EBOOK

DATABASE CONSIDERATIONS FOR DOCKER

NUODB®

Docker and NuoDB: A Natural Fit

Both NuoDB and Docker were developed with the same fundamental principles in mind: distributed processing, resource efficiency, administrative simplicity, and elastic scalability. This natural affinity means that the two technologies work in tandem, right out of the box.

This short e-book introduces a few key concepts about the natural alignment between Docker and NuoDB, and explains how and why NuoDB is a natural fit for container computing.

Introduction

NuoDB is a dynamic database solution well suited for migrating legacy applications to cloud architectures and accelerating development of new applications at global scale. NuoDB supports the SQL and ACID requirements that have driven the database industry, and is designed for horizontal scale, commodity infrastructure, redundancy, and policy-driven operations.

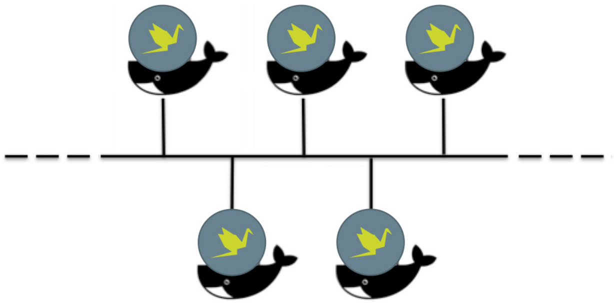
During the time that NuoDB has been maturing its peer-to-peer model, containers have grown in popularity in large part due to Docker and its ecosystem of tools and services. However, not all applications or services can exploit the on-demand scaling model that Docker enables – a problem especially apparent when it comes to database systems.

Goals of a Container

Containers are first a tool for isolation. Their lineage comes from the BSD implementation of 'chroot' and later jails, as a way to provide a secure environment for file access and execution. As systems requirements evolved, containers became a lightweight means to add resource isolation (memory, CPU, network etc.) on a per-process basis. The end result is a model for security and predictable operation of individual processes on shared resources.

Docker has taken the container model and added tools for instance definition, software dependency, and rapid development and deployment. The result is a toolchain that makes users more productive and makes software easier to run in disparate environments. Layered on the core container model are the clustering tools that make it simple to scale applications and monitor or automate large deployments. The key to realizing the benefits on containers is to use them with applications that require scalability and flexibility. For data management services specifically, there is the added challenge of being flexible in how and where data is stored.

> “
> **For data management services specifically, there is the added challenge of being flexible in how and where data is stored.**
> ”



The NuoDB on-demand model means new peers can be added at any time they are needed.

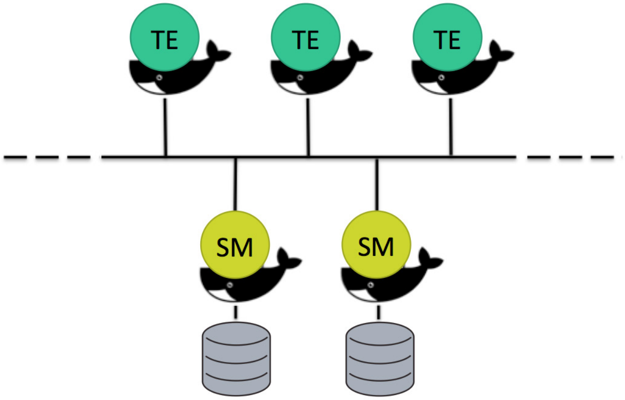
On-Demand Scalability and Resource Independence

NuoDB's underlying design principles align perfectly with container models like Docker. Organizations can use the two together to maintain application flexibility and achieve on-demand scalability on commodity hardware without sacrificing transactional consistency, data durability, or the familiarity of SQL.

A NuoDB database uses a peer model: coordinated processes that can run across a heterogeneous infrastructure independently. New peers (or containers for these peers) can be added any time they are needed to increase throughput, add redundant points of storage, reduce latency to remote users, or react to failures of existing peers in the database.

Like containers, each peer in a NuoDB database is also independent, meaning that each has the same capabilities and can take on exactly the same workloads. When Docker needs to handle more users or respond to higher throughput requirements, new containers are started. When that capacity is no longer needed, any of the containers can simply be stopped. NuoDB peers can respond in the same way requiring no special logic for partitioning data or balancing the number of peers.

One type of NuoDB peer is called a Transaction Engine (TE). These are purely in-memory, transient processes that collectively act as a caching and execution tier, making NuoDB an in-memory database. Starting a container to host a TE simplifies management, as there is no need to connect to specific storage or run on known hosts. Building on Docker clustering models, available servers can be treated as a logical resource pool, and automated resource management can be used to contract or expand the database as-needed.



Transaction Engines are purely in-memory, transient processes that collectively act as a caching and execution tier, making NuoDB an in-memory database. Storage Managers (SMs) focus on data durability (see page 6.)

Flexible Data Storage

Another type of NuoDB database peer is called a Storage Manager (SM). Like the Transaction Engine, Storage Managers are also caches but they focus on the task of making data durable and providing access to data that isn't currently in any TE's cache.

Because NuoDB databases can be deployed across heterogeneous resources, there is some flexibility in deciding where and how SMs are run. For instance, they could be run on physical or VM instances, outside the on-demand container model and closely aligned with physical storage. They could also be run inside containers, storing data against mounted volumes or network file shares.

The durability peers (SMs) are not the performance bottleneck of traditional RDBMSs because the running database is in-memory on the TEs. Also, because the NuoDB database is decomposed into an object representation, SMs can be deployed inside containers while using object stores, like Amazon S3, for storage.

Each SM may be configured to store a subset or the entirety of the database. The choice comes down to the deployment requirements of each application. Some may be focused on redundancy, some on disaster recovery and resilience, and others on write throughput or aggregate storage capacity. For example, applications running across physical sites might be optimized where data is typically needed or for governance rules stating where data can or cannot reside. Determining which storage peers to run in a container and which to run outside should be based on the requirements that are most important to that deployment.

Next Steps

It takes an architecture designed with on-demand capacity, resource independence, and storage flexibility to support data management in a container environment, and NuoDB does just that.

We invite you to try the NuoDB Tour and discover how you can benefit from its alignment with container models like Docker.

Take the Hands-On Docker Tour

www.nuodb.com/docker-tour

Read our TechBlog on Docker

www.nuodb.com/techblog/docker

About NuoDB

First envisioned by industry-renowned database architect and innovator Jim Starkey, NuoDB was developed to tackle the multiple challenges associated with cloud computing and the rise of global application deployments.

Backed by three former CEOs of the four original relational database companies, NuoDB addresses a seemingly impossible problem: Build a database suitable for mission-critical workloads – maintaining both SQL capabilities and full ACID compliance – while simultaneously delivering global access, on-demand scalability, and cloud- or container-based deployment.

In short, become the only database that can maintain transactional consistency and integrity at global scale.

NuoDB is headquartered in Cambridge, MA, USA, with offices in Dublin and Belfast. For more information, visit nuodb.com.

info@nuodb.com | +1 (617) 500-0001 | www.nuodb.com

© 2016 NuoDB, Inc., all rights reserved.