

한국형 e-Navigation 운영 시스템의 데이터 메시지 큐잉을 위한 AMQP 분석

장원석* · 김범준** · † 강문석

*,**,† 한화시스템 해양연구소

Analysis of the AMQP for Data Message Queueing of Korean e-Navigation Operation System

Won-Seok Jang* · Beom-Jun Kim** · † Moon-Seog Kang

*,**,† Naval R&D Center, Hanwha Systems

요 약 : 한국형 e-Navigation 운영 시스템은 해양 안전과 관련된 다양한 서비스를 제공하도록 설계되고 있다. 이 서비스들은 하나의 소프트웨어에서 제공되는 것이 아닌, 별개의 소프트웨어로 구성된다. 한국형 e-Navigation 운영시스템에 탑재되는 서비스 소프트웨어들이 통합되기 위해서는 서비스 소프트웨어간 데이터를 교환할 수 있는 데이터 메시지의 정의가 필요하다. 또한 한국형 e-Navigation 운영시스템 서비스의 동작개념에 따라 짧은 시간에 많은 메시지가 네트워크를 통해 전송될 것으로 예상된다. 이와 같은 이유로 데이터 메시지를 효율적으로 관리하기 위해서는 고빈도, 고용량 메시지를 저장하기 위한 버퍼 또는 데이터 메시지 큐가 필요하다. 이에 본 논문에서는 한국형 e-Navigation 운영 시스템에 적용될 수 있는 데이터 메시지 큐인 AMQP(Advanced Message Queuing Protocol)의 특징을 분석하고 그 결과를 보였다.

핵심용어 : e-Navigation, Message Queue, AMQP

Abstract : The Korean e-Navigation operating system is designed to provide various services related to marine safety. These services are not configured to be provided in one software, but are made in separate software. In order to integrate the service software installed in the e-Navigation operating system, it is necessary to define the data message which can exchange data between the service software. Depending on the operating concept of the service, many messages are expected to be transmitted over the network in a short time. There is a need for a buffer or message queue to store messages as a way to manage messages efficiently. Therefore, in this paper, we analyze the types and characteristics of Advanced Message Queuing Protocol(AMQP) suitable for Korean e-Navigation operating system and shows that's result.

Key words : e-Navigation, Message Queue, AMQP

1. 서 론

한국형 e-Navigation 운영시스템은 2016년부터 현재까지 연구개발을 수행하면서 해양 안전에 필요한 서비스를 도출하고 이 서비스가 적절히 서비스 될 수 있도록 설계를 고도화하고 있다(안, 2015). 현재 개발중인 한국형 e-Navigation 운영 시스템의 서비스는 “사고취약선박 모니터링”, “사고취약선박 선내지원 모니터링”, “최적안전항로 지원”, “소형선박용 전자해도 스트리밍”, “도선 및 예선 지원”, “해양 안전정보 제공”으로, 각 서비스는 별개의 소프트웨어로 개발된다. 즉, 각 서비스 소프트웨어는 한국형 e-Navigation 운영시스템으로 통합되어 동작하게 되므로 이들 서비스 소프트웨어들의 통합 및

연동을 위해서는 데이터 미들웨어가 필수적이다.

한국형 e-Navigation 시스템은 서비스간 데이터메시지의 양이 많고 이를 처리하기 위한 속도 또한 상당히 빨라야 하므로, 높은 성능의 데이터 미들웨어가 필요하다.

DDS(Data Distribution Service)는 데이터 처리속도가 높고 낮은 오버헤드를 제공하며, 높은 데이터 무결성을 제공하므로 이러한 한국형 e-Navigation 운영시스템의 요구사항에 적합한 데이터 미들웨어이다. DDS 데이터 미들웨어와 데이터 메시지는 UDP를 기반으로 높은 전송속도와 처리속도를 제공하지만 많은 연산과 데이터 처리가 필요한 서비스 소프트웨어는 이를 처리하고 못할 수 있다.

이런 이유로 타 서비스와 실시간으로 연동되지 못하고 지

† 교신저자 : 정희원, moonseog.kang@hanwha.com 054)460-8847

* 정희원, cws0714@hanwha.com 054)460-8717

** 정희원, realdever@hanwha.com 054)460-8519

연이 발생하게 되면 전체 시스템의 성능 및 신뢰성 저하를 초래할 수 있으므로, 수신된 데이터 메시지를 저장한 후 서비스 소프트웨어 내부의 각 모듈에 적시에 제공할 수 있는 버퍼 또는 메시지 큐를 적용하게 되면 데이터 처리속도의 차에 의한 악영향을 저감시킬 수 있다.

2. Advanced Message Queuing Protocol(AMQP)

Advanced Message Queuing Protocol(AMQP)는 벤더 중립적으로 설계된 메시지 큐 프로토콜로, 엔터프라이즈 비즈니스 시스템의 메시지 흐름을 관리한다. 2004년에 JPMorgan과 iMatrix사에서 개발했으며 이후 Java Messaging Service(JMS), Microsoft MQ, RabbitMQ등 다수의 AMQP 제품들이 출시되었다(Deepak, 2014). 현재 AMQP 1.0이 국제 표준으로 인증되었으며(ISO/IEC 19464) 점차 그 사용영역이 넓어지는 추세이다.

AMQP는 간단한 큐 형태부터 DDS와 유사한 Publish-Subscribe 구조와 같은 복잡한 구조까지 사용자의 필요에 따라 다양하게 사용할 수 있다.

2.1 단순 Queue 형태

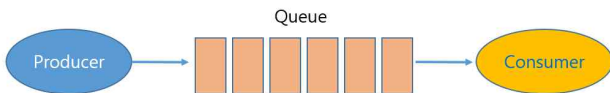


Fig. 1 Producer-Consumer Pattern of AMQP

이 형태는 일반적인 큐와 마찬가지로, 간단한 큐 형태로 사용할때의 모델이다. Producer가 데이터 메시지를 Queue에 넣으면, Consumer가 데이터를 가져가는 일반적인 Producer - Consumer 패턴이다.

2.2 단일 큐 복수 컨슈머 형태

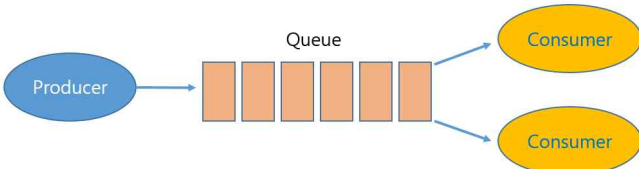


Fig. 2 Compete Consumers Pattern of AMQP

이 형태는 하나의 메시지 큐에 있는 데이터를 복수의 컨슈머가 동시에 처리하는 모델이다. 이 모델은 경쟁 소비자 패턴(Compete Consumers Pattern)으로, 여러 메시지를 동시에 처리하므로 확장성, 가용성 측면에서 이득을 볼 수 있는 패턴이

다. AMQP는 이 패턴이 가져야 하는 먹등성을 제공하므로 메시지 처리 순서가 균일할 수 있다.

2.3 Publish/Subscribe 형태

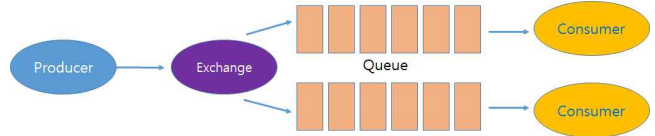


Fig. 3 Publish/Subscribe Pattern of AMQP

이 모델은 Producer가 생성한 데이터 메시지를 복수의 Consumer에게 전달하는 형태이다. 경쟁 소비자 패턴은 하나의 메시지 큐의 데이터를 처리하는 데 비해 이 형태는 Exchange를 통해 큐를 구분하여 저장하고, 각각의 컨슈머에게 전달하게 되므로 하나의 메시지가 동시에 복수의 컨슈머에게 전달되게 된다.

AMQP 이전의 메시지 큐들은 이러한 점에서 유용하지 못했으며, 플랫폼에 종속적인 경우가 많아 이기종 간 메시지 전송에 사용하기에는 메시지 브릿지를 사용하는 등 불편한 점이 많았다.

반면 이 장에서 소개한 것 이외에도 AMQP는 다양한 형태의 큐 모델을 제공하므로 사용자는 적절한 형태의 큐로 변경 또는 확장할 수 있고 향후 시스템의 설계가 다소 변경되더라도 유연하게 적용할 수 있다.

3. 한국형 e-Navigation 적용을 위한 AMQP 분석

3.1 개발 언어 측면

한국형 e-Navigation 운영 시스템을 구성하는 서비스 소프트웨어는 각기 다른 개발언어와 환경 하에서 개발되고 있기 때문에, 이러한 환경에 적용하기 위해서 우선 AMQP는 다수의 개발언어를 지원할 필요가 있다.

Table 1 Language Support of AMQP(Tomas,2013)

Language	Active MQ	Hornet Q	Qpid	Rabbit MQ	Zero Q
C	O		O	O	O
C++			O	O	O
Erlang				O	O
Java JMS	O	O	O	O	
.NET			O	O	O
Objective-C					O
PHP				O	O
Python			O	O	O
Ruby			O	O	O

Table 1에서 보인것과 같이 다양한 AMQP 제품군이 있으며 이들은 각기 지원하는 언어가 상이하다. 한국형 e-Navigation에서 주로 사용되는 언어는 C++과 Java이므로 이들을 지원할 수 있는 AMQP를 사용해야 한다.

3.2 구조적 측면

한국형 e-Navigation 운영 시스템의 서비스 소프트웨어들은 각기 다수의 컴포넌트들로 구성되어 있다. 각 서비스에 데이터를 제공해 주는 Data Service Provider(DSP)의 예를 보면, 다수의 모듈이 동시에 필요한 데이터를 처리하는 구조로 설계되어 있다.

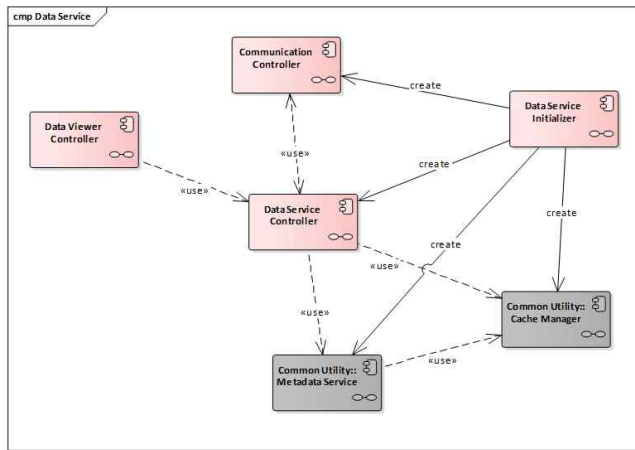


Fig. 4 Components of Data Service Provider

Fig. 4 는 DSP의 컴포넌트 다이어그램으로, Data Service Controller 컴포넌트가 각 모듈의 동작 제어를 통해 데이터 메시지의 처리를 순차적으로 수행하도록 구성되어 있는 것을 볼 수 있다. 이 구조에 AMQP의 경쟁 소비자 패턴을 적용하게 되면 순차 처리를 AMQP에 맡길 수 있으므로 보다 유연한 구조의 설계를 수행할 수 있게 된다.

3.2 성능 측면

한국형 e-Navigation 운영 시스템에서 처리해야 하는 데이터는 내부 데이터 뿐 아니라 외부의 데이터까지 존재하므로 그 양과 크기가 대단히 클 것으로 예상되고 있다. 다수의 시스템이 연동되는 e-Navigation 운영 시스템은 각 소프트웨어의 처리속도가 중요하여 하나라도 성능이 저하되게 되면 전체의 성능에 영향을 미칠 수 밖에 없다. 이 구조 하에서 데이터 메시지 처리 성능은 중요하며, 큐 없이 모든 메시지를 지연없이 처리할 수 없는, 워크로드가 큰 서비스에 적용할 경우 이득을 볼 수 있을것으로 보인다.

4. 결 론

한국형 e-Navigation 운영 시스템은 서론에서 언급한 것처럼 고용량, 고빈도의 데이터 메시지를 처리해야 하는 시스템으로, 서비스 소프트웨어 각각의 처리속도가 중요한 품질 요소 중 하나이다. 그러나 서비스 소프트웨어마다 처리해야 하는 데이터 량이나 워크로드가 다를 수 있으므로 속도의 차이가 나타날 가능성이 있다. 이때 AMQP와 같은 메시지 큐를 이용해 메시지의 누락을 방지하고 메시지 동시 처리와 같은 방식을 적용하여 데이터를 처리하면 데이터 메시지 처리 성능의 향상과 함께 타 서비스와의 연동시 실시간성, 동기성을 보장할 수 있을 것으로 보인다.

그러나 현 상태에서는 단지 AMQP의 특징과 적용 가능성을 분석했을 뿐인 단계이다. 따라서 향후의 연구에서, 본 논문에서 분석한 AMQP의 특징과 한국형 e-Navigation 운영시스템에의 적용 가능성을 바탕으로 실제 서비스 소프트웨어 중 하나를 선정하여 적용하고, AMQP의 Enqueue, Dequeue 성능과 DDS와의 연계시 성능등을 실험하고 그 결과를 보일 예정이다.

후 기

이 논문은 2017년 해양수산부 재원으로 한국해양과학기술진흥원의 지원을 받아 수행된 연구임(IMO 차세대 해양안전 종합관리체계 기술개발)

참 고 문 헌

- [1] 안광(2015), e-Navigation을 활용한 해상교통관리체계 개선방안에 관한 연구, 해양환경안전학회지 21권 2호, pp. 164-170
- [2] Tomas Bayer(2013), ActiveMQ, Qpid, HornetQ and RabbitMQ in Comparison, <https://www.predic8.com/activemq-hornetq-rabbitmq-apollo-qpidd-comparison.htm>
- [3] Deepak Arora, Varun Kumar, Prabhat Kumar Verma(2014), AMQ Protocol Based Performance Analysis of Bare Metal Hypervisors, Journal of Engineering Research and Applications, Vol 4, Issue 6(Version4), pp. 56-64
- [4] RabbitMQ. <http://www.rabbitmq.com>.