



NTP Software White Paper

Agent-based vs. Agentless Management

How to get the system management information you need, cost-effectively, and without making the problem worse along the way

Abstract

The need for network and systems management is clear. The challenge is to do it cost-effectively and in a way that doesn't have a negative impact on the performance of the systems you are trying to manage. One of the critical questions in this regard is whether to use an agent on each managed device or deal with them remotely via agentless management. This white paper explores that question with the goal of helping you understand the pros and cons of each approach.

Rev. 1.2, July 2001

Introduction

The need for applications and systems management almost goes without saying. But collecting the data to proactively manage your network is a lot of work. To do it cost-effectively, the challenge is to gather that information and act on it in a way that makes the situation better, not worse. Spending 10% of the network's resources on management is probably not a winning scenario.

There are really only two ways to watch a system or an application. You can do it locally, right on the machine you are trying to manage (agent-based), or you can do it remotely (agentless), by probing the managed system to see if it is alive and asking it for data from time to time. These are fundamentally different approaches. Which one is better? Which one should *you* choose?

Agent-based and agentless management are profoundly different as technologies and architectures, and in their costs and benefits. The right choice for a given situation comes from understanding these differences, the situation in question, and the technology and resource constraints. The purpose of this paper is to give you a basis for this understanding.

What Do The Terms Agent-based And Agentless Mean?

Agent-based management

Agent-based management always involves software running locally on the managed machine as part of the management infrastructure and solely devoted to the task of management. There are not-so-smart agents and there are smart agents.

A not-so-smart agent runs on the local machine and only collects management information that is subsequently offered up to some other system. There is certainly value in this. The data may not be remotely available without this agent, and/or the collection agent may package the data and transmit it more efficiently than what might otherwise be possible. But, bottom line, a not-so-smart agent does little more than gather data.

A smart agent, on the other hand, not only gathers data, it can also inspect the information at hand and take action. For example, a smart agent can discard data points that are not meaningful. It can trigger an alert based on local analysis of values and ranges of data, etc.

Agentless management

Agentless management is characterized by the absence of any management software or process on the target machine. This doesn't mean that the target cannot supply management data, rather that some other system needs to poll for that information and no action can be taken locally.

From a systems management perspective, everything that happens is done remotely.

The Pros and Cons of Agent-based Management

On the face of it, it might appear that agent-based management is a lot of extra work. After all, it means we have to put management code on every machine and then maintain that configuration over time. It's true; this is work. But it is often worth it. Let's look at the details:

Pros

- **Data is collected locally** – no network traffic for basic data gathering
- **Uninteresting data can be discarded** – reduced network load for whatever might be transmitted for summary analysis
- **The local agent continues to act even when the network is down** – systems management is not suspended by a network failure, false negatives are eliminated
- **Agents can be tightly integrated** – a local agent can more easily effect an automatic corrective action
- **Few or no dependencies** – since management is local, the status of other devices has little or no effect on the target machine

Cons

- **Software needs to be installed on every machine** – increased deployment effort and possible negative interaction effects with other software on the machine
- **A smart local agent uses system resources** – the local agent is an application on the managed machine like any other. It will use some system resources

The Pros and Cons of Agentless Management

While agent-based management has more pros than cons, you can't always deploy an agent. There are plenty of network devices that won't accommodate the installation of 3rd party agent technology. You might even want to monitor systems on which you don't have the rights to install an agent. In these cases, agentless is your only choice.

Pros

- **No agent to deploy** – simpler deployment (not necessarily less costly)
- **Lightweight on the managed system** – in most cases there is less drain on the resources of the managed machine

Cons

- **Increased network load** – constant polling; and the closer the tolerance, the more polling
- **False negatives or lost view** – a disruption of the network has the same result as the target machine being down, no data
- **Dependency chains, critical points of failure** – as installed, agentless management has one machine watching many devices. This makes that one machine critical to everything. In addition, there are often choke points

(routers, for example) between that machine and communities of managed systems

- **Corrective actions are hard to implement** – since we don't have a local context in which to act, anything we might want to do to automatically remedy a fault has to be by remote execution. This creates some security risk and adds additional failure scenarios to our remedial script

Agent-based and Agentless Management in The Real World

In the real world, there are no free lunches and no simple problems worth solving. Most networks have a combination of devices and applications that vary in their manageability. Routers, switches, and the like are black boxes. Generally, you can't put 3rd party code on them even if you want to. This means that if you want to include them as part of your systems management architecture you will have to use agentless management to do it.

On the other hand, you can't cost effectively manage a complex network and its applications without agents. There would be too much network traffic, and too many situations where the polling station loses contact with (and thereby control of) a critical system resource.

Remember, the payoff from this technology isn't to automatically tell you that you have a problem. Your users will do that faster than any computer program can figure it out. The value of this technology is that it will automatically take corrective actions to repair a fault, and then tell you at your leisure what happened and what was done to cope with it. These corrective actions are just too hard to write and too error-prone if your management approach is solely agentless.

A real world systems management architecture must include both agentless and agent-based management to be effective. For critical systems and applications, you need a smart local agent that can act even when the network is down, and can easily be configured to take corrective actions. For black box systems and other situations where agents are not cost effective, you need agentless technology. Only the combination gets the job done well.

Conclusions

- Both agentless and agent-based management have their place, together in most systems management architectures
- For the most part, only smart agents make sense. There is generally not enough value in using agents solely for data collection (you pay all the costs with few of the benefits), although this may be required in some situations
- The ease with which you can integrate the agent and configure corrective actions is a key component of a cost-effective solution. Many system management implementations fail over this point alone