



# 1. Introduction

This booklet is about one aspect of system and network monitoring: collecting, storing, reviewing, and acting on numbers (or, more formally, data points) that tell us something relevant about our environment. We are concentrating on instrumenting our systems and networks rather than on examining or monitoring the services we provide. We focus on monitoring our environment and less on controlling the equipment that controls it. But with any monitoring process comes the ability to influence the thing being monitored, either automatically or through conscious action on the part of the person(s) doing the monitoring. As Lord Kelvin<sup>1</sup> said, “If you cannot measure it, you cannot improve it,” and we seek to improve our processes by monitoring what is going on in them.

## 1.1 What Is Monitoring?

When we talk about monitoring, we often define it as “periodic sampling of system, device, application, environmental, or network status information, recorded and/or dispatched for analysis and/or correction.” Or, more succinctly, “information about systems and networks that is collected, analyzed, and acted upon.”<sup>2</sup>

System, network, and environmental monitoring is an important part of system and network administration because it:

- ❖ Contributes to reliability and availability.
- ❖ Helps get problems identified (and fixed) faster.
- ❖ Contributes to a better overall quality of service.
- ❖ Contributes to a more enjoyable and effective working life for system administrators.

Limoncelli/Hogan/Chalup [Limoncelli07] may have said it best when they wrote, “A service is not complete and cannot properly be called a service unless it is being monitored.” If there is no monitoring then you’re just running software.

## 1.2 Why Monitor?

We’ll claim that monitoring is done for three primary purposes:

### Exceptions and Anomalies

- ❖ To identify and report unusual conditions, failures, possible problems that may need attention.

1. [http://en.wikipedia.org/wiki/William\\_Thomson,\\_1st\\_Baron\\_Kelvin](http://en.wikipedia.org/wiki/William_Thomson,_1st_Baron_Kelvin).

2. Which also includes inaction: a decision not to act is still a decision.

## 2 / Introduction

Examples: Is the Web server serving up Web pages? Is the disk full? Is the data center on fire?

### Trends

- ❖ To collect numeric data that characterizes some aspect of a system or network over time.

Examples: Network bandwidth utilization, CPU utilization, Web site hit counts, number of bytes used on a file system, the temperature in the data center.<sup>3</sup>

### History

- ❖ For historical analysis of time-stamped data, primarily for record-keeping and trouble-shooting.

Examples: Billing, tracking who was doing what and when, or responding to discreet (or indiscreet) inquiries from law-enforcement officials.

In this booklet we will concentrate primarily on data collection and monitoring for “trends,” with some identification of exceptions. We are primarily looking at the collection of information that can be quantified numerically, and how we can visualize and analyze the data we have collected over time. Some of the numbers we collect may be simple binary state information (e.g., whether a door is open or closed, a light is on or off, there is water on the floor or not), but in general we’re not going to be looking at methods for checking the state of services (e.g., whether the mail server’s SMTP port is communicating properly, or whether the network connection is up or down). For trend monitoring, we’re more interested in the *volume* of traffic for a port or service, but if you have a ready means of determining application state<sup>4</sup> or port status, that state information can often be supplied as a binary input to the monitoring software we are describing (see section 2.4.9). If the numbers we collect have values that are too high or too low, we may have an indication of an “exceptional” condition that we will want to report and/or act on.

Unfortunately, monitoring is not a task that has the benefit of immediate time savings. Like most tasks, setting up a robust monitoring system will take time and will likely require that you learn how to use a few different tools. Monitoring may also never save you time on the back end either—if you adhere to a rigorous maintenance schedule and have perfect firewalls, all your monitoring will ever show you is that things are ok. But the *potential* savings from a monitoring system, in terms of both time and money, are *huge*. When something does go wrong, your monitoring system will help you pinpoint the problem quickly, perhaps avoiding a cascade of failures.

3. Which might help you predict that the data center is about to catch fire, or in retrospect, to know when the fire that raised an exception actually started.

4. The actual maintenance of system and application state is more under the aegis of tools like Cfengine [Burgess07], LCFG [Anderson08], or BCFG [Desai08].

## 1.3 Service Level Agreements

For some people, the primary reason to monitor is to verify that they are conforming to a service level agreement (SLA) or to help them correct problems before they are in violation of an SLA. Their monitoring systems collect hard data and alert them when they are not in compliance.

From a formal perspective, this makes sense: the needs of the business define what services are required and what quality (or level) of service is needed (the SLA—how good is “good enough”), and then systems and tools are designed and implemented to meet those needs and conform to the SLA.

One problem with this focus on monitoring for SLA compliance is that most sites don't have well-defined SLAs for all services. You might have an SLA that says that “if a file server dies, sysadmins will be notified within five minutes and the ‘on-call person’ will be able to start working on it within 15 minutes” (though a better SLA would be that “the file server will be up 99.99% of the time,” and let the sysadmins responsible for the file server create a monitoring and alerting policy that achieves that). But who has an SLA for every service or environmental component? And, really, who wants to write them all?

Another problem is that this approach takes some of the ambition out of system and network administration: instead of aiming for perfection, it aims only for “good enough.” We prefer to aim high, and use after-the-fact reporting and comparison with SLAs as a yardstick that we've achieved more than “good enough.”

We like the abstract notion that SLAs and monitoring are simply a means to verify compliance, but unlike more formal, dry books, we're not going to focus on SLAs. We'll concentrate instead on enabling the satisfaction of a job well done, and the thrill of discovery—the “ah hah!” moment when you realize that there really is a reason you've been looking at all those data, charts, and graphs. Dan wouldn't have discovered (and fixed) the problems described in chapter 6 if he hadn't started his monitoring system just for the fun of it.

## 1.4 What Types of Data Can We Collect?

The short answer is “anything we want,” but for the purposes of this booklet we are primarily interested in data that can be expressed numerically and that can be used to characterize or quantify the state of a device, service, or environmental condition.

Obvious examples are counters or gauges measuring how many mail messages have been processed or how full a disk is. But we can also keep track of less obvious data, such as how long it takes to retrieve a Web page or how quickly a mail message can be processed and dropped in a mailbox.

If you look around your environment and apply a little creativity, you can probably come up with an almost unlimited set of data you can collect and analyze. HVAC system activity, the number of telephone calls through your PBX, the load on your power bars or UPSes, the temperature and humidity in the server room, the number of pages printed on the expensive printer: these are just a few examples, and we haven't even

considered computers or network equipment yet. We discuss this topic in more depth in section 2.1.

### 1.5 Contents of This Booklet

The remainder of this booklet is divided into seven chapters. Chapter 2 looks at the data you can collect and how you can collect it. Chapter 3 covers how you can store your data. In chapter 4 we have a look at visualizing your data using charts and graphs. We cover problem detection and notifications in chapter 5. Analysis and examples of real-world problem detection and resolution are covered in chapter 6. Chapter 7 provides pointers to software packages, hardware vendors, and other sources. And chapter 8 offers some additional advice, a summary of what we've covered, and where to go from here.

You'll notice that this booklet tends to shy away from quick and easy answers and instead presents (sometimes conflicting) alternatives. You may walk away from your initial simple question with even more questions, but we'll help you sort out the wheat from the chaff and start you down the right path to develop a monitoring solution that satisfies *your* needs and solves *your* problems.

You'll also notice that John and Dan don't always agree—and that turns out to be a good thing, because not only do you get to see divergent viewpoints, you get to see why we disagree. That will, hopefully, help you decide which of us you want to listen to, and know why your decision will be right for your problem space.