MyCQ Features

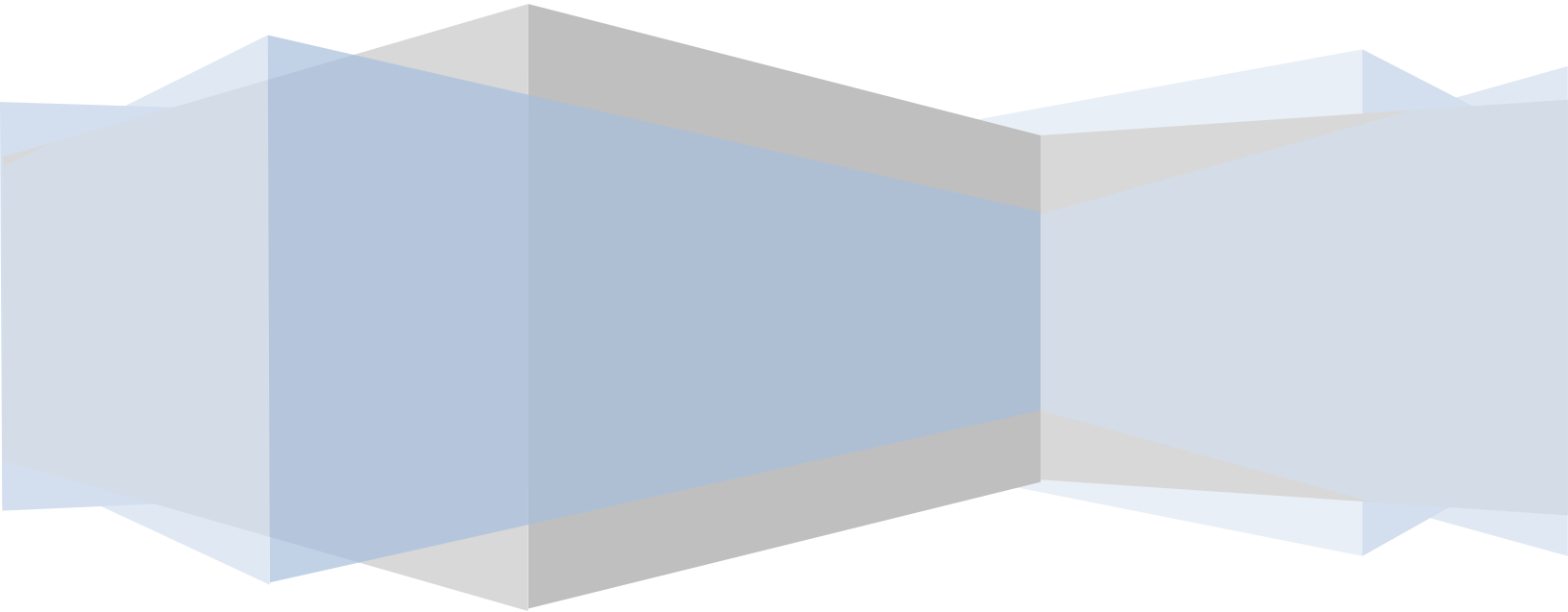


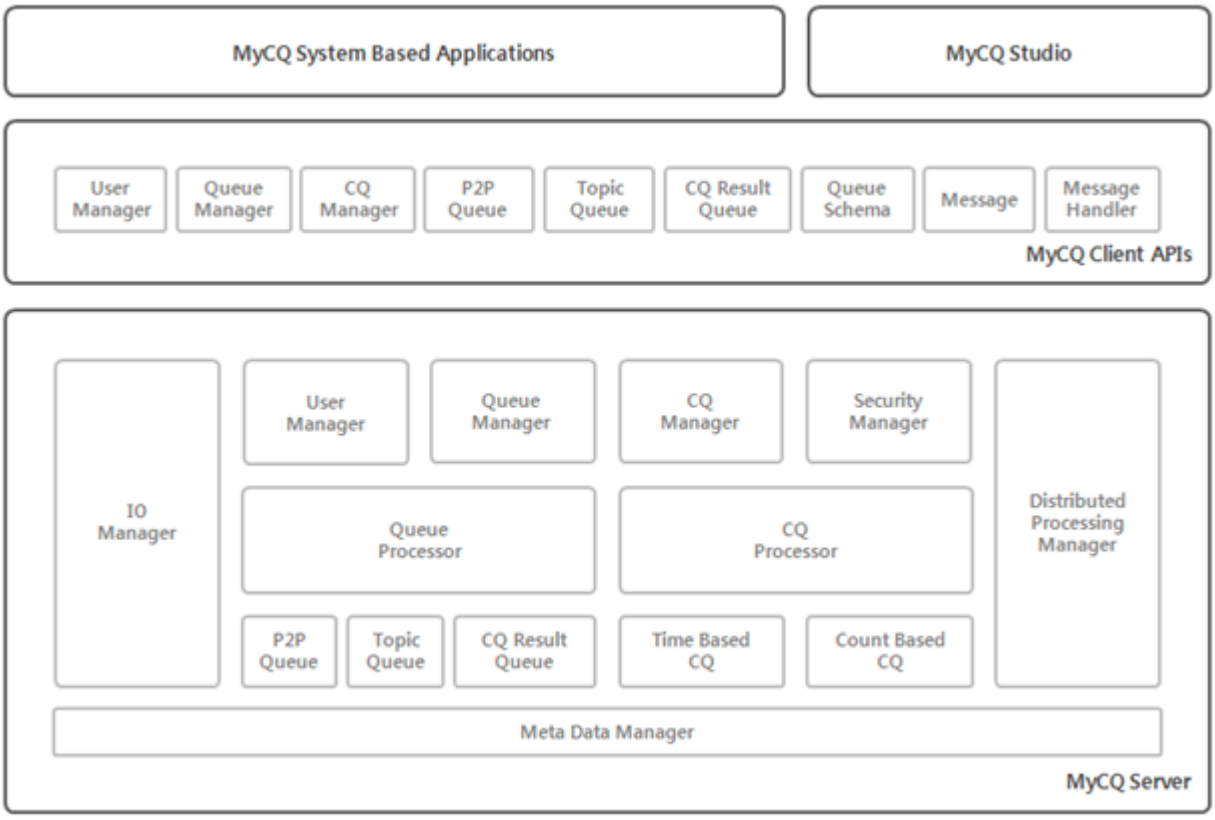
Table of Contents

Overview

MyCQ 의 시스템 구조, 주요특징, 그리고 연속질의에 대한 설명입니다.

시스템 구조

MyCQ 의 전체 시스템 구조입니다. MyCQ 는 크게 MyCQ Server 와 MyCQ Client 로 구성되어 있습니다. MyCQ Server 는 사용자 관리, 큐 관리, CQ 관리 등의 기능을 제공합니다. MyCQ Client 는 MyCQ Server 의 기능을 사용할 수 있는 API 를 제공합니다. MyCQ Studio 는 MyCQ Server 를 이용한 응용시스템 개발과 관리의 편의성을 제공합니다.



주요 특징

MyCQ 의 주요 특징에 대한 설명입니다.

메시지 전달 및 처리 기능.

- 1:1, 1:N, N:1, N:N 메시징을 위한 메시지큐 제공.
- P2P Queue, Topic Queue, CQ Result Queue 제공.

데이터 모델링 기능.

- 메시지 큐의 스키마를 정의할 수 있으며, 이를 위한 다양한 데이터 타입을 지원.

실시간 이벤트 처리 기능.

- 표준 SQL 기반으로 확장된 실시간 연속질의 언어인 MyCQL 제공.
- MyCQL 은 시간 및 개수에 따른 슬라이딩 윈도우제공.
- Aggregation, Date and time, Utility functions 제공.
- 실시간 이벤트 처리 스케쥴링 제어 기능.

고성능.

- 10K 이상의 메시지 큐 제공.
- 10K 이상의 동시 연속질의 처리 기능 제공.
- 10K 이상의 동시접속 기능 제공.

MyCQ Client API 제공

- C, C++
- .NET
- Java

MyCQ Studio

- GUI 기반 MyCQ 관리 도구.
- 템플릿 애플리케이션 제공..

실시간 이벤트 처리

MyCQ 는 연속질의 처리 기반으로 실시간 데이터 스트림을 분석하고 처리합니다. 연속질의란 질의를 한번만 등록하고, 질의의 결과가 발생할 때마다 비동기적으로 질의결과를 통보 받는 방식입니다. 이에 반해 기존 DBMS 는 매번 질의를 요청하고 질의결과를 사용자가 직접 전달받는 구조로 되어있습니다. 실시간 데이터 스트림을 분석하고 처리하기에는 MyCQ 의 데이터 처리방식이 기존 DBMS 의 데이터 처리방식에 비해 효율적입니다.

Category	MyCQ Server	DBMS
Query	Continuous	One time
Notion of time	Yes	No
Data sets of a query	Unbounded	Finite
Dealing of unreliable data	Yes	No
Reactive capability	React automatically	Mostly passive

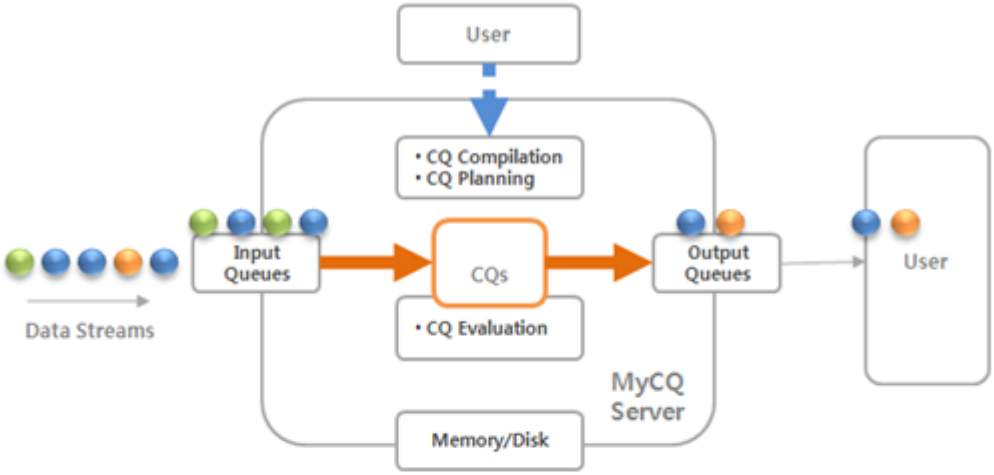
아래는 MyCQ 와 기존 DBMS 의 구조 및 동작 방법에 대한 차이를 나타낸 그림입니다.

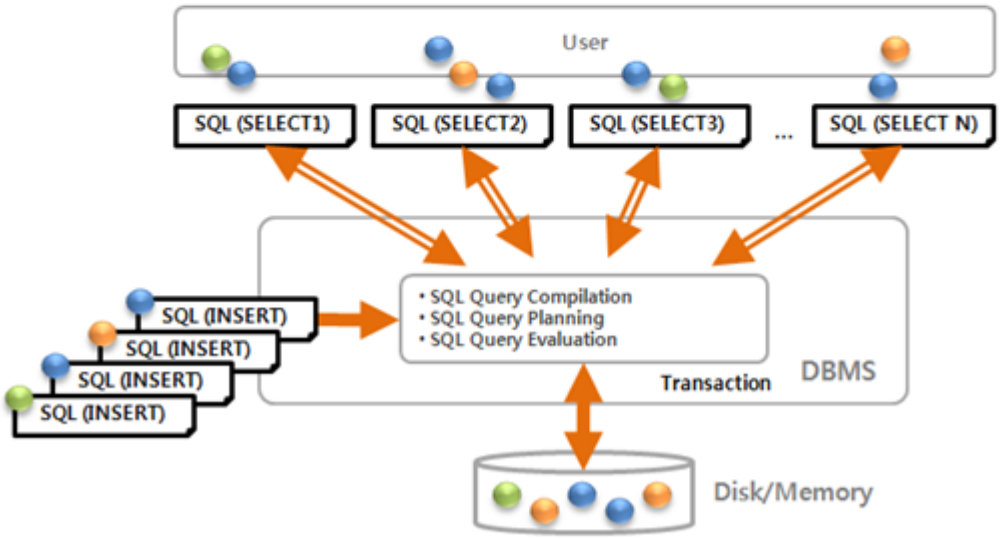
아래 그림에서 상단은 MyCQ 의 동작방법을 나타내며, 하단은 기존의 DBMS 의 동작 방법을 나타냅니다. MyCQ 는 사용자가 연속질의 등록을 요청 하면, 최초 등록시에 컴파일과, 질의 Planning 이 수행된 후, 연속질의가 등록됩니다. 연속질의 등록이 완료되면, MyCQ 는 연속적으로 등록된 질의에 대한 Evaluation 을 수행하며, 그 결과는 출력 큐로 전달됩니다.

기존의 DBMS 는 MyCQ 와는 달리 사용자가 매번 질의를 요구하여 결과를 전달받는 방식입니다. 기본적으로 사용자의 질의 요구시마다 질의 컴파일과 Planning, Evaluation 이 수행됩니다. 물론 DBMS 시스템에 따라 효율성을 높히기 위한 방법들을 적용하여 질의 처리의 효율성을 높이기는 하였지만, 근본적으로 매번 질의요구는 발생하게 됩니다.

데이터 스트림은 연속적으로 끊임없이 발생하는 특징이 있으며, 이런한 데이터를 처리하기 위한 데이터 바인딩 방법 및 질의 처리방법에 있어서, 기존 DBMS 는 기능적 한계와 성능적 한계가 있습니다. 일반적인 기존의 DBMS 는 실시간 처리 기능이 없습니다. 그러므로, 데이터 스트림을 실시간으로 바인딩하고 처리할 수 있는 방법이 없습니다. 또한, 기존의 DBMS 는 사용자가 매번 질의를 요청하는 구조이므로, 기본적으로 질의처리의 효율성이 떨어지는 것은 물론이며, 연속적으로 발생하는 데이터에 대한 실시간 질의처리 결과를 보장할 수 없습니다.

MyCQ 는 데이터 스트림 전달 및 처리를 위해 최적화된 미들웨어입니다. 이는 기존의 DBMS 를 대체하기 위한 것이 아니며, 기존의 DBMS 의 데이터 스트림 처리를 위한 기능 부족 및 성능 부족의 문제를 해결하기 위한 미들웨어라 볼 수 있습니다.



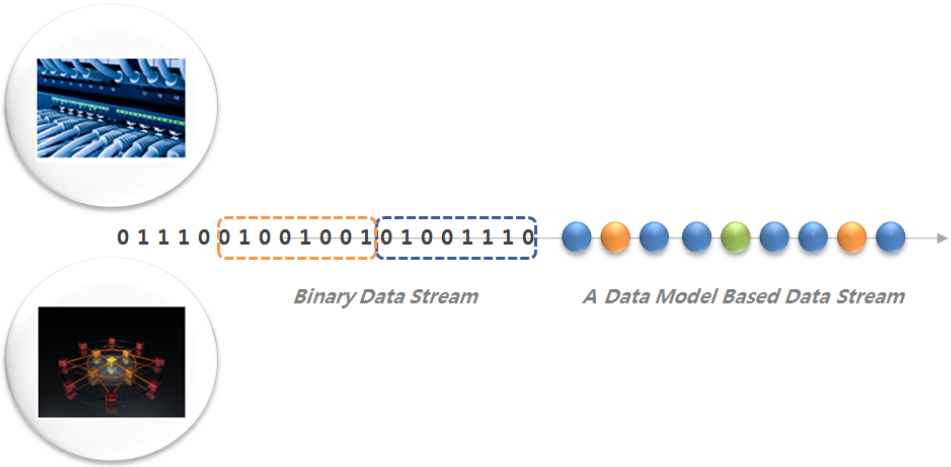


Technologies

MyCQ 의 기술적 특징에 대해 설명합니다.

데이터 추상화 & 모델링

MyCQ 는 데이터 추상화와 모델링을 위한 큐 스키마와 다양한 데이터 타입을 지원함으로써, 데이터 스트림의 전송, 분석, 처리를 위한 편리하고 효율적인 방법을 제공합니다.



MyCQ 의 큐 스키마 구조와 데이터타입

큐 스키마의 구조와, 데이터 타입에 대한 자세한 설명은 MyCQ Client API 를 참조하세요.

Queue Schema

- Column Index
- Column Name
- Column Type
- Column Size

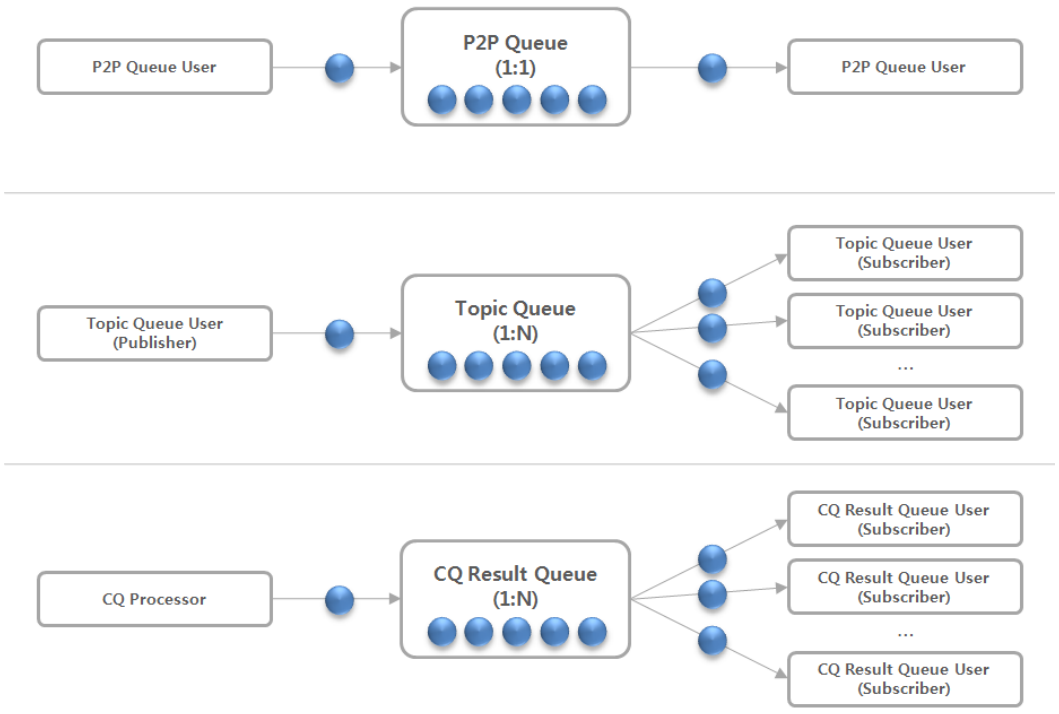
Data Types

- BOOLEAN
- BYTE
- SHORT

- USHORT
- INT
- LONG
- FLOAT
- DOUBLE
- DATE
- TIME
- DATETIME
- STRING
- VAR_STRING
- BINARY
- VAR_BINARY

메시지 전송

MyCQ 는 P2P 메시지, Publish/Subscribe 메시지 및 연속질의 결과 메시지 전송을 위해, 메시지 큐 기반의 동기/비동기 메시지 전송 기능을 제공합니다.



실시간 연속질의

MyCQ 는 실시간 데이터 스트림에 대해 이벤트를 정의하고 검출 할 수 있는 연속질의 언어 제공합니다. 보다 자세한 사항은 MyCQL Grammar 문서를 참고하시기 바랍니다.

MyCQL 특징

- Selection: Complex filtering.
- Aggregation: Complex aggregation.
- Multiplexing and de-multiplexing: decomposing and merging logical streams.
- Frequent item queries: Top-k or threshold queries.
- Stream mining: Pattern matching, similarity searching and forecasting.
- Joins: Multi-stream joins.
- Windowed queries: Time/Count based queries

다음은 MyCQL 의 예입니다.

MyCQL 의 Window 는 데이터 스트림의 시간 또는 개수에 대한 바인딩을 정의하며, Select 는 바인딩된 데이트 스트림에 대한 질의를 정의합니다.

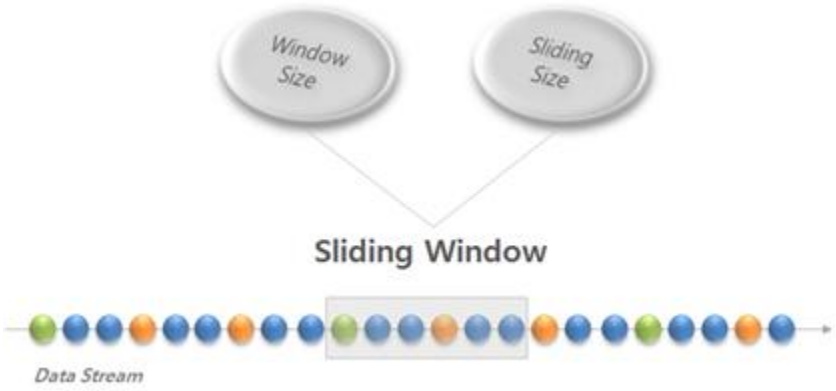
```
window
    queue1 as window1[size=5sec, slide=5sec]
select
    avg(col1)
from
    window1
```

```
window
    queue1 as window1[size= 10sec, slide=10sec]
    queue2 as window2[size= 10min, slide=5min]
select
    count(col1)
where
    window1.col1 = window2.col1 and window1.col2 > 10
from
    window1, window2
```

```
window
    queue1 as window1[size=100, slide=100]
select
    avg(col1)
from
    window1
```

다음 그림은 MyCQL 의 Sliding Window 에 대한 그림입니다.

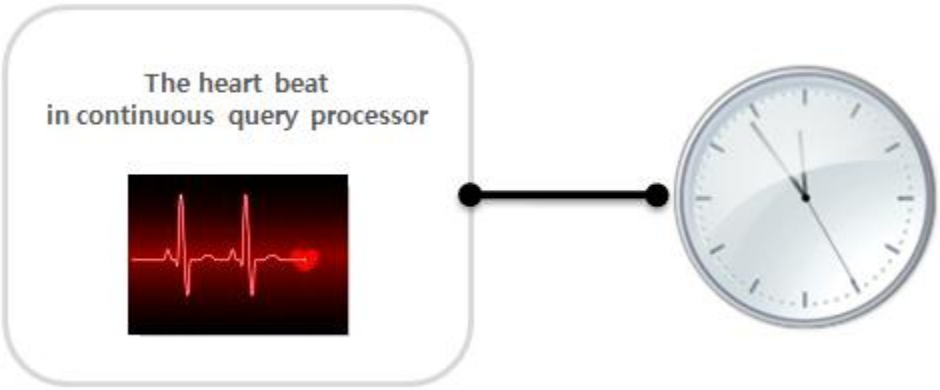
Sliding Window 는 MyCQL 의 Window 로 정의합니다. Window 는 데이터 스트림을 시간 또는 개수로 바인딩 할 수 있습니다. Window 는 크게 Normal 바인딩, Overlapping 바인딩, Jump 바인딩을 지원합니다. Normal 바인딩은 정의된 시간 또는 개수 단위로 Sliding Window 가 이동하는 것으로 데이터 스트림의 전체를 처리할 경우에 적당합니다. Overlapping 바인딩은 Sliding Window 가 중첩되며 이동하는 것으로써, 데이터 스트림을 중첩하여 처리할 경우 적당합니다. Jump 바인딩은 Sliding Window 가 건너 뛰어 이동하는 것으로써, 전체 데이터 스트림을 처리를 하지 않고 부분적으로 처리를 해야 하는 경우에 적당합니다.



Heart Beat

MyCQ 는 실시간 연속질의 스케쥴링을 컨트롤 할 수 있는 기능을 제공함으로써, 실제 시간과 연속질의 처리기의 시간을 동기화 할 수 있는 방법을 제공합니다.

Sliding Window 에 정의된 시간은 MyCQ 의 실시간 스케쥴링 시간에 따라 상대적으로 적용됩니다. 이는 MyCQ 가 설치되어 운영되는 시스템의 성능에 따라 영향을 받을 수 밖에 없습니다. MyCQ 의 Heart Beat 기능은 MyCQ 의 실시간 스케쥴링 시간을 조정할 수 있으며, 이를 이용하면, 실제 시간과 MyCQ 의 실시간 스케쥴링 시간을 동기화 할 수 있습니다.



메모리 관리

MyCQ 의 고성능 메시지 전송과, 효율적인 연속질의 처리를 가능하게 하는 중요한 기술적 특징의 하나는 최적화된 메모리 관리기법입니다.

MyCQ 의 메모리 관리 전략은 고성능에 최적화 되어있습니다. 메시지 전송 및 연속질의 처리 시 발생하는 메모리 복사를 최소화 하여 처리함으로써, MyCQ 는 제한된 메모리 공간에서도 고성능의 메시지 전송과 효율적인 연속질의 처리, 그리고 대규모의 메시지 큐 사용을 가능하게 합니다.

프로그램 용이성

MyCQ 는 최대 편리성을 위해 설계 되었습니다.

MyCQ Client 의 API 이름은 사용목적과 부합되어 이해하기 쉬우며, API 에 대한 친절한 설명과 다양한 예제를 포함하고 있습니다. 또한, 다양한 개발환경을 위한 라이브러리를 제공합니다. 개발자들은 이들을 기반으로 실시간 애플리케이션 개발을 위한 보다 높은 생산성을 기대할 수 있습니다.

MyCQ Client

- C, C++,
- .NET
- Java

MyCQ Client 라이브러리는 쉽게 애플리케이션에 포함(Embedded)될 수 있으므로, 별도의 라이브러리 설치가 필요 없으며, 애플리케이션 배포의 편리성을 제공합니다.

관리 용이성

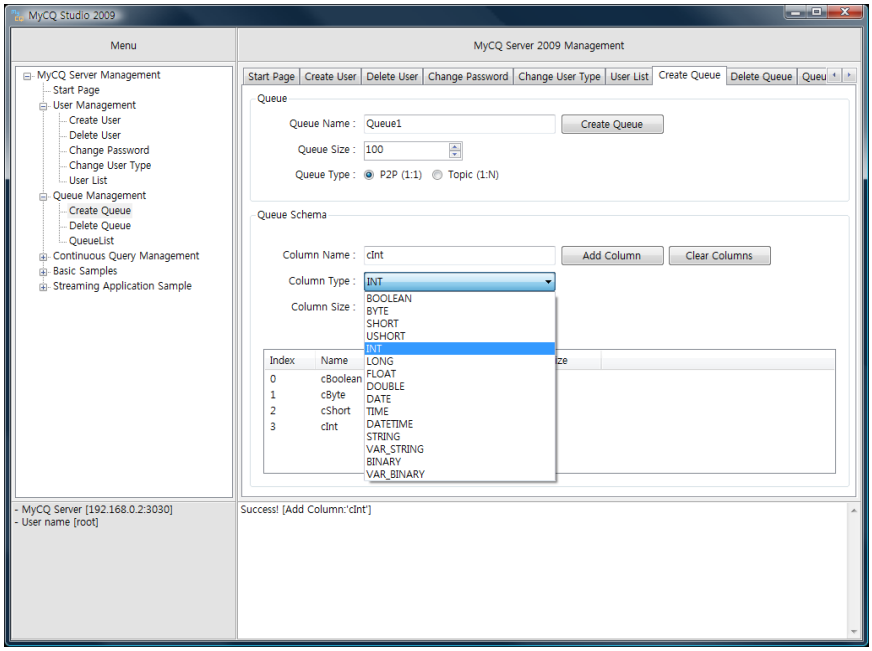
MyCQ 는 GUI 관리도구인 MyCQ Studio 를 제공합니다.

MyCQ Studio

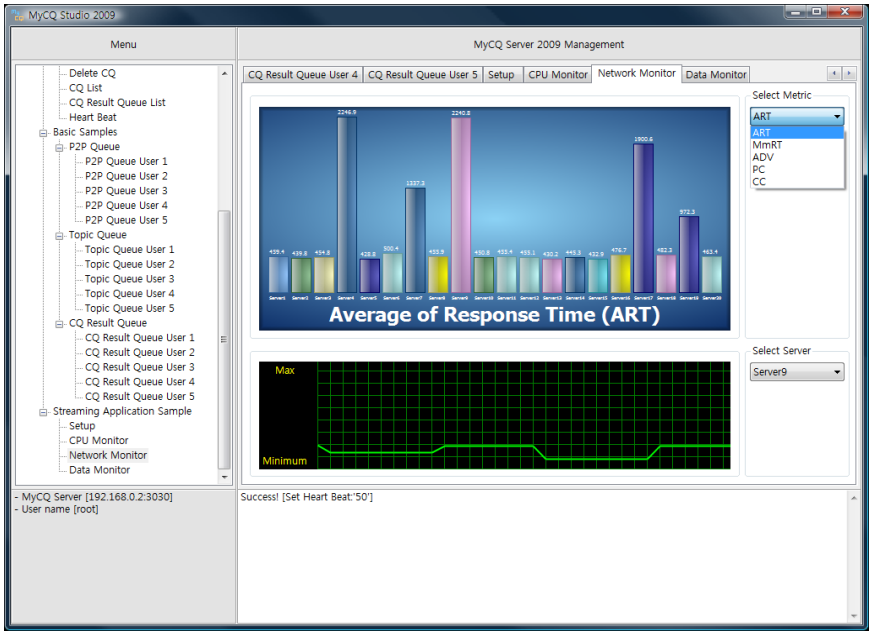
- 사용자 관리
- 큐 관리
- 연속질의 관리
- 기본 예제
- 스트리밍 애플리케이션 예제.

MyCQ Studio 는 스트리밍 애플리케이션을 쉽게 개발할 수 있도록 도와줍니다. 큐 관리, 연속질의 관리를 GUI 상에서 마우스 클릭만으로 설정하고 확인 할 수 있습니다. 또한 스트리밍 애플리케이션 샘플을 제공함으로써, 스트리밍 애플리케이션 개발에 대한 경험을 체험할 수 있습니다.

예) 큐관리



예) 샘플 1



예) 샘플 2