Performance is one of the main issues that every administrator faces at one point or another. This book will help you to identify and mitigate it.

The book starts with an introduction to Zabbix and the most common mistakes that are usually overlooked while deploying it. You will discover the importance of an impressive and easy-to-use front-end and how to develop one, along with comparisons between Apache, Nginx, and lighttpd web servers. Storage, as an important aspect of every deployment, is covered and you will be shown how to choose the right storage system for you and configure it to suit your deployment.

Towards the end of the book, you will walk through the intricacies of the operating system, the final piece of the puzzle that holds all the components together. You will then dive into advanced topics including using a specific piece of hardware for each of the components, the Zabbix proxy, and firewall settings. By the end of the book, you will be able to tune your Zabbix implementation to achieve maximum performance.

Who this book is written for

This book is aimed at Zabbix administrators who want to learn how to improve the performance of the Zabbix system. Basic knowledge of Zabbix’s components and features is assumed.

What you will learn from this book

- Identify Zabbix’s numerous ways of tuning and find the best match for your environment
- Familiarize yourself with tools including web servers, database servers, storage, and the support system of Zabbix
- Discover the most common performance configuration errors made by administrators
- Explore the importance of the Zabbix proxy in your deployment
- Learn why keeping the Zabbix SIA updated is key to dealing with performance issues
- Master the art of efficiently troubleshooting Zabbix’s performance

In this package, you will find:

- The author biography
- A preview chapter from the book, Chapter 1 'Evolution of Zabbix'
- A synopsis of the book’s content
- More information on Zabbix Performance Tuning
Luciano Alves is Brazilian. He lives in the city of Porto Alegre in southern Brazil. He has been a Zabbix evangelist since 2007. Since 1994, he has been working to support the ICT infrastructure using his expertise in targeting and monitoring platforms. Luciano has had Zabbix professional certifications since 2008, and in 2012, he became one of the first official instructors outside of Zabbix SIA. He has trained over 200 professionals from different countries (Brazil, Argentina, Uruguay, Ecuador, Chile, Spain, and Colombia) to work with the Zabbix monitoring platform.

He is the founder of Unirede Soluções Corporativas, one of the main global partners of Zabbix SIA and currently the only premium partner in all of the Americas. Luciano is also a speaker at open source events, always talking about Zabbix. Besides this, he is a moderator of the official Zabbix forums in Portuguese and Spanish, and has worked on the translation of the official documentation into Portuguese.
Once, I read a book called *The Outliers* by Malcolm Gladwell. In this book, the author talks about the circumstances that lead people to get certain results in their areas of expertise, whatever they may be—music, sports, math, physics, and so on. The key aim of the author is to assert, based on facts and data, that there are people with extraordinary qualities, but the result depends on the effort, time, and energy devoted to that activity. The opportunity to stand out comes when we are in the right place at the right time. There is a popular saying, "Luck is what happens when capacity/competency and opportunity meet." I think Alexei Vladishev was lucky.

When I met Alexei in 2008, I was surprised by his simplicity and mild nature at the places where he worked, whether it was a training session, a lecture in an auditorium, a meeting with the client, or anything else. He's one of those guys who have in themselves a great guiding force for their actions and decisions, and who don't accept the label of a "pop star" in the open source world. In one of our conversations, he told me about one of the reasons for devoting his professional life to Zabbix SIA and his time to the development of Zabbix—improving people's lives. Simple, and without any "mega" aspiration! He only wanted to help people get their work done better. From this originate his other decisions regarding Zabbix; for example, there is no paid professional or enterprise version.

When I was invited to write this book, I thought about the points I have just described. I thought that it would be a great opportunity to bring to a greater number of people the knowledge that I had gained over the years working with Zabbix in large companies. It would also be a way to give to this field something in return for what I had obtained from it over the years. I hope this book will help you take full advantage of Zabbix, and may it contribute to the goal Alexei had—that of Zabbix being a tool for improving people's lives.
What this book covers

Chapter 1, Evolution of Zabbix, explains that the start with Zabbix is not always glamorous as it does not always have the most advanced features. If you're experiencing performance issues with Zabbix, it is likely that you can solve them without new hardware or software.

Chapter 2, Zabbix and I – Almost Heroes, talks about how Zabbix has evolved due to performance issues in each version. We need to look out for the new features. In this chapter, we also talk about the importance of Zabbix, which grows as the other teams and areas of companies become aware of the potential of this tool.

Chapter 3, Tuning the Zabbix Server, tells you a little more about the possibilities and needs for adjustments to the Zabbix server. It is important to realize that many performance problems can be solved by adjusting the Zabbix server's settings.

Chapter 4, Tuning the MySQL Database, talks about the motivations behind choosing MySQL as the database for Zabbix, and the main MySQL settings related to the performance of Zabbix. In this chapter, you understand a few things about MySQL's behavior with Zabbix.

Chapter 5, Tuning the Frontend, teaches you how to speed up the delivery of Zabbix data to users and improve their experience with the frontend. Nowadays, we can choose from new web servers. Apache is not the sole alternative, nor is it the "killer one."

Chapter 6, Adjusting the Storage, mentions what we must take into account when we think of the hardware, especially the storage to use with Zabbix. To get the best performance from Zabbix, we need to understand how information flows into it and adjust the settings to control the flow of data (reading and writing).

Chapter 7, Tuning the Operating System, considers the fact that the operating system is an important part in this control flow process. We should make sure that we search in our Linux distribution to find the best practices for performance, always taking into account the component that the operating system is involved with (the Zabbix proxy, Zabbix GUI, Zabbix server, or Zabbix database).

Chapter 8, Doing the Extra Work, informs you that it is possible to target the components of Zabbix (the Zabbix server, GUI, and database) on different pieces of hardware. It will also become clear to you that the great hardware consumer is the Zabbix database, and it is this component that must be given the best hardware.
Chapter 9, Using the Zabbix Proxy, proves that the Zabbix proxy can be a powerful ally of the Zabbix server for improving environmental performance. It is good practice to start the environment by already considering a Zabbix proxy, even if it is on the same network in which we have the Zabbix server.

Chapter 10, Monitoring the Health of Zabbix, asserts that Zabbix's internal metrics are an important part of performance tuning. These metrics can't be evaluated alone. We need to understand that they could be affected by other factors, such as the Zabbix database settings.

Chapter 11, The Next Challenge, puts forward a challenge: understanding where and how Zabbix can support the company's business. We should not think of this powerful tool as only something to be used to monitor the IT infrastructure.

The online chapter, Performance Features and Improvements since Zabbix 1.8, provides some information about the new features and improvements related to performance since Zabbix 1.8. The chapter is only available online at https://www.packtpub.com/sites/default/files/downloads/B03410_Appendix.pdf.
If you are working with Zabbix or any other monitoring tool, you may be knowing a little about downtime costs. Downtime affects companies, products, and even the service's reputation. There is a lot of research that tells us that companies around the world lose a large amount of money because of system outages.

Probably, your company has services or products that depend on IT infrastructure. But the question is: what exactly do you know about these dependencies? Probably a little, but this isn't something that only you missed. Our experience shows that a lot of companies don't know about it.

In this chapter, we will try to explain how Zabbix works in most environments, and the common mistakes we tend to make. We will cover the following topics:

- Starting our journey
- Choosing the right tool
- The first wrong step with Zabbix
- A little about my first steps
- Good practices
- Simplifying Zabbix
- Challenges in Zabbix
- List of don'ts
- The beginning of the real challenge
Starting our journey

Our journey starts when, say, you arrive at work on a sunny Monday (maybe a cloudy day), and your boss is waiting for you in the parking lot. The systems had an outage last night and you need a solution to predict this situation. Now what? Which way to run? Which tool to use? The first step is to try the Internet search engines. Some references will appear—some old, some not so well-known, and so on. But which one should you use? And how to choose the correct tool among so many of them? The specifications and presentations of each one are quite interesting. Apparently, some of them fit into your needs, but which one to use? There are pieces of software, tools, and platforms for all tastes, flavors, and budget sizes. What is your budget? Of course, the lowest possible! Do you have experience with open source? Do you know that the main tools of this type are already professionalized and there are companies that develop and provide support services for these tools? So why not follow this line (gain flexibility and achieve a deployment without exorbitant costs) and work with an open source tool to monitor the environment?

Choosing the right tool

Let's discuss a common scenario. You create a matrix adhesion that lists the main features and defines weights for each. So, beginning the prerequisites and testing the concepts does not take long, and you will realize that Zabbix stands out among all comparative tools. The features are very interesting; Zabbix proves flexible enough to meet all the demands of environmental monitoring. It has an API that allows integration with other systems and applications. You can extract reports regarding recurrence alerts on servers and network assets. It is also possible to have monitoring based on historical data and real-time monitoring. Zabbix works with website monitoring, JAVA, IPMI, SNMP, ODBC, and more. Using Zabbix, it is possible to create rules for servers and other devices to start monitoring without human intervention, that is, automatically. Everything seems to fit the company's needs, and there is a large and active users' community helping and supporting this tool. The tool also has a distributed monitoring model that uses proxies (Zabbix proxy) to ensure data collection even if the monitored environment has no communication with the Zabbix server. Another point to be stressed on is the Zabbix GUI, which is pretty rich and full of possibilities. The developer (Zabbix SIA) has a partnership program supported in many countries and uses the local languages. Why wait more? Some users might say that all the features mentioned so far are present in other tools as well, and in some aspects, they may be even better than Zabbix. So what is the advantage in choosing this tool over others?

From my point of view, Zabbix was born with a very advanced concept compared to other players at that time. I remember my compliance matrix perfectly; there were very important points and impacts on our business model that only Zabbix met.
An example is that the distributed monitoring at that time (version 1.4 of Zabbix) did not have the concept of the Zabbix proxy, but the tool already had the concept of distributed monitoring based on nodes (a functionality that was removed in Zabbix 2.4), and this was something new at that time. Another impact on our business was the ability to segment the environment into a user group and a group of hosts with specific permissions for every requirement (read-only, read-write, and so on).

Here's yet another example—the centralized model that Zabbix always had. In this model, the Zabbix agent is a mere collector, which may or may not gain any intelligence. The alert rules and collection settings are managed and controlled by the Zabbix server, thereby avoiding the need to access the agents when the need for adjusting the collection and alert settings arrives.

Let's get back to the most important items in our business way back in 2007. It's been a few years of learning and development of Zabbix SIA, with the hope of it becoming the best open source monitoring tool. In this sentence lies another great argument for Zabbix: the tool is true open source (in the words of Alexei Vladishev). In various projects in which we participated over the years, it became clear that Zabbix does not leave much to be done by the main commercial monitoring tools, along with the advantage of being open source.

The first wrong step with Zabbix

Another common scenario occurs when the environment created for development, testing, and approval of Zabbix is converted into a production environment.

If the first scenario looked familiar, or at least plausible, this will not be hard to imagine. It is likely that the manager was not waiting for us with a mission and with a big check to be used in the acquisition of hardware and services for the monitoring service. The fact is that this project will often start without many features and without a lot of trust from the rest of the company. In the most common scenario, we begin a test in an environment with restricted servers, and one of the most common mistakes is turning this environment into something that should support all systems and enterprise environments.

Often, Zabbix replaces another monitoring platform that is already in use. In this case, the birth of Zabbix in your environment is more controlled and planned, as there is something existing that you need to replace. But still, another scenario repeats; that is, it uses the same concepts and ideas of the old tool in the implementation of Zabbix. This is another interesting point because users often compare both the tools and lack understanding about Zabbix's features and concepts that ensure correct use of the platform.
These situations can turn into a trap and contribute negatively to the performance of Zabbix.

**Getting started with Zabbix**

My studies in this wonderful monitoring platform, Zabbix, began in 2005, with the first version of the software. Actually, my experience in monitoring systems dates back to 1994. Since then, I’ve encountered several tools, commercial and open source, for monitoring IT environments. I admit that open source tools have always been my favorite because of the flexibility that they offer and also because of the developers’ ideology. These two points have always guided me in finding and using IT solutions.

When I came across Zabbix in 2005, the platform had not grown enough to be used in our production environments. It started growing from then on, and apparently had good potential. In 2007, with Zabbix version 1.4, we understood that the platform had evolved enough to meet our and our customers’ environments. Then we encountered one of the scenarios mentioned in the previous section (using a tool with the concepts of another tool), and faced difficulties in this scenario. The fact is that Zabbix has a learning curve that is a bit steeper than other platforms, because Zabbix’s size and the number of features that this platform provides are greater than those of any other technology. Now, challenges will arise because of the large number of features and possibilities with Zabbix, but it grants us fairly wide comprehensiveness in our IT environment. The use of all of these resources without proper planning and understanding of all of Zabbix’s components can lead us to a common situation where the monitoring environment will collapse, thus making us abandon the platform.

**Good practice**

Your environment may have been born in a more controlled and planned manner. You or your staff may have received the correct guidelines, or participated in a training session before starting the deployment of Zabbix in your environment to avoid some inconvenience as the environment grew. However, such cases are rare. My experience shows that first, Zabbix is born out of a desire or requirement of technical teams involved with IT infrastructure, and those responsible for these deployments are solitary heroes who seek a high-level solution with low costs. Often, we do not have the budget to implement this project in our companies. As this monitoring platform shows its value to your company in terms of business, it starts gaining importance and strength. At this point, however, the first challenge arises, because the screen response in the director’s or president’s office must be an adequate and consistent response to the user’s satisfaction (which is quite sensitive in this case).
Simplifying Zabbix

Zabbix's basic objective is data gathering. Basically, this is what Zabbix does. Of course, the collected data will be processed and stored for future comparisons or consultations at regular time periods. The data will also be compared with thresholds (triggers) and viewed by users on screens, maps, and charts. It needs to be cleaned on a routine basis. At this point, things start becoming more complex with Zabbix. Although the platform has a very simple concept (work in data gathering and evaluation), there are factors related to the processing of such data that must be evaluated, and some parameters need to be adjusted to ensure that Zabbix operates properly. This leads to satisfaction of the users and administrators.

In later chapters, we will talk about the concepts and terminology that we define here in this chapter, for your better understanding.

Zabbix was born with its own concepts, terms, and ways of monitoring functions. As you may know, Zabbix was created by Alexei Vladishev in 1998 (in 2001, he published Zabbix's first release). Since the first release, Zabbix has had specific guidelines to work:

- All rules about thresholds, triggers, and alerts are managed by the Zabbix server (not the Zabbix agent).
- Almost all configuration tasks are done at the Zabbix GUI.
- The Zabbix GUI is PHP-based (using a web server and a web browser).
- All of the data (configuration and historical) is stored in a relational database (we are close to storing historical data in a NoSQL database).
- The Zabbix server was developed in the C language (mainly because C has a small footprint).

With this information, we need to start thinking mainly about four Zabbix components: the Zabbix server, Zabbix proxy, Zabbix database, and Zabbix GUI. Each one has its own characteristics and requirements:

- **Zabbix server**: This is the engine—the collector itself—responsible for gathering and/or receiving data from the environment. It is written in the C language and communicates with the Zabbix agents, Zabbix proxy, and Zabbix database. It is the main component of this environment, and manages all the rules (collections, triggers, alerts, and so on).
- **Zabbix GUI**: This is the Zabbix interface where users can see the data gathered by the Zabbix server in the environment. It is written in PHP, uses a web server (supporting PHP), and communicates with the Zabbix database. The Zabbix GUI communicates with the Zabbix server for some minor functions.
Evolution of Zabbix

- **Zabbix database**: This is the Zabbix data repository. The backend database of Zabbix can be Oracle, IBM DB2, PostgreSQL, MySQL, or SQLite3. In this book, we will cover examples and case studies used with MySQL as the database.

- **Zabbix proxy**: This is an optional component, but as we will see throughout the chapters, when it comes to Zabbix's performance, it is of utmost importance. Its main function is to assist the Zabbix server in data gathering in the monitored hosts. The data gathered by the Zabbix proxy is first kept in a temporary database, and is subsequently sent to the Zabbix server.

These four components create the Zabbix monitoring solution. Throughout this book, we will cover the main aspects related to the performance of each of these components, looking for a clearer and more objective view of the elements and configuration parameters that will directly influence performance.

Challenges in Zabbix

Your challenge starts when you convince your boss that you are responsible for implementing an open source tool to monitor the IT environment for your company. Time progresses and the Zabbix platform starts gaining more and more responsibilities and visibility. However, suppose some important steps were forgotten, or you didn't have all of the information needed to carry out more detailed planning or sizing. The fact is that Zabbix has earned the reputation of an all-seeing eye, and now your company uses Zabbix to support business growth and ensure proper delivery of services.

Since 2007, I have been working with the Zabbix community, and since 2012, I have been a Zabbix certified trainer. In these days, I have heard and seen a lot of guys talking about performance issues with Zabbix. I have no doubt that most of these problems are related to a misconfiguration or misunderstanding about Zabbix's parameters and concepts. Some basic information about Zabbix that people usually don't know or don't care about is as follows:

- **The number of hosts isn't the most important thing for performance**: Usually, people ask, "How many hosts can I manage with Zabbix?" The right question should be, "How many new values per second (nVPS) can I manage with Zabbix?" So, you need to know that one host gathering 100 items is the same as 100 hosts gathering one item each.
• **The default templates shouldn't be used in a production environment:**
  It happens that default templates are the only examples that show you how to use item keys, triggers, graphs, LLD, macros, and other Zabbix features and functions. Such templates usually have gathering intervals shorter than what you really need.

• **How many users will use the Zabbix interface:** This is a point that is almost always forgotten. Usually, people start using Zabbix alone or together with a few guys, and they have only a few maps and screens. But what if you need to create a lot of users to use and explore the Zabbix interface? What if your boss asks you to create some dashboards, putting a lot of data together? At this point, you'll start to think about web server performance.

• **Using a default database deployment:** The MySQL database comes with almost all Linux distributions, but `my.cnf` isn't fit to work with Zabbix. I mean, the default MySQL deployment isn't the best configuration that you can work with. Of course, you will need to adjust some basic (maybe advanced) parameters to attain the best performance with Zabbix. People don't care about read or write parameters. It’s very important to know how Zabbix works and then prepare your database to work with Zabbix.

• **Item types and value types will directly affect performance:** Do you know that active items are better than passive items? It's very important to know that when using active items, the Zabbix server has less work to do, and each Zabbix agent handles its own queue. Do you know that numerical data is better than text data? Zabbix uses different tables for each data type (float, integer, text, log, and so on), and each database table has a different row configuration.

• **Time retention needs to be shorter than the template's default configuration:** By default, Zabbix works for 90 days to retain historical data and 365 days (a year) to retain trends data. Of course, you don't need to retain 90 days of historical data from an `icmp.ping` item key. Nor do you need to retain 365 days of trends data from this key. So, you need to choose the right period to retain your data (historical and trends). You will need to retain some data for a long time, and you can get rid of the other data earlier.

• **The number of triggers and the functions with them will affect performance:** Some people don't realize that a trigger with a very simple function, such as `last()`, has better performance than a trigger with a more complex function, such as `min()`, `max()`, or `avg()`.

• **Items that are not supported can affect Zabbix's performance:** The Zabbix server will always try to gather these items, and if they have some error, the Zabbix server will work without results.
List of don'ts
You need to know some basic things about what to avoid when you start working with Zabbix.

Starting a Zabbix deployment without planning
Lack of planning is the main item in this list of don'ts. Sorry to be repetitive, but if you start a Zabbix deployment without planning, you will have performance issues. So, it is important to know both your environment and Zabbix well.

Use of default templates
Templates that Zabbix SIA sends together with Zabbix are only for testing, and they may be for proving concepts; they are not for use in a production environment. We’ll need to create our own templates based on our needs. In the next chapter, you will know that all the default templates are not really meant for you.

Use of default database settings
It doesn't matter which database engine are you using (Oracle, MySQL, pgSQL, or DB2). You will need to change some parameters and tune your database engine. So, you'll need to know about the SQL statements that Zabbix uses and a few more things, as follows:

- How many users will you have? This is precious information to know if you need to tune your database for read or write operations.
- You need knowledge about the hardware. Do you have a storage-backed database? Do you have local disks? How about SAS, SATA, and SSD disks? This is another piece of important information to help you with database tuning.
- Do you have dedicated hardware for the database server? If yes, you can manage the database memory settings (cache and buffer) much better.
- Is this database server a shared server? I mean, is your database server dedicated to the Zabbix database or you have another database together with it? And if you change something to improve Zabbix's performance, will it affect another application?
The beginning of the real challenge

When I started working with Zabbix and deployed our first project with Zabbix, some thoughts that surrounded me were as follows: whether this tool is a reliable one, whether it is possible to use it in large environments, how many users I can have using the Zabbix GUI, and how many hosts or npvs I can manage with Zabbix.

Of course, we started working with Zabbix after a lot of tests and simulations, but a test environment isn't the same as most customers' environments.

In our first project with Zabbix in a large environment (Zabbix version 1.4), we had no Zabbix proxy, caches, or any buffer inside the Zabbix server. Of course, we experienced a lot of troubles regarding performance. We started this project using the Oracle database (because our customer wanted it). After working on this project for some weeks, we began to realize that our performance could be degraded. Our Zabbix GUI was unresponsive, and we were getting some screen errors saying something related to table locks. At such times, the Zabbix database used to execute a lot of SQL update operations in a table called _ids_. The _ids_ table is very short, with an unexpressive column and row amount. But why did we get these errors? How was Zabbix doing its work?

At this point in time, we asked Zabbix SIA about this behavior. They told us that we had no performance issues with the Zabbix server, but had issues with the Oracle database. We received this information and thought that maybe the application (the Zabbix server) has no performance issues, so let's tune the Oracle database. Therefore, we started working hard on tuning a lot of Oracle parameters. Our Oracle DBAs adjusted all the possible parameters to improve our performance. But we still had performance issues, even though they were few. At this point (2007 to 2008), we were stuck with the project and went back to the planning table.

Our Zabbix-certified guys began a deep investigation to know exactly how (by SQL statements or TCP/IP stack), when (while gathering new values or accessing gathered data), why (to clean-up old data or to create trends data), by whom (the Zabbix server pollers processes or Zabbix server trappers processes), and with how much effort Zabbix will be needed to execute all tasks.

Of course, we have new features nowadays, and it is easier to manage performance. When we started working with Zabbix, we used to read the Zabbix forum threads, looking for a magical solution to our errors. But our environment was not the same, as some specific tuning was made. I mean, zabbix_server.conf, which works like a charm on my environment, can be bad for you.
From the Zabbix forums, it is possible to get a lot of tricks and tips on how to improve Zabbix's performance. Some say they are happy with Zabbix's performance in a large environment and others say they are unhappy with it in a small environment.

But you really need to know about Zabbix's internal tasks, flows, and process. You also need to know about your environment. After using all of this knowledge, you will experience the best monitoring tool you ever knew.

Summary

In this chapter, we saw that the start with Zabbix is not always glamorous and not always start with the most advanced features. The important thing is to realize that planning is the basis of a successful implementation of Zabbix. For reasonable planning, it is important to know the tool reasonably well, and if we are going to plan properly, we must know the tool in great depth.

If you're experiencing performance issues with Zabbix, it is likely that you can solve them without new hardware or software. But for this, you need to know Zabbix, its components, and the possibilities of each. In the next chapter, we will move on to cover this newborn environment, as this is where the majority of people started experiencing performance problems. What happens when everyone wants to use Zabbix? What is the impact of disorderly growth? Let's try to get those answers in the following chapters.
Where to buy this book
You can buy Zabbix Performance Tuning from the Packt Publishing website.
Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.
Click here for ordering and shipping details.