

Running large installations of PRTG in a virtual environment

Best Practice Guide



Best Practice Guide: Running large installations of PRTG in a virtual environment
Version 1.0

© 2020 Paessler AG

All rights reserved. No parts of this work may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: August 2020 in Nuremberg

Table of contents

- Part 1 Introduction 4
 - 1 Purpose and scope 6
- Part 2 Preparation of the virtual infrastructure 7
 - 1 Basic configuration 9
 - 2 Advanced configuration 13
 - 3 Further optimization and troubleshooting 14

Part 1

Introduction

1 Introduction

Virtualized infrastructures are becoming more and more important because of increased efficiency, availability, reliability, and mobility, and because of reduced costs. A long-established concern regarding virtualization is the expected performance penalty when moving applications to a virtual environment. However, hardware and virtualization vendors have worked together over the past few years to nearly eliminate virtualization overhead. Today, there is practically no performance penalty if you virtualize operating systems and applications. Virtualization adds a new layer of complexity, however. Therefore, it is important to follow some basic rules as well as some important architecture and configuration principles to achieve the same level of performance as on a physical server.

The best practices presented in this document address such architecture and configuration principles for running large installations of PRTG Network Monitor (hereinafter referred to as PRTG) in a virtual environment. Please implement and follow these guidelines closely to get the best performance and reliability for your virtualized PRTG installation.

1.1 Purpose and scope

This Best Practice Guide is intended for IT professionals who want to learn more about running larger installations of PRTG in a virtualized infrastructure.

The recommendations in this document are specific to VMware as the market leader in virtualization for larger installations of PRTG with more than 5,000 sensors. Most PRTG installations from 500 to 5,000 sensors do not need any specific optimization and are also supported on Microsoft HyperV.

This document assumes that you, the reader, have a working knowledge of PRTG and VMware ESXi.

Part 2

Preparation of the virtual infrastructure

2 Preparation of the virtual infrastructure

In the following sections, you can find some best practices and recommendations regarding the preparation of your virtual infrastructure:

- [Basic configuration](#) ⁹
- [Advanced configuration](#) ¹³
- [Further optimization and troubleshooting](#) ¹⁴

2.1 Basic configuration

There are some basic configuration best practices that you must follow if you want to run a large installation of PRTG on a virtual machine.

- [Software](#) ⁹
- [Virtual machine hardware](#) ¹⁰
- [Server hardware](#) ¹¹

Software

For software considerations, we recommend the following:

- Do not run large PRTG installations in highly shared virtual environments.
- Make sure that your entire virtual environment is up to date.
- For guests, use a server operating system.
- Make sure that the ESXi host and the guest operating system use a reliable time source.

Best practice	Detailed information
We recommend that you avoid running large PRTG installations in virtual environments that are highly shared.	Examples for highly shared virtual environments are: <ul style="list-style-type: none"> ▪ Amazon EC2 T2 instances ▪ Microsoft Azure A-series and B-series virtual machines ▪ Google Compute Engine instances of the type E2 and N1 ▪ VMware vCloud environments
We recommend VMware ESXi version 6.7 U3 .	Recommended hypervisor in order of highest performance and scalability: <ul style="list-style-type: none"> ▪ VMware ESXi 6.7 U3 is recommended ▪ VMware ESXi 6.5 U3 is supported ■ For more information, see the VMware document Performance Best Practices for VMware vSphere 6.7 .
We recommend Windows Server 2019 or Windows Server 2016 .	Recommended operating systems in order of latest release: <ul style="list-style-type: none"> ▪ Windows Server 2019 ▪ Windows Server 2016 ▪ Windows Server 2012 R2 (only for remote probes in PRTG)

Best practice	Detailed information
<p>Use w32time or NTP as the time source for both the PRTG core server systems and the remote probe systems.</p>	<p>Any time leap can have considerable effects on your monitoring system, for example, false alarms, missing alarms, or sensors that show an Unknown status.</p> <p>Most instances have time synchronization enabled in Windows. This is the default setting for a Windows operating system as a member of an Active Directory domain. If this is the case for your operating system, make sure that you disable the time synchronization via VMware Tools. Having time synchronization enabled both in the Windows operating system and via VMware Tools causes time leaps and further issues.</p> <p>■ For more information, see the VMware Knowledge Base: Timekeeping best practices for Windows, including NTP (1318).</p>

Virtual machine hardware

For the virtual machine hardware, we recommend the following:

- Provide a sufficient number of virtual CPUs (vCPU) and reserve the CPU resources.
- Provide sufficient, reserved memory.
- Use the fully virtualized virtual machine hardware adapters that provide the best performance with the lowest overhead.
- Use thick-provisioned disks for the virtual machine. Avoid using thin-provisioned .vmdk files.
- Make sure that the storage performance is sufficient, monitor the datastore latency, and watch out for the "noisy neighbor" effect.

Best practice	Detailed information
<ul style="list-style-type: none"> ▪ For a PRTG core server system with a few sensors on the local probe only, we recommend a vCPU configuration with one virtual socket. ▪ For a PRTG core server system with a certain number of sensors on the local probe, we recommend 2 virtual sockets in the virtual machine configuration. 	<p>Use the following configurations:</p> <ul style="list-style-type: none"> ▪ Allocate 8 to 24 vCPUs to the PRTG core server. ▪ Allocate 4 to 8 vCPUs to remote probes on 1 virtual socket. <p>By default, VMware uses a configuration with multiple virtual sockets that each have a single core CPU. However, more than 2 virtual sockets do not have any positive impact on the performance of PRTG.</p>

Best practice	Detailed information
<p>We recommend that you provide a minimum of 16 GB RAM. However, 32 GB RAM or more might be necessary.</p>	<ul style="list-style-type: none"> ▪ Monitor your PRTG installation closely. You might need to add additional RAM depending on your configuration and usage of PRTG. ▪ Reserve all guest memory.
<ul style="list-style-type: none"> ▪ We recommend that you use a VMXNET3 virtual network adapter. ▪ We recommend that you use a PVSCSI or NVMe disk adapter. 	<p>The recommended virtual machine hardware adapters provide the best performance with the lowest overhead.</p>
<p>We recommend that you use thick-provisioned .vmdk files.</p>	<p>The PRTG core server is a write-intensive application. There is a storage performance penalty on write for thin-provisioned virtual machines because storage blocks have to be allocated on the local virtual disk before the write can proceed.</p>
<ul style="list-style-type: none"> ▪ For storage configurations, we recommend the use of SSDs or NVMe. The use of SAS over SATA is preferred. ▪ For remote storage configurations, you need to keep an eye on the read/write queue times and rates. ▪ Move "noisy neighbors" that keep the datastore busy and negatively affect the performance of PRTG to a different datastore. 	<ul style="list-style-type: none"> ▪ We recommend that you use dedicated, purpose-built datastores. Large-scale installations of PRTG that usually have many concurrent users and that need a high responsiveness benefit from a dedicated datastore that guarantees performance. ▪ Sharing datastores with other virtual machines might be supported, but only with compatible, low I/O workloads. Keep the number of virtual machines on shared datastores as small as possible. ▪ Monitor datastore latency and watch out for the "noisy neighbor" effect. If PRTG is backed by a datastore that frequently experiences a high read or write latency, either migrate the virtual machine that runs PRTG or the "noisy neighbor" to a different datastore to improve performance. ▪ If possible, try to avoid RAID 6 storage configurations. ▪ Do not host other virtual machines that generate substantial storage I/O on the virtual infrastructure of PRTG. ▪ Do not host other write-intensive workloads on the same datastore.

Server hardware

As server hardware setting, we recommend the following:

- Select the power usage mode that offers the highest possible performance.

Best practice	Detailed information
<p>Turn off power-saving functions at the BIOS or UEFI level and at the operating system level.</p>	<p>Most modern server hardware offers different modes in its BIOS or UEFI setup regarding power usage versus performance.</p> <p>Selecting the highest possible performance mode might result in slightly higher power consumption. Consult your server vendor for details.</p>

2.2 Advanced configuration

The following configuration best practices are for advanced users.

- Configure a high CPU Latency Sensitivity.
- Monitor the critical CPU-related metrics.

Best practice	Detailed information
<p>We recommend that you configure a high CPU Latency Sensitivity on the machine that hosts the PRTG core server.</p>	<p>The PRTG core server is a single-process, heavily multithreaded application that reacts to CPU Latency (CPU Latency = CPU Co-Stop + CPU Ready + operating system scheduler time) in a very sensitive manner, like a database.</p> <ul style="list-style-type: none"> ■ See the following VMware documents on how to configure latency sensitivity in vSphere 5.5 and vSphere 7: <ul style="list-style-type: none"> ▪ Deploying Extremely Latency-Sensitive Applications in VMware vSphere 5.5 ▪ Adjust Latency Sensitivity
<p>We recommend that you monitor the following CPU-related metrics:</p> <ul style="list-style-type: none"> ▪ CPU Usage ▪ CPU Ready Time in milliseconds (ms) ▪ CPU Co-Stop Time in ms 	<p>Note:</p> <ul style="list-style-type: none"> ▪ For applications like the PRTG core server or a database, you need to try to balance out the number of virtual cores, the CPU Usage, as well as the CPU Ready Time and the CPU Co-Stop Time. Do not blindly follow the rule that lower CPU Usage equals lowering the number of virtual cores. ▪ CPU Ready: When virtual CPUs (vCPU) are assigned to a virtual machine, it must wait for all configured virtual cores to be available to execute its CPU cycle, even if it will not use all of its virtual cores. ▪ CPU Co-Stop: When the hypervisor tries to reserve the physical core to be assigned as a vCPU, it must ensure that all configured vCPUs on one virtual socket come from the same physical core. <p>Adding too many vCPUs to a virtual machine or virtual socket can lead to an increase in CPU Ready and CPU Co-Stop Time. Try to target a CPU Ready Time that is lower than 25 ms on average and a CPU Co-Stop time that is lower than 10 ms on average. Note that vCenter displays the CPU Ready Time in ms, whereas ESXCLI accessed via SSH displays the CPU Ready Time in percent.</p> <ul style="list-style-type: none"> ■ For a tool that helps you convert the CPU Ready Time in ms to percent and vice versa, see www.vmcals.com.

2.3 Further optimization and troubleshooting

If you followed all previous configuration best practices and there are still issues, have a look at the following optimization and troubleshooting tips.

- Align the virtual CPU (vCPU) layout to the actual hardware or host configuration.
- Consider your configured NUMA nodes.
- Consider turning off Hyper-Threading.
- Use CPU pinning.

Best practice	Detailed information
We recommend that you align your vCPU layout to the actual hardware or host configuration.	The hypervisor has to completely allocate all the vCPUs on a virtual socket before giving them to the virtual machine, preferably on the same hardware socket at a given time.  Do not try to stretch the socket boundaries by not synchronizing the vCPU layout.
We recommend that you consider NUMA nodes that you have configured at the host level as distinct physical sockets.	If it is not possible to consider NUMA nodes as distinct physical sockets, you need to at least align your vCPU layout to match the vSockets to whole NUMA groups.

■ For more information, see the VMware document [Performance Best Practices for VMware vSphere 6.7](#).

