



# Deployment Guide

NorthScale Memcached Server

Version 1.0

**© 2010 NorthScale, Inc.**

[Unpublished - all rights reserved under the Copyright Laws of the United States]

THIS DOCUMENTATION CONTAINS CONFIDENTIAL INFORMATION AND TRADE SECRETS OF NORTHSSCALE, INC. USE, DISCLOSURE, OR REPRODUCTION IS PROHIBITED WITHOUT THE PRIOR WRITTEN PERMISSION OF NORTHSSCALE, INC.

# Contents

---

<b>Contents</b> .....	<b>3</b>
<b>Deployment Guide</b> .....	<b>4</b>
<b>Introduction</b> .....	<b>5</b>
NorthScale Memcached Server.....	5
NorthScale, Inc.....	5
<b>Design Guidelines</b> .....	<b>6</b>
Design Considerations.....	6
Benefits of an Independent Cache Tier.....	6
Hardware Considerations.....	7
Topology Considerations.....	7
Protocol Considerations.....	8
Data Considerations.....	8
Working with the Cache Tier.....	8
<b>Deployment Scenarios</b> .....	<b>9</b>
Using the NorthScale Smart Client Library.....	9
Using Standard and Recommended Client Libraries with the Cache Port (11211).....	9
Using Standard and Recommended Client Libraries with the Proxy Port (11212) Only.....	10
Using Standard and Recommended Client Libraries with the Cache (11211) and REST Management API (8080) Ports.....	11
<b>Contact NorthScale</b> .....	<b>12</b>
Corporate.....	12
Technical Support.....	12
<b>Index</b> .....	<b>13</b>

## Deployment Guide

This document describes how to deploy the NorthScale Developer KitNorthScale Memcached Server in a production environment. Use the [Design Guidelines](#) to guide you in designing your deployment, and use [Deployment Scenarios](#) to help you determine the best deployment approach for your environment.

## Introduction

Welcome to the NorthScale Memcached Server.

### NorthScale Memcached Server

NorthScale Memcached Server is an enhanced distribution of memcached, created and supported by leaders of the memcached open source project. Memcached is the de facto standard for web application data caching because it is simple, fast, infinitely scalable...and free. Without compromising any of these properties, NorthScale has packaged memcached making it easy to get, install and manage; while adding secure multi-tenancy, dynamic cluster scaling and browser-based administration. NorthScale Memcached Server, first in a family of elastic data infrastructure software products from NorthScale, is an ideal starting point for IT teams adopting a scale-out data architecture to reduce the cost and complexity of data management for their web application and cloud computing infrastructures.

### NorthScale, Inc.

NorthScale, Inc., the leading provider of elastic data infrastructure software, provides products that enable customers to dramatically lower costs while simultaneously improving the scalability and performance of their web applications. NorthScale is in production behind some of the world's busiest web applications: it is the primary database for the popular FarmVille and Café World applications at Zynga; and it provides a shared data management platform for NHN, Korea's largest web application operator with nearly 70 million unique users. Founded in 2009 and headquartered in Mountain View, Calif., NorthScale is a privately held company funded by Accel Partners and North Bridge Venture Partners.

## Design Guidelines

This topic describes issues that you should consider when designing your NorthScale Memcached Server deployment.

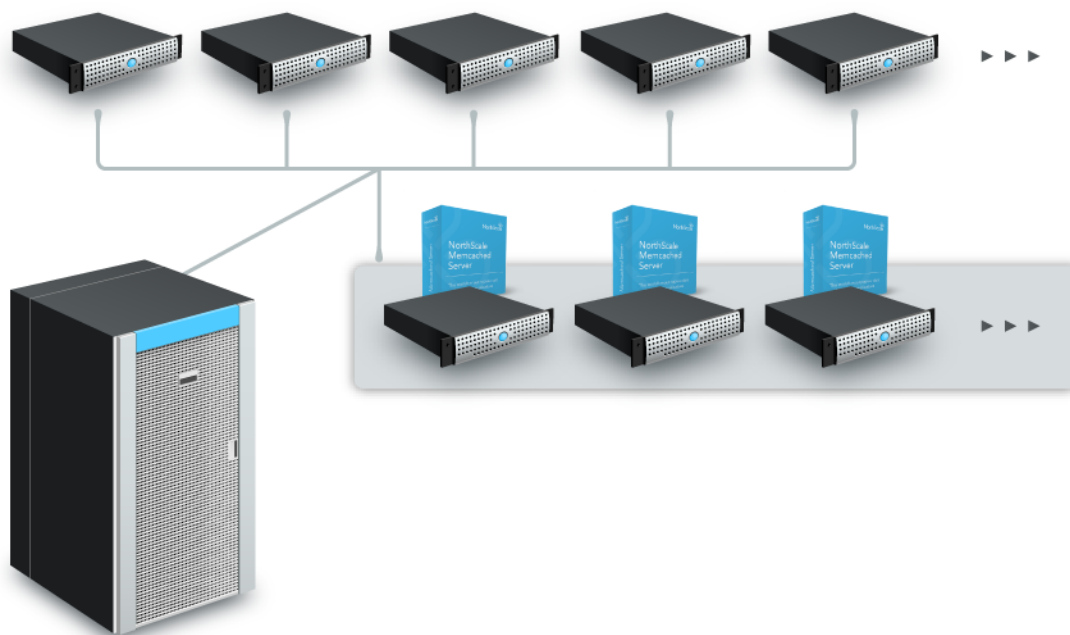
### Design Considerations

The power of memcached is in its distributed architecture and scalability. The simple memcached API allows web developers to store arbitrary data in-memory without worrying about the traditional scalability concerns of an RDBMS. Memcached scales simply by adding server nodes to a running cluster. As you add server nodes, the size of your available cache increases and your cluster scales out horizontally without application code changes.

### Benefits of an Independent Cache Tier

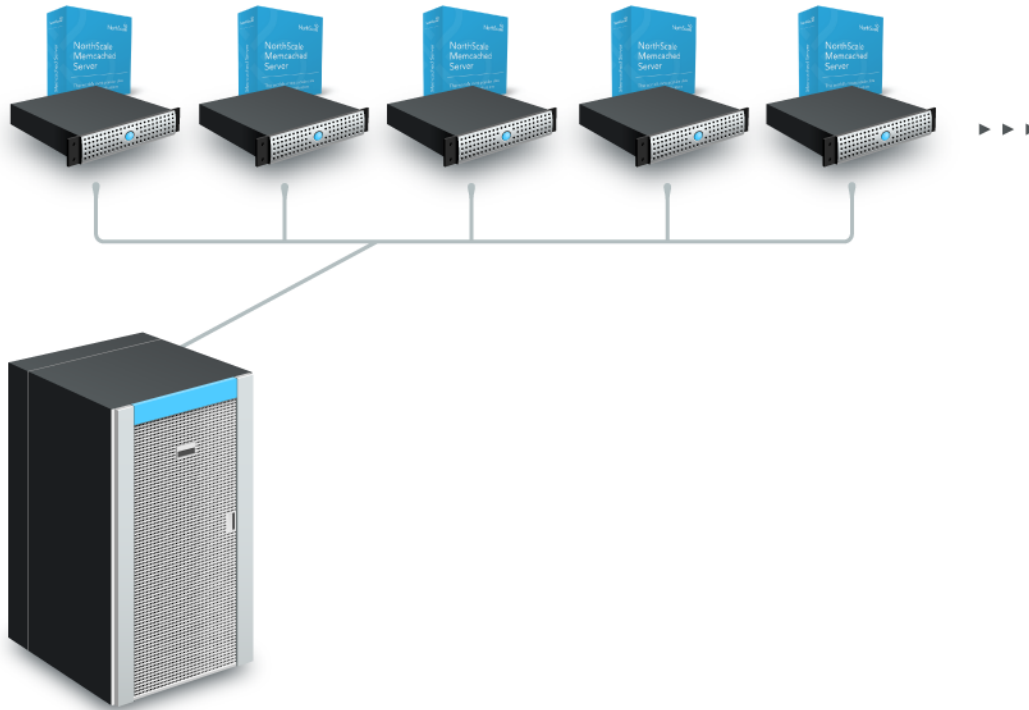
In typical tiered web architectures, stateless web application servers (behind a load balancer) scale independent of stateful database systems. Consequently, web application servers are considered to be easily scalable and stateless. Meanwhile, database systems are extremely difficult to scale or have upper limits on their ability to scale up vertically. When memcached is added alongside a traditional RDBMS, however, it is possible to build an extremely scalable, high-performance data layer without re-architecting your web applications.

It is highly recommended that NorthScale Memcached Server be deployed on its own cache tier in order to leverage the immediate scalability benefits of memcached.



This recommended deployment topology requires that NorthScale Memcached Server run on dedicated server hardware with similar or identical system configurations. As your web application's requirements change, both adding and removing NorthScale Memcached Server nodes from a cluster is extremely straightforward.

Alternatively, NorthScale Memcached Server can be installed and configured on the same server nodes as web application servers.



More often than not, however, as described above, it is necessary to scale your web application servers separately from your data and cache layers. This alternative deployment topology requires less system resources but introduces future scalability challenges.

In addition, because NorthScale Memcached Server software is designed to be symmetric and stateless, it can easily be cloned and deployed in virtualized environments.

## Hardware Considerations

It is highly recommended that NorthScale Memcached Server run on 64-bit equipment with at least 16GB of memory. Of course, it is possible to run NorthScale Memcached Server on 32-bit equipment with significantly less memory as well.

- Recommended: Quad-core, x86-64 compatible, 3GHz, 32GB RAM
- Minimum: Dual-core, x86-32 compatible, 2GHz, 4GB RAM

## Topology Considerations

When planning your NorthScale Memcached Server deployment in a production environment, consider the following topology issues:

- Each server node is installed on a different host machine.
- Clusters span across multiple machines.
- It is recommended that a production deployment contain clusters with a minimum of two (2) server nodes.
- Clusters can grow to 100 server nodes and beyond.

- Every server node that is part of the same cluster must be part of the same subnet.
- Every server node must be able to communicate directly with every other server node on a particular set of ports (as described in *Getting Started*).
- A single server node is considered to be a cluster of one.
- Consider adding nodes in pairs.

## Protocol Considerations

NorthScale Memcached Server is compatible with both the ASCII and binary memcached protocols. NorthScale recommends using the binary protocol implementation because of its performance benefits and support for SASL authentication. The NorthScale Smart Client Library, for example, implements the binary protocol. Not every standard memcached client library supports the binary protocol. For additional information on the binary protocol implementation, refer to the *Developer Guide*. All smart clients use the binary protocol.

## Data Considerations

Web developers are encouraged to store database results, page fragments, application state, session data, and whatever else in the cache. Data that requires frequent database requests are particularly effective for caching. By default, NorthScale Memcached Server implements the LRU (Least Recently Used) cache structure in addition to expirations. When the cache is full, items are replaced in the following order: expired items first, followed by unused items.

## Working with the Cache Tier

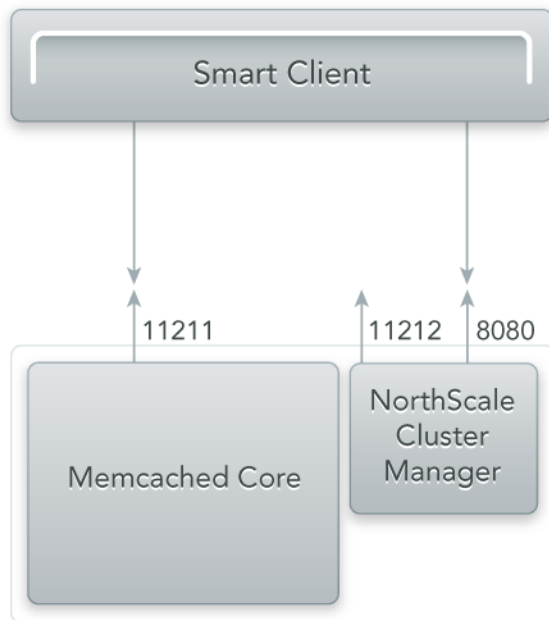
Because NorthScale Memcached Server is an in-memory, distributed cache, there is always a risk that the data requested by an application is not available and a database request needs to be made to retrieve it. This means that web application data stored in a cache tier is volatile and does not persist. NorthScale is currently in beta with software solutions that address these and other issues.

## Deployment Scenarios

This section describes four different deployment scenarios for NorthScale Memcached Server. When deploying NorthScale Memcached Server, web developers have an option to use traditional memcached, recommended or smart client libraries. Each scenario offers certain advantages and tradeoffs for you to consider when designing a deployment in your production environment. Note that the referenced port numbers can be adjusted if needed using the NorthScale Web Console.

### Using the NorthScale Smart Client Library

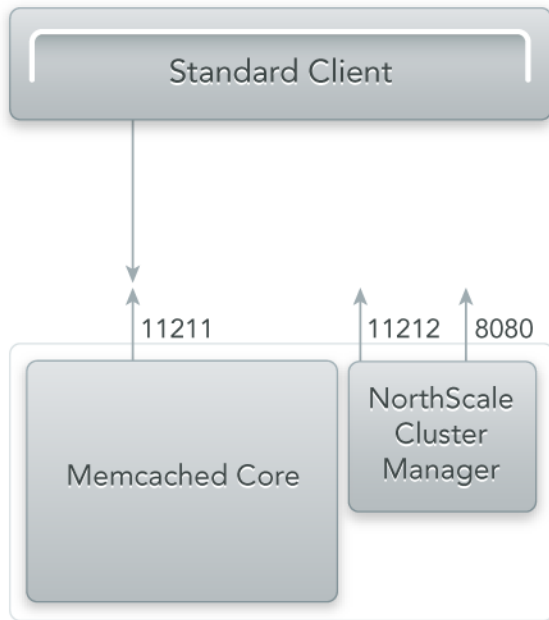
In the following scenario, a smart client uses both ports 11211 and 8080.



This kind of deployment is ideal because the solution is deployed on an independent cache tier (assuming you can select and use a new client machine). Using a new smart client provides advantages without requiring you to write custom code. Smart clients know how to communicate with the REST API, get information about updates to the server list, and then adapt based on those updates. In addition, a smart client can support multi-tenancy.

### Using Standard and Recommended Client Libraries with the Cache Port (11211)

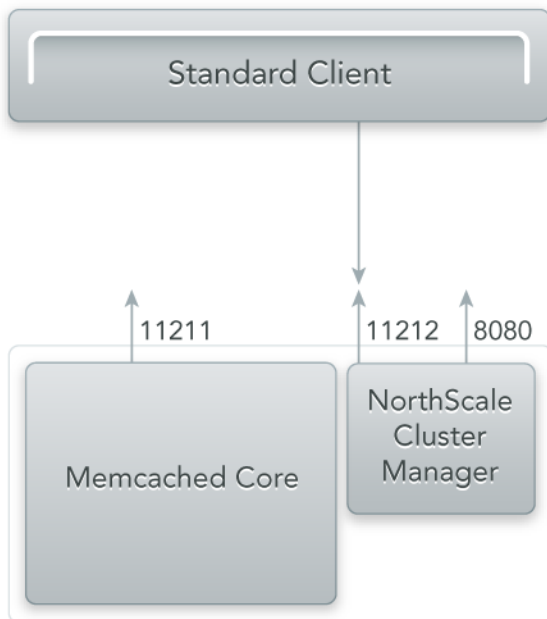
In the following scenario, a standard client uses port 11211 (cache port).



The advantage of this kind of deployment is that it offers the highest performance option for existing, off-the-shelf memcached client libraries. The disadvantage is that there is no ability to automatically add new memcached servers to a cluster without updating a server list file.

## Using Standard and Recommended Client Libraries with the Proxy Port (11212) Only

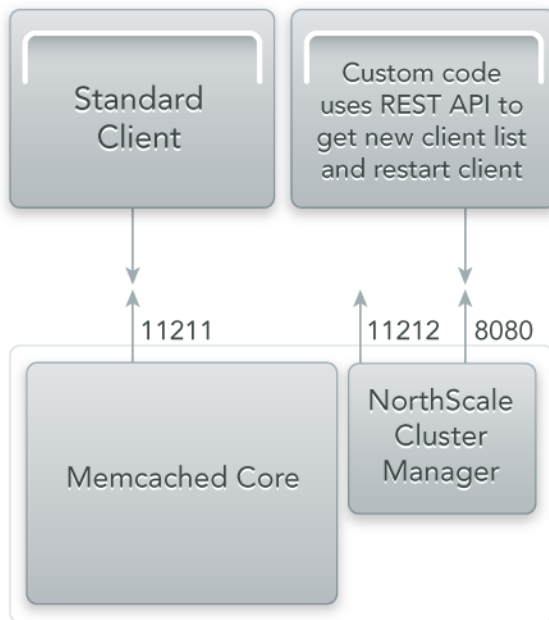
In the following scenario, a standard client uses port 11212 (proxy port).



This kind of deployment is best in cases where extreme performance is not required. The advantage of this approach is that it allows a completely automated addition and removal of server nodes. The NorthScale Cluster Manager can redirect client requests to the appropriate nodes when new nodes are added to the cluster. The tradeoff is that this approach adds some overhead until the server list can next be updated.

## Using Standard and Recommended Client Libraries with the Cache (11211) and REST Management API (8080) Ports

In the following scenario, a standard client uses port 11211 (cache port), and some code that leverages the NorthScale Management REST API uses port 8080, making requests using the NorthScale Management REST API to get a new server list and restart.



The advantage of this kind of deployment is that it offers:

- the highest performance option for existing, off-the-shelf memcached client libraries
- the ability to perform automated server list updates and restarts using simple client code that can talk with the REST API

## Contact NorthScale

### Corporate

**Phone:**

+1 650-417-7500

**Email:**

info@northscale.com

**Address:**

200 West Evelyn Ave., Suite 110

Mountain View, CA 94041

[www.northscale.com](http://www.northscale.com)

### Technical Support

**Support Line:**

+1 650-417-7500

[support@northscale.com](mailto:support@northscale.com)

<http://www.northscale.com/support/>

# Index

---

	<b>A</b>	
address		12
	<b>C</b>	
cache tier, working with copyright		8 2
	<b>D</b>	
data considerations		8
deployment		
overview		4
scenarios		9
design considerations		6
design guidelines		6
	<b>E</b>	
email address		12
	<b>H</b>	
hardware considerations		7
	<b>I</b>	
independent cache tier		6
	<b>P</b>	
phone		12
protocol considerations		8
	<b>T</b>	
topology considerations		7