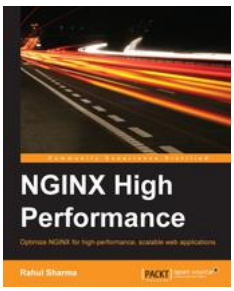


Search this title...

Bookmarks (0)

1: Working with NGINX ✔

2: Benchmarking the Server

3: Tweaking NGINX Configurati...

Chapter 3: Tweaking NGINX Configur...
(/Mapt/Book/Networking-And-Servers/97...

NGINX configuration syntax
(/Mapt/Book/Networking-And-Servers/97...

Configuring NGINX workers
(/Mapt/Book/Networking-And-Servers/97...

Configuring NGINX I/O ✔
(/Mapt/Book/Networking-And-Servers/97...

Configuring TCP
(/Mapt/Book/Networking-And-Servers/97...

Setting up the server
(/Mapt/Book/Networking-And-Servers/97...

Summary
(/Mapt/Book/Networking-And-Servers/97...

4: Controlling Buffers, Timeouts...

5: Configuring the Network Stack

6: Using NGINX Cache

7: Extending NGINX

Appendix A: Index

Configuring TCP

HTTP is an application-based protocol, which uses TCP as the transport layer. In TCP, data is transferred in the form NGINX provides directives to alter the behavior of the underlying TCP stack. These parameters alter flags for an ind

TCP_NODELAY

TCP/IP networks have the "small packet" problem, where single-character messages can cause network congestion packets are 41 bytes in size, where 40 bytes are for the TCP header and 1 byte has useful information. These small p around 4000 percent and can saturate a network.

John Nagle solved the problem (Nagle's algorithm) by not sending the small packets immediately. All such packets ar of time and then sent in one go as a single packet. This results in the improved efficiency of the underlying network. for up to 200 milliseconds before sending the data packages to the client.

It is important to note that the problem exists with applications such as Telnet, where each keystroke is sent over wi a web server, which severs static files. The files will mostly form full TCP packets, which can be sent immediately inst milliseconds.

The `TCP_NODELAY` option can be used while opening a socket to disable Nagle's buffering algorithm and send the NGINX provides the `tcp_nodelay` directive to enable this option. The directive is available under the `http` , sections of an NGINX configuration:

```
http{
  tcp_nodelay on;
}
```

The directive is enabled by default.

Note

NGINX use `tcp_nodelay` for connections with the keep-alive mode.

TCP_CORK

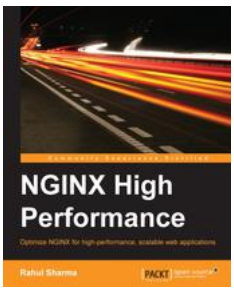
As an alternative to Nagle's algorithm, Linux provides the `TCP_CORK` option. The option tells the TCP stack to app when they are full or when the application instructs to send the packet by explicitly removing `TCP_CORK` . This res packets being sent and, thus, improves the efficiency of the network. The `TCP_CORK` option is available as the `T` and Mac OS.

NGINX provides the `tcp_nopush` directive to enable `TCP_CORK` over the connection socket. The directive is a `ver` , and `location` sections of an NGINX configuration:

```
http{
  tcp_nopush on;
}
```

The directive is disabled by default.

Note



NGINX uses `tcp_nopush` for requests served with `sendfile` .

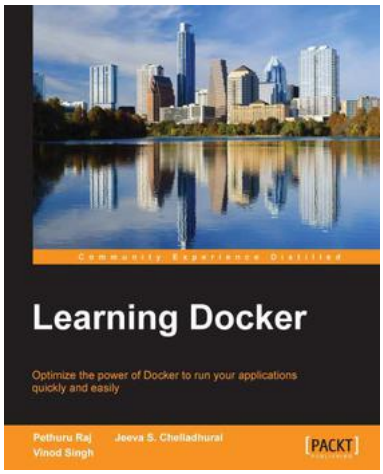
Setting them up

The two directives discussed previously do mutually exclusive things; the former makes sure that the network laten tries to optimize the data packets sent. An application should set both of these options to get efficient data transfer.

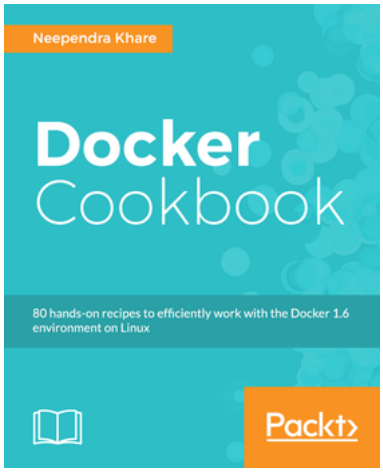
Enabling `tcp_nopush` along with `sendfile` makes sure that while transferring a file, the kernel creates the n packets before sending them over wire. The last packet(s) can be partial TCP packets, which could end up waiting wi NGINX makes sure it removes `TCP_CORK` to send these packets. Since `tcp_nodelay` is also set at this point, t sent over the network without any delay.

Next Section (/mapt/book/networking-and-servers/9781785281839/3/ch03lvl1sec21/settin

Related titles



(/mapt/book/virtualization_and_cloud/9781784397937)



(/mapt/book/virtualization_and_cloud/9781783984862/1/ch01lvl1sec09/introduction)