Distributed Caching Platforms

Anil K. Nori
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
1 425 703 2171

anilnori@microsoft.com

ABSTRACT

With the advances in processing, memory, and connectivity technologies, applications are becoming increasingly distributed, data-centric, and web based. These applications demand support for large number of users at high performance, requiring extreme scale, high availability, and low latency data access. Such applications are known as Extreme Transaction Processing (XTP) applications. Distributed Caching Platforms exploit the advances in the memory and networking technologies to fuse memory on multiple machines into a single unified global memory providing data access at low latencies. These Distributed Caching Platforms are evolving into in-memory data platforms for XTP applications. This tutorial provides a comprehensive overview of Distributed Caching Platform technologies and their usage.

1. The Next Generation OLTP – Extreme Transaction Processing

Most business, consumer, and entertainment data is now born and delivered in digital form. Applications are becoming more data-centric – applications are consuming more data; they access, generate and manipulate more data; application logic is becoming increasingly data-driven. The advances in processors, memory, storage, and connectivity have paved the way for such next-generation applications, whose data can reside anywhere (i.e. on the server, in the cloud, on the clients) and that support access from anywhere. This naturally leads to loosely coupled, distributed, multi-tiered, and composite application architectures. These loosely coupled applications require support for large number of users, high performance, throughput, and response time. Most web applications fall into this category. As the adoption of web technologies is rapidly increasing, so is the demand for high performance and throughput for the web applications. Similarly the acceptance of Service Oriented Architectures (SOA) in enterprises is also driving distributed and scalable systems. Additionally, the traditional OLTP applications are evolving into Extreme Transaction Processing (XTP) applications. Gartner defines XTP as an *application style aimed at supporting the design, development, deployment, management and maintenance of distributed TP applications characterized by exceptionally demanding performance, scalability, availability, security, manageability and dependability requirements*. Like OLTP applications, XTP transactions are real-time, concurrent, business-critical transactions. Being very mission critical they require very high performance, scale, availability, and security.

2. Distributed Caching Platform (DCP)

The distributed applications described above span machines and tier. Data and applications components can reside in different tiers with different semantics and access patterns. For example, data stored in the backend database is authoritative and requires high degree of data consistency and integrity. Typically, there tends to be single authoritative source for any data instance. Most data in the mid-tier, being operated by the business logic can tends to be copy of the authoritative data. Such copies are known as reference data. The reference data is suitable for caching, and thereby support extreme low latency access and scale for accesses to the data. Consider a product catalog application aggregating product information across multiple backend application and data sources. Most common operation on the catalog data is read (or browsed); a typical catalog browse operation iterates over a large amount of product data, filters it, personalizes it, and then presents the selected data to the users. Key based and query based access is a common form of operation. Caching is a critical requirement for catalog access. If not cached, operations against such an aggregate catalog require the operations to be decomposed into operations on the underlying sources, invoke the underlying operations, collect responses, and aggregate the results into cohesive responses. Accessing the large sets of backend data for every catalog operation can be prohibitively expensive, and can significantly impact the response time and throughput of the application. Caching the backend product data closer to the catalog application can significantly improve the performance and the scalability of the application. Similarly, aggregated flight schedules are another example of reference data.

Similarly transaction (or session) data, also known as activity data, is generated while the transaction is running and is typically accessible only by that transaction, until the data is committed. Such activity data is update-intensive but not shared. Consider the shopping cart data in an online buying application. There is one shopping cart, which is exclusive, for each online buying session. During the buying session, the shopping cart is updated with products purchased. The shopping cart is visible and accessible only to the buying transaction. While the buying session is active, the shopping cart is accessed both for read and write; however it is not shared. This exclusive access nature of the activity data makes it suitable for caching. To support large scalability of the buying application, the shipping carts can be distributed across the cluster of caches. Upon checkout, once the payment is applied, the shopping cart is retired (from the cache) to a backend application for further processing.

We observe that most of the OLTP data is either shared reference data or exclusive activity data. In order to provide low latency access, the reference data and activity data is cached in in-memory caching platforms. In order to provide scale and availability for the cached data, these caches are distributed and replicated. Such caches are known as Distributed Caching Platforms (DCP). In addition, to support the data access patters of the OLTP applications, the DCPs support rich DBMS capabilities.

In this tutorial we will comprehensively present the Distributed Caching Platform technologies.

3. Tutorial Outline

The tutorial includes the following key topics:

- Introduction
- Type of Cached Data
  - Reference Data
  - Activity Data
  - Resource Data
- Scenarios and Key Requirements for Extreme Transaction Processing
  - Latency
  - Scale
  - Availability
  - Consistency
- Distribute Caching Platform Concepts
  - Partitioned Caches
  - Replication Caches
  - High Availability
  - Geo-replicated Caches
  - Local Caches
- Other DCP Features
  - Concurrency
  - Durability
  - Query, Transactions
  - Geo-Replication
  - Cache Management
  - Embedded caches
- DCP Architectures
- Survey of Commercial DCPs
- Research Opportunities – Challenges
  - Large Memories, Flash drives, multi-core processors
  - Extreme low latency transaction processing
  - Cloud computing
  - Etc.