of their backup RTEMS. Otherwise, a single RTEMS failure can lead to a cascade effect: if 900 agents fail over from RTEMS A to RTEMS B it can cause RTEMS B to fail due to overloading. Then the agents on RTEMS B could migrate to RTEMS C, and so on, causing an enormous system-wide outage.

A dedicated backup RTEMS can prevent this. The dedicated backup will ease the need for load distribution and eliminate the possibility of a cascade failure. In addition, a backup RTEMS will be a relative snap to configure because every agent in the environment will point to the same backup RTEMS.



Using a dedicated backup RTEMS also simplifies monitoring because it makes it easy to see which agents are not using their primary RTEMS. During normal operation, the backup RTEMS shouldn't have any agents, so if agents begin reporting to the backup, there must have been some communication failure between the agent and RTEMS. Or, after an outage is resolved, it becomes easy to identify which agents should be migrated back to their normal RTEMS. Every agent that is reporting to the backup RTEMS following an outage can be easily spotted and then simply be restarted and begin reporting back to their primary RTEMS.

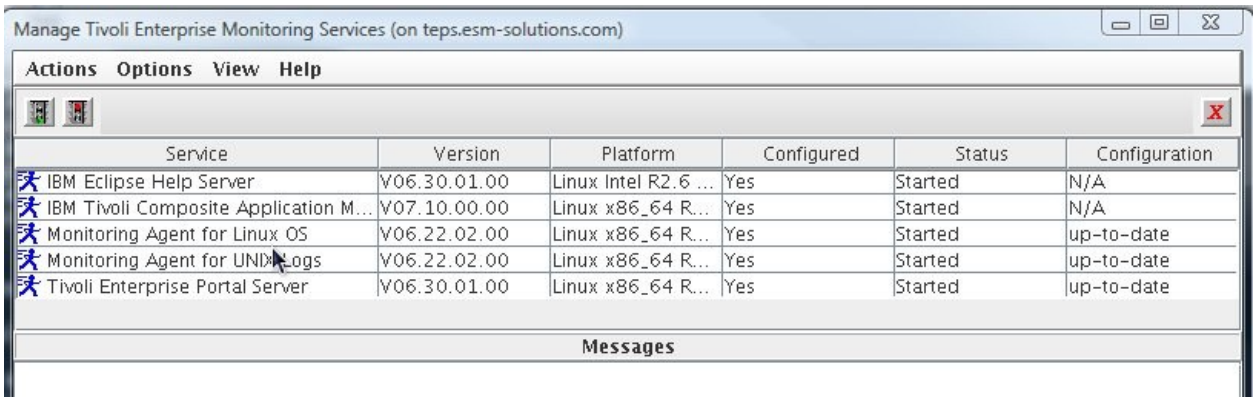
TEMA—The Worker Bee

TEMA is what actually runs the monitors on the end system. Some would argue that the TEMA is more important than the TEMS. The logic behind this, going with the bee analogy, is that the TEMAs are the worker bees that serve the TEMS ("queen bee"), so the TEMS would be useless without its hardworking TEMAs

.

One thing about the TEMAs is that you are likely to have multiple different TEMAs on any individual server. This is because each TEMA monitors one individual aspect of the system. For instance, one TEMA on a server could be monitoring the operating system while another monitors the DB2 database, and a third could be monitoring WebSphere or a web server. It wouldn't be unusual to have a machine that is running half a dozen or more TEMAs that are unique and are not redundant to each other.

You can tell which different agents are installed on a server by viewing the ITM management interface. This dialog will indicate all of the ITM components that are installed and the status of each of those components.



In addition, each TEMA on a server will have a unique product code. Every product code corresponds to the TEMA's function, so product codes can be used to determine which job each TEMA on a machine is supposed to be doing. Each product code is two characters and represents a specific type of TEMA agent—e.g., ux is the Unix Operating System Agent. You can see a list of current product codes here: http://www.esm-solutions.com/software/tivoli/itm-ibm-tivoli-monitoring/ibm-tivoli-monitoring-product-codes.html

Generally, it is not necessary to provide redundancy for the TEMA. If the TEMA fails, it should not impact the delivery of services to your customers. However, if your TEMA is monitoring an application that uses clustering and failover, the TEMA should be configured to be managed by the clustering software in the same fashion as the application. For example, if your WebSphere MQ service fails over from server A to server B, the monitoring agent should be stopped on server A and started up on server B as part of the failover sequence.

You can ensure availability of your TEMAs, however. Each of your monitored servers should be running an operating system agent that monitors your basic system resources, such as disk space, CPU utilization, etc. This operating system can be configured to also monitor the other application agents on that server and attempt to automatically restart those agents in the event that the agent fails.

Conclusion

Although ITM is not a foolproof system, understanding the different components of ITM and how they interact can go a long way toward keeping an ITM system up and running. Hopefully this paper has presented some ideas that will help you protect your ITM systems from outages. Happy monitoring!

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

- IBM Tivoli Monitoring 6.2 for Implementers (TOS08G)
- IBM Tivoli Monitoring 6.2 Historical Reporting and Data Collection (TV372G)
- IBM Tivoli Monitoring 6.2.1 Agent Builder (TV382G)
- IBM Tivoli Monitoring 6.2.3 Advanced Administration (TM062G)
- IBM Tivoli Monitoring 6.2.3 for Implementers (TV354G)
- IBM Tivoli Monitoring 6.2.3 Fundamentals (TM022G)
- IBM Tivoli Monitoring 6.3 Advanced Administration (TM063G)
- IBM Tivoli Monitoring 6.3 for Implementers (TV355G)
- IBM Tivoli Monitoring 6.3 Fundamentals (TM023G)

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

David Biondi is the founder and CEO of ESM Solutions, an IT consulting company that is recognized as an IBM Advanced Business Partner and is accredited across multiple IBM product groups. In addition to working as an IT consultant, David also takes pride in his work as an instructor and course director for Global Knowledge.

David has a strong commitment to broadening his knowledge base and keeping his skills cutting edge. He is ITIL Foundations certified and maintains current certifications on many IBM products including Tivoli and WebSphere. David is also pursuing a master's degree in computer science from Georgia Tech where he is focusing on artificial intelligence and machine learning.