

한국형 e-Navigation 대용량 데이터 처리 플랫폼의 운용성 증대를 위한 서비스 응답 큐의 우선순위 재구성 기법

김명훈 · 강문석

한화시스템

A Method on Reconfiguring job priority in Service Queue for Enhancing Operability of e-Navigation Data Service Platform

Myeong-hun Kim · Moon-seog Kang

Naval R&D Center, Hanwha Systems

E-mail : mhkim1216@hanwha.com

요 약

본 기법은 한화시스템에서 2016년부터 착수한 한국형 e-Navigation 사업 중 대용량 데이터 처리 플랫폼(Data Service Platform, 이하 DSP)의 현장 운용성을 증대시키기 위해 DSP의 서비스 큐로 유입되는 다양한 외부 요청을 최적으로 재구성하여 끊임없는(Seamless) 서비스를 제공하는 방법이다. 항해 중인 선박이 DSP에 요청하는 다양한 서비스에 즉시 응답하기 위해서는 DSP가 응답데이터를 구성하는데 소요되는 시간과 선박이 요구하는 서비스를 사전에 예측하는 행위가 필요하다. 따라서 단기간에 DSP로 유입되는 다수의 요청들을 병렬로 예측분석하고 응답 큐를 재구성하여 응답 시간을 단축하는 기법을 구현함으로써 데이터의 크기나 서비스 동시 요청 수에 관계없이 DSP가 끊임 없는 서비스를 제공할 수 있게 하였다.

ABSTRACT

The method on reconfiguring job priority in service queue has been developed to provide seamless service and to enhance operability of Data Service Platform(DSP) in Korean e-Navigation project that performed by Ministry of Oceans and Fisheries(MOF) since 2016. It plays a critical role for providing seamless services of DSP to expect which services ships request to DSP and how much time it costs to make services for responding them to ships in advance. Therefore, as developing a method on reconfiguring jobs in service queue of DSP with a noble algorithm to analyse priority index in parallel, DSP can provide seamless services to ships regardless of data volume and the number of requests that stacked in service queue.

키워드

Seamless Service, Reconfiguring Service Queue, Data Service Platform, DSP, e-Navigation

1. 서 론

한국형 e-Navigation은 우리나라의 해상 환경에 특화된 e-Navigation으로 국제해사기구(IMO)의 e-Navigation 개념에 어선, 연안 소형선 대상 서비스 제공 등을 추가하여 우리나라 해상 환경에 최적화된 새로운 시스템을 구현한 것이다[1]. 현재 프로토타입 개발이 완료된 e-Navigation은 Data Service Platform(이하 DSP)이 데이터 센터의 역할

을 하며 PortMis, V-Pass, GICOMS의 데이터를 기반으로 항해 중인 선박에 다양한 서비스(SV10~SV52)를 제공한다[그림 1]. 여기서 데이터 서비스의 지속성을 보장하기 위해서는 DSP의 무중단 운용성을 확보하는 것이 중요한데, 이는 DSP가 다수의 선박으로부터 수신하는 다양한 서비스 요청을 처리하여 응답하는데 소요되는 시간을 최소화하는 작업이 요구된다. 본 논문에서는

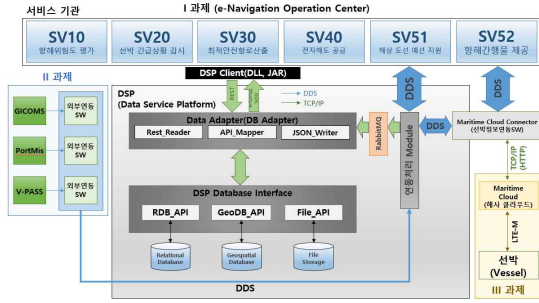


그림 1. 한국형 e-Navigation 시스템 구성도

e-Navigation의 DSP에 최적화된 서비스 큐 내부의 작업 우선순위를 최적으로 재구성하여 서비스 응답의 지연을 최소화하는 기법을 제시한다.

II. 서비스 큐 내부 Job 재구성 기법

2-1. 서비스 큐 아키텍처

DSP의 기능적 목표는 끊임없는(Seamless) 정확한 서비스를 제공하는 것이며 이를 실현하기 위해 DSP는 서비스 큐 최적화 알고리즘을 기반으로 서비스 큐 내부의 Job(선박의 요청)을 재구성한다. 현재의 서비스 큐는 FIFO의 방식을 그대로 적용하고 있으며 동시 요청이 많은 경우 DSP 내부 병목현상과 서비스 응답 대기시간이 길어지는 문제가 발생하는데, 본 연구에서 제안하는 서비스 큐 최적화 알고리즘(이하 SOA)은 DSP 분석모듈 내부 SQ Priority Calculator 세부 모듈이 서비스 큐 우선순위 지수(이하 SPI)를 산출하여 각각의 Job에 할당하고 SPI지수가 높은 순서에 따라 서비스 큐 내부 Job의 순서를 재구성한다.[그림 2].

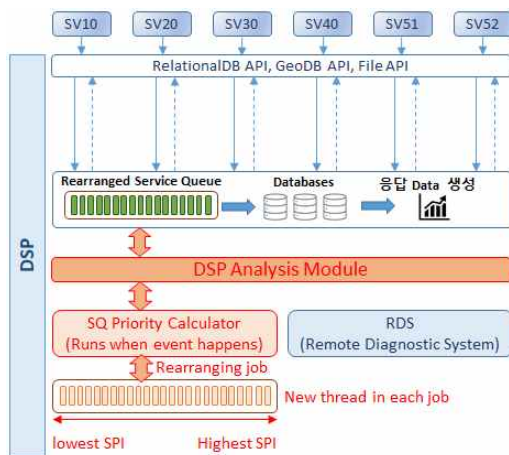


그림 2. DSP 아키텍처 및 SPI 산출 모듈

응답 큐를 재구성하여 즉시 응답할 수 있는 Job을 우선적으로 처리하여 서비스 큐의 여유를 확보하고 응답소요가 큰 응답에 리소스를 보다 더 할당하여 DSP의 병목현상을 줄이는 것이 본 연구의

주요한 기대효과이다.

2-2. 서비스 큐 최적화 기법

서비스 큐 내부 Job을 재구성하는 기준인 서비스 큐 우선순위 지수(SPI :Service queue Priority Index)는 다음과 같이 표현되며, SPI 지수가 가장 높은 Job이 1순위 dequeue 자격을 가진다.

$$SPI_i = \exists_{pool} + \log\left(\frac{wt_i \times Similarity_i}{etor_i}\right) \quad (1)$$

$$= \exists_{pool} + \log\left(\frac{wt_i \times \bar{v}_{prev} \bar{v}_{cur} \cos\theta}{etor_i}\right) \quad (2)$$

wt_i 는 서비스 큐로 유입된 Job의 대기시간(유입 후 현재까지의 ms 단위의 시간)이며 \exists_{pool} 은 동일한 요청이 과거에 존재했는지 검사하여 캐시에 해당 요청에 대한 응답이 잔존하는 경우 1, 없는 경우 0의 값을 가진다. $etor_i$ 는 Job에 대한 응답 데이터를 구성하는데 소요되는 추정 시간이며, 응답 데이터를 포함하고 있는 칼럼의 데이터 크기, 개수, 그리고 파일이 존재하는 경우 파일의 크기에 비례하여 커진다. 실제 데이터 베이스 CRUD연산과 파일 입출력에 소요된 시간이 아니라 추정된 (estimated) 시간을 의미한다.

$Similarity_i$ 는 과거의 요청과 현재의 요청의 코사인 유사도 값이며 $\bar{v} = \{co_1, co_2, co_3, \dots, co_{n-1}, co_n\}$ 는 요청 시 참조를 하는 데이터베이스의 칼럼들의 집합을 벡터로 표현한 것이다.

III. 실험 결과 및 고찰

다음 실험을 통해 SPI지수 기반의 서비스 큐 재구성 기법이 다수의 서비스 응답을 모두 완료하는데 소요되는 시간과 개별 서비스 응답소요시간을 모두 단축시킬 수 있음을 증명한다.

3-1. 임의(Random) 서비스 할당 실험

본 실험은 그림 3의 실험 환경을 기반으로 진행되었다. 임의 서비스 생성기(RSG) 모듈은 SV10 ~ SV52 각각 200개씩 총 1200개의 서비스(Job)를 생성하고 서비스 큐에 enqueue한다. DSP의 병목현상으로 인한 오류를 방지하기 위하여 0.1초 간격으로 Job을 enqueue하며, DSP의 SQ Priority Calculator는 서비스 큐로 유입되는 Job의 SPI 지수를 산출하는 쓰레드를 순서대로 생성하여 계산 후 Job을 크기 순으로 정렬한다. 위의 과정은 완전한 Queue 자료구조를 가지는 기존의 DSP와 1초마다 서비스 큐를 재정렬하는 본 연구에서 제안하는 SOA 알고리즘이 적용된 DSP에 각각 100회씩 수행하여 아래 두 가지 값의 평균값을 확인한다.

- 서비스 큐의 모든 Job에 대한 응답소요시간
- 개별 Job에 대한 응답소요시간

첫 번째 모든 Job에 대한 응답소요시간은 큐에 첫 번째 Job이 유입된 시점부터 마지막 Job이 처리 완료된(Dequeue) 시점까지의 시간이며, 개별 Job에 대한 응답소요시간은 해당 Job이 큐에 유입된 시점부터 처리 완료된 시점까지의 시간을 의미한다.

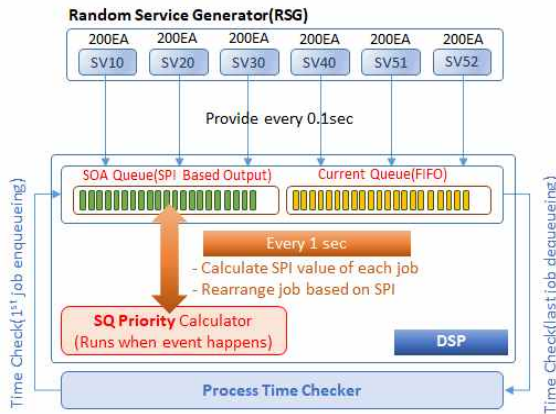


그림 3. 임의 서비스 할당 실험 구성도

3-2. 실험 결과의 요약
표 1과 같이 전체 Job과 개별 Job에 대한 응답소요 시간을 각각 21.97%P, 19.67%P 단축시킬 수 있었다.

표 1. SOA 알고리즘 적용 전후의 결과 비교

	토탈 Job Processing		개별 Job Processing	
	SOA 적용전	SOA 적용후	SOA 적용전	SOA 적용후
값	755.68sec	589.64sec	0.61sec	0.49sec
개선	-21.97%P		-19.67%pP	

IV. 결론

본 연구에서 제시하는 SOA 기법이 DSP의 Job을 보다 더 효율적으로 처리할 수 있는 방법임을 증명하였다.

References

[1] 한국형 e-Navigation 사업 [internet]. Available: http://www.smartnav.org/html/SMA-RT-Navigation_New/about_smart_navigation.php

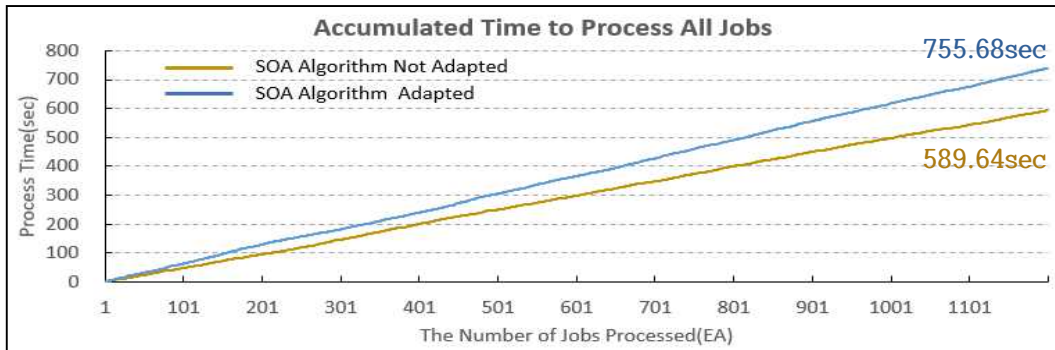


그림 4. 모든 Job에 대한 응답소요시간 측정 비교

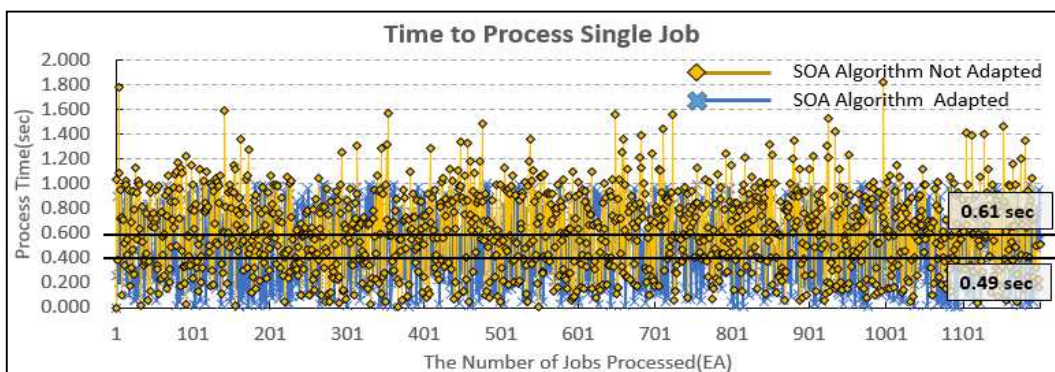


그림 5. 개별 Job에 대한 응답소요시간 비교