논문 2021-16-34

BigCrawler: 엣지 서버 컴퓨팅·스토리지 모듈의 동적 구성을 통한 효율적인 빅데이터 처리 시스템 구현 및 성능 분석

(Implementation and Performance Aanalysis of Efficient Big Data Processing System Through Dynamic Configuration of Edge Server Computing and Storage Modules)

김 용 연, 전 재 호, 강 성 주*
(Yongyeon Kim, Jaeho Jeon, Sungjoo Kang)

Abstract : Edge Computing enables real-time big data processing by performing computing close to the physical location of the user or data source. However, in an edge computing environment, various situations that affect big data processing performance may occur depending on temporary service requirements or changes of physical resources in the field. In this paper, we proposed a BigCrawler system that dynamically configures the computing module and storage module according to the big data collection status and computing resource usage status in the edge computing environment. And the feature of big data processing workload according to the arrangement of computing module and storage module were analyzed.

Keywords : Bigdata Analytics, Edge Computing Server, Data Crawling, Kubernetes, Elastic Search

Ⅰ. 서 론

산업현장이 점차 디지털화되면서 생산 설비와 운영 프로세스, 기업용 시스템 등으로부터 다양하고 방대한 양의 데이터가 수집되고 관리되고 있다. 이러한 빅데이터에 대한 분석을 통해 기존에 관찰하지 못했던 흐름이나 추세의 파악, 이상 현상 감지와 미래 예측 등이 가능하며, 이를 통해 산업 환경에 잠재된 위험에 대응하거나 생산 라인의 효율적인 운영과 제품의 불량률 감소 등을 통한 경제적 효과를 거둘 수 있다 [1]. 하지만 기업이 자체적으로 빅데이터 역량을 갖추기 위해서는 빅데이터 분석과 저장을 위한 컴퓨팅·스토리지 자원과 이를 운영·활용하기 위한 인력이 필요하며, 상용 클라우드를 활용하더라도 기업 민감 데이터의 외부에 저장에 따른 보안 문제와 산업현장과 클라우드 간의 네트워크 지연 문제 등이 발생한다 [2].

엣지 컴퓨팅은 사용자 또는 데이터 수집원 (data source)의 물리적인 위치와 가까운 곳에서 컴퓨팅을 수행하는 방법으로서 산업 현장과 밀착된 엣지 컴퓨팅 환경에서 빅데이터를 처리하면 보안과 지연 문제를 해소할 수 있다 [3]. 하지만 엣지 컴퓨팅 환경은 클라우드 컴퓨팅 환경에 비해 시스템 자원, 전력, 주변 환경 등 운영 조건이 제한적이다. 구체적으로, 엣지 컴퓨팅 환경에 배치되는 엣지 서버 클러스터의 규격은 대체로 단일 서버나 4U 수준의 클러스터 시스템이며 [4-6], 이처럼 제한된 서버 시스템 규격에 임의의 수요 기업의 빅데이터 처리 요구사항에 맞도록 컴퓨팅·스토리지 자원을 효율적으로 구성하는 것은 도전적인 영역이며 최근 이와 관련한 연구들이 수행되고 있다.

Kim은 컨테이너 기반의 공개 소스 가상화 플랫폼인 도커 기반의 빅데이터 관리 플랫폼 개발을 수행하였지만, 빅데이터의 일반적인 처리 단계인 실시간 수집, 처리 환경에만 초점을 맞추어 다양한 요구사항이 반영되는 엣지 환경에서 필요한 스토리지와 연산 노드의 동적 구성은 반영되지 않았다 [7]. Kim은 엣지 컴퓨팅 환경에서 저지연 서비스를 제공하기 위한 Krane 스케줄링을 수행하여 네트워크 연결을 안정적으로 제공하는 실험을 진행하였으나, 노드의 저지연만을 목표로 실험을 수행하여 수집된 빅데이터를 저장, 관리하기 위한 스토리지 자원에 대한 역할은 고려되지 않았다 [8]. Shin은 빠른 데이터 처리 및 응답을 가능하게 하는 스마트공장용 빅데이터 수집·저장·처리와 빅데이터 인프라의 통합구조를 제시하였다. 하지만 정적인 제조·생산 현장에서만 적용 가능한 구조이기 때문에 실시간 의사결정에 따라 빅데이터 처리 요구사항이 변하는 엣지 컴퓨팅 환경에서 활용하기 위해 추가적인 설계가 필요하다 [9].

본 논문에서는 산업현장의 엣지 컴퓨팅 환경에서 활용될 엣지 서버 기반의 빅데이터 처리 시스템인 BigCrawler의 설계 제안과 구현 및 성능 분석 내용을 다룬다. 제안하는

*Corresponding Author (sjkang@etri.re.kr)
Received: Oct. 22, 2021, Revised: Nov. 30, 2021, Accepted: Dec. 9, 2021.
J. Jeon: ETRI (Senior Researcher)
Y. Kim: ETRI (Senior Researcher)
S. Kang: ETRI (Principal Researcher/Project Leader)
※ 이 논문은 2021년도 정부 (과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2020-0-00844, 엣지 서버 시스템 자원 관리 및 제어를 위한 경량 시스템 소프트웨어 기술 개발).

ⓒ IEMEK J. Embed. Sys. Appl. 2021 Dec. 16(6) 259-266
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2021.16.6.259

BigCrawler 시스템은 크게 두 가지의 특화된 관리 기능을 제공한다. 첫 번째로, 임의의 규격을 갖는 엣지 서버를 대상으로 컴퓨팅 기능에 특화된 모듈과 스토리지 기능에 특화된 모듈 간의 구성 변경이 가능하다. 이를 통해 수요 기업의 빅데이터 처리 요구사항에 따른 서버 모듈 (노드) 구성과 관련 빅데이터 솔루션 구성이 가능하다. 본 논문에서는 4-모듈 규격의 엣지 서버 클러스터를 대상으로 컴퓨팅 모듈과 스토리지 모듈의 구성을 {3:1}, {1:3}으로 변경하며 빅데이터 처리 워크로드에 따른 특성을 관찰하였다. 두 번째는 산업현장 데이터의 수집을 위한 크롤러의 동적 배치 기능이다. 산업현장으로부터 직접적으로 데이터를 수집하는 과정인 크롤링은 높은 수준의 컴퓨팅 연산을 필요로하는 과정으로서, BigCrawler는 산업현장의 빅데이터 수집 상황 및 컴퓨팅 자원 사용 현황에 따라 크롤러 응용 프로그램을 컴퓨팅·스토리지 모듈에 동적으로 배치할 수 있다. 본 논문에서는 엣지 서버 클러스터를 구성하는 컴퓨팅 모듈과 스토리지 모듈에 크롤러 배치를 다양하게 변화하면서 빅데이터 처리 워크로드에 따른 특성을 관찰하였다.

본 논문의 내용은 다음과 같이 구성되어 있다. 2장에서는 본 연구를 위한 배경 설명으로서 BigCrawler를 구성하는 모듈을 제공하기 위한 쿠버네티스 (Kubernetes), 일래스틱 서치 (Elasticsearch) 기술과 빅데이터를 수집, 분류하는 크롤러 기술, 실험 결과 측정을 위한 쿠버네티스 클러스터의 자원 모니터링에 대하여 설명한다. 3장에서는 BigCrawler 시스템의 설계를 제안한다. 4장에서는 구현 시스템의 다양한 구성에 따른 실험 사례와 결과를 비교하고 5장에서 결론을 맺는다.

Ⅱ. 배경 기술

1. 쿠버네티스

쿠버네티스 [10]는 여러 개의 서버로 구성된 클러스터에 컨테이너를 배포하고 운영하면서 서비스 간 연결을 쉽게 해주고, 다양한 컨테이너의 확장 및 관리를 자동화하기 위한 컨테이너 오케스트레이션 플랫폼이다. 쿠버네티스는 그림 1과 같이 클러스터의 관리 역할을 수행하는 마스터 노드 (master node)와 실제 응용이 실행되는 워커 노드 (worker node)로 구성된다. 클러스터 내에서는 컨테이너의 집합으로 이루어진 파드 (Pod)를 응용 최소 단위로 관리하며, 마스터 노드는 리소스 요구사항과 클러스터의 상태를 확인하여 파드의 실행 요청 시 해당 파드를 실행할 수 있는 최적의 워커 노드 선택과 배포를 수행한다.

2. 일래스틱 서치

일래스틱 서치는 루씬 (Lucene) [11] 기반 분산 검색엔진으로 정형, 비정형, 반정형 등 다양한 유형의 데이터들을 검색하고 결합할 수 있다. 그리고 쿠버네티스와 연동하여 노드를 추가하면 클러스터가 인식할 수 있도록 설계되었고, 데이터가 분산 저장되고 노드별로 병렬 처리되어 빅데이터 환경에서 빠른 검색 및 분석이 가능하다. 그리고 그림 2와 같이 로그스태시 (Logstash)라는 이름의 데이터 수집 및 로그 파싱 엔진, 그리고 키바나 (Kibana)라는 이름의 분석 및 시각화 플랫폼과 함께 일래스틱-로그스태시-키바나 (ELK) 스택이라는 솔루션으로 제공되고 있다 [12].

3. Crawler

크롤러는 다양한 데이터를 수집해서 분류하고 스토리지에 저장하는 응용이다. 크롤링 데이터는 데이터 생성 스타일에 따라, 정형 (구조화된 데이터), 반정형 (로그데이터) 그리고 비정형 (SNS, 이미지, 동영상 등) 데이터로 구분된다. 디지털화된 산업현장에서 수집되는 빅데이터를 소비자의 요구에 맞게 재생산하기 위해서는 크롤링 기술이 필요하다. 스크래핑이 단순히 분석을 위한 특정 데이터를 추출하는 것이라면 크롤링은 그림 3과 같이 자동화된 크롤러가 여러 웹페이지를 목적에 맞게 수집 및 분류하고 찾아낸 데이터를 저장한 후 쉽게 찾을 수 있게 인덱싱하는 작업을 포함한다 [13]. 하지만 산업현장의 다양하고 방대한 디지털 정보를 무분별하게 크롤링하는 것은 서버에 부담을 주고 그에 대한 비용을 지불해야 하므로 효율적인 크롤링 기술이 필요하다.

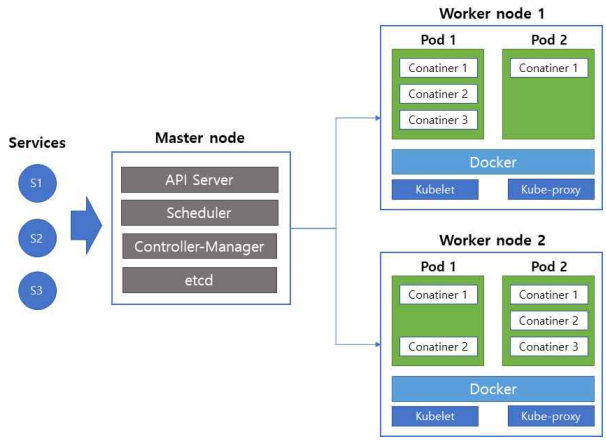


그림 1. 쿠버네티스 구조
Fig. 1. Kubernetes architecture

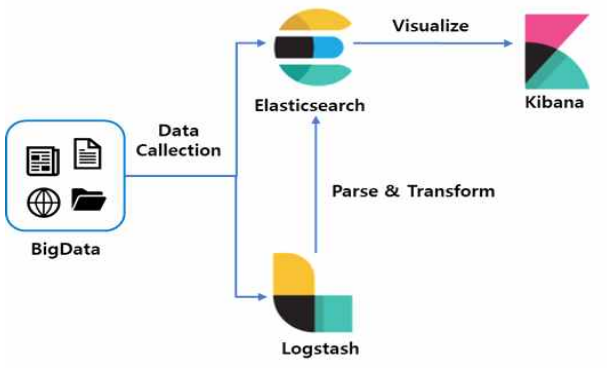


그림 2. 일래스틱 스택 구조
Fig. 2. Elastic stack architecture

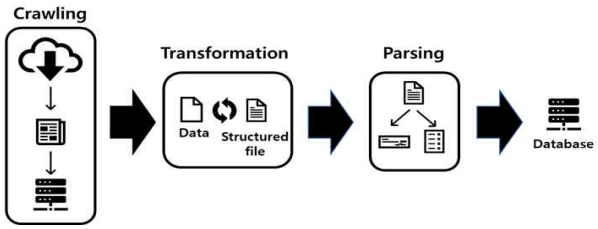


그림 3. 크롤러 구조
Fig. 3. Crawler architecture

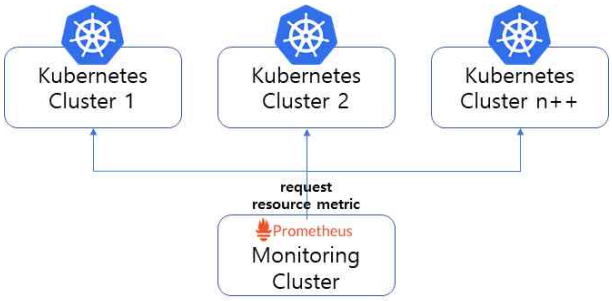


그림 4. 쿠버네티스 모니터링 구조
Fig. 4. Kubernetes monitoring architecture

4. 자원 모니터링

쿠버네티스 클러스터 내부에서 다양한 자원들과 서비스들이 구성되었을 때 조합 간의 성능을 비교하기 위하여 서비스들이 사용하는 자원 사용률에 대한 모니터링이 필요하다. 그림 4와 같이 쿠버네티스의 자원 모니터링은 외부 컴포넌트를 사용하여 수행한다. 쿠버네티스 외부 컴포넌트를 이용한 모니터링은 대표적으로 프로메테우스 (Prometheus)가 있다 [14]. 프로메테우스를 사용하여 클러스터에서 사용하는 컴퓨팅 및 스토리지 자원을 시계열 데이터로 수집하여 모니터링 할 수 있다.

Ⅲ. BigCrawler

1. BigCrawler 개요

본 논문에서 제안하는 BigCrawler는 산업 현장 근처의 엣지 컴퓨팅 환경에서 활용될 엣지 서버 클러스터 기반의 빅데이터 처리 시스템이다. 그림 5와 같이 주요 상용 클라우드 [15, 16]가 제공하는 데이터 획득 (Data Ingestion), 데이터 저장 (Data Store), 데이터 처리 (Data Process), 데이터 가시화 (Data Visualization) 등 주요 빅데이터 처리 단계를 4-모듈 규격의 엣지 서버 클러스터가 제공한다. BigCrawler는 산업 현장의 빅데이터 처리 형태에 따라 서버 모듈과 구성 요소의 재구성이 가능하다. 빅데이터의 처리는 현장의 실제 요구사항에 따라 그림 4의 빅데이터 처리 단계 중 세부 단계의 비중 차이에 따라 구성이 달라진다. 예를 들어 임의의 산업 현장 A는 방대한 데이터의 수집과 주기적인 빅데이터 처리/분석을 수행하는 반면, 또 다른 산업 현장 B는 수집되는 데이터를 실시간으로 분석하며 이상 상황에 신속하게 대응하는 형태를 보일 수 있다. 이 경우 산업 현장 A는 데이터 획득과 처리 비중이 높고, 산업 현장 B는 산업 현장 A에 비해 데이터 처리의 비중이 상대적으로 높을 것으로 예상할 수 있다.

그림 5. 빅데이터 처리 단계
Fig. 5. Bigdata Processing Stages

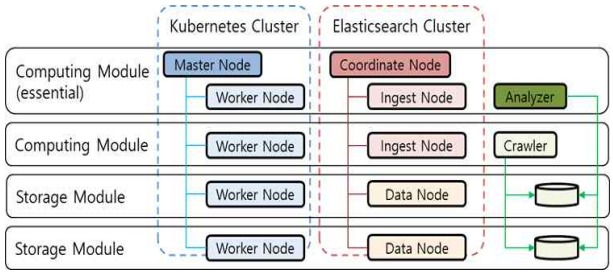


그림 6. BigCrawler의 구성 모듈과 클러스터 별 역할
(컴퓨팅 모듈과 스토리지 모듈이 2:2의 경우)
Fig. 6. BigCrawler's configuration modules and cluster-specific roles

2. BigCrawler의 모듈 구성

BigCrawler는 엣지 서버 클러스터를 구성하는 서버 모듈을 컴퓨팅 모듈과 스토리지 모듈로 구분한다. 컴퓨팅 모듈은 데이터 획득, 분산 저장, 데이터 처리와 사용자 인터페이스 제공과 같이 연산 집약적인 작업 수행에 최적화된 모듈이며, 스토리지 모듈은 대용량의 저장소와 상대적으로 적은 연산 능력을 통해 빅데이터 저장과 컴퓨팅 모듈로부터의 데이터 분석 요청에 대해 일부 처리 기능을 갖춘 모듈이다. BigCrawler는 현장의 빅데이터 처리 형태와 활용 계획에 따라 엣지 서버 클러스터에 하드웨어 노드 구성 시 역할에 맞는 솔루션을 설치 및 최적화하는 기능을 제공한다. 하드웨어 노드에 구성되는 솔루션은 앞 장에서 설명한 쿠버네티스와 일래스틱 서치이며, 모듈의 역할에 따라 쿠버네티스 클러스터 상에서의 역할과 일래스틱 서치 클러스터 상에서의 역할이 정해진다.

그림 6은 이에 대한 구체적인 설명을 나타내는 것으로서, 컴퓨팅 모듈과 스토리지 모듈이 각각 2대씩 구성된 경우를 보여준다. BigCrawler는 최소 1대 이상의 컴퓨팅 모듈을 가져야하며, 이는 쿠버네티스 클러스터 상에서 마스터 노드 역할을 하며, 일래스틱 서치 클러스터 상에서 코디네이트 노드 (coordinate node) 역할을 한다. 마스터 노드를 제외한 나머지 노드들은 쿠버네티스 클러스터 상에서 워커 노드 (worker node) 역할을 하며, 컴퓨팅 모듈 또는 스토리지 모듈이 될 수 있다. 컴퓨팅 모듈 중 마스터 노드가 아닌 모듈은 일래스틱 서치 클러스터 상에서 인제스트 노드 (ingest node)가 되어 데이터 획득을 담당하고, 스토리지 모듈은 일

래스틱 서치 클러스터 상에서 데이터 노드 (data node)가 되어 데이터 저장을 담당하며, 코디네이트 노드의 요청에 대응하여 관련 데이터 수집을 수행한다.

3. BigCrawler의 동적 성능 최적화

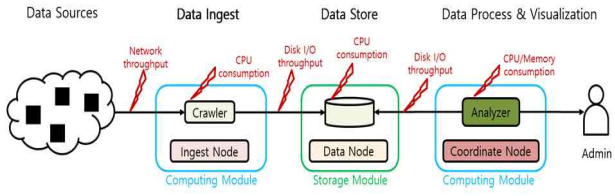
일반적인 상황에서 BigCrawler를 통해 빅데이터 처리 단계가 진행되는 동작 흐름은 그림 7-(a)와 같이 도시된다. 컴퓨팅 모듈의 인제스트 노드는 크롤러를 통해 데이터 수집원으로부터 데이터를 획득해 스토리지 모듈의 데이터 노드에 분산 저장한다. 산업 현장의 빅데이터 관리자가 필수 컴퓨팅 모듈 (그림 2의 코디네이트 노드)을 통해 빅데이터 분석 요청을 하면 분석기는 데이터 노드로부터 데이터를 수집해, 가공하고, 분석한 결과를 관리자에게 제공한다.

동작 흐름에 따라 나타나는 주요 성능지표 발생 시점은 다음과 같다. 먼저 크롤러가 빅데이터를 획득할 때 연산 자원과 네트워크 자원 소모가 발생하고 크롤러가 데이터 노드에 데이터를 분산 저장할 때 연산 자원 소모와 스토리지 입출력이 발생한다. 데이터 처리/분석 시에는 데이터 노드에서의 연산 자원 사용과 스토리지 입출력이 발생하고, 수집된 데이터를 분석/가공 및 가시화할 때는 연산 자원과 메모리 자원의 사용이 주로 발생한다.

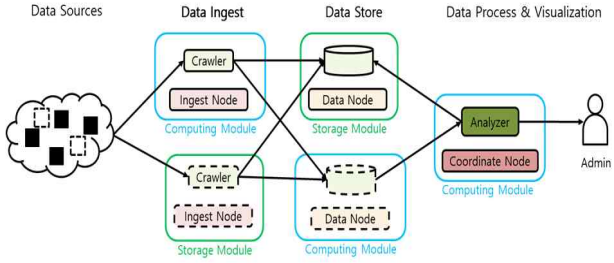
하지만 엣지 컴퓨팅 환경에서는 현장의 일시적인 요구사항이나 물리적 변화에 따라 빅데이터 처리 성능에 영향을 주는 다양한 상황이 발생할 수 있다. 제품 생산 현장의 예를 들면, 갑작스러운 주문이 증가하면 생산 장비의 가동이 늘어나게 되고, 이는 수집 대상 데이터의 증가로 인해 데이터 획득을 수행하는 컴퓨팅 모듈의 과부하로 연결된다. 또한 동시다발적인 이상 데이터가 발생하면 생산 장비의 실행 중단을 막기 위해 실시간으로 이상 데이터를 처리해야 한다. 이를 위해 데이터 획득과 저장하는 모듈을 줄이는 대신 이상 데이터를 분석하는 모듈의 수를 증가시켜야 하는 상황도 고려할 수 있다.

BigCrawler는 쿠버네티스의 컨테이너 오케스트레이션 기능을 통해 엣지 클러스터 자원 상황에 따라 운영 중인 크롤러의 확장과 재배치 기능을 제공한다. 일반적인 상황에서는 연산 집약적인 작업을 수행하는 크롤러를 컴퓨팅 노드에 배치하지만, 위의 사례와 같이 수집 대상 데이터의 증가로 컴퓨팅 모듈에 과부하가 발생하는 상황에서는 그림 7-(b)와 같이 스토리지 모듈에도 인제스트 노드 역할을 부여하여, 컨테이너화된 크롤러를 스토리지 모듈에 동적으로 배치하고, 이를 통해 전체 엣지 서버 클러스터의 데이터 수집 능력을 향상할 수 있다. 또한 데이터 분석 모듈을 늘려야 하는 상황에서는 컴퓨팅 모듈에 데이터 노드 역할을 부여하여, 전체 엣지 서버 클러스터에서 데이터 저장 비중을 줄이고 컴퓨팅 비중을 늘려서 데이터 분석 작업 능력을 향상시킬 수 있다.

이어지는 4장에서는 BigCrawler의 컴퓨팅 모듈과 스토리지 모듈 간의 다양한 구성을 수행하여 컴퓨팅 모듈과 스토리지 모듈의 구성에 따라 발생하는 주요 성능 측정 지점에의 변화를 측정, 비교하고, 스토리지 모듈의 크롤러 실행



(a)



(b)

그림 7. BigCrawler의 동작 흐름
(a) 일반적인 상황에서의 동작 흐름과 주요 성능 측정 지점
(b) 갑작스러운 상황에 대응하기 위한 동적 성능 최적화
Fig. 7. BigCrawler Workflow
(a) Workflow under normal circumstances and key performance measurement points
(b) Dynamic performance optimization to respond to sudden situations

여부에 따른 컴퓨팅/데이터 성능 변화를 측정, 비교하여 BigCrawler의 효용성을 살펴본다.

Ⅳ. 실 험

1. 실험환경

본 실험에서는 표 1과 같이 두 가지 종류 (컴퓨팅 모듈, 스토리지 모듈)의 모듈이 4U를 구성해 하나의 엣지 서버 클러스터를 이룰 때, 서버 모듈의 구성과 크롤러의 배치 변경에 따른 성능 차이를 비교하였다. 이를 통해 실제 BigCrawler 적용 현장에서 빅데이터 처리 규모와 특성에 따라 적절한 모듈 구성과 최적화 전략에 활용할 수 있는 실험 데이터를 도출하는 것을 목표로 하였다.

표 1. 클러스터 구성과 연산·스토리지 종류
Table 1. Cluster configuration and operation/storage types

Module	Processor	Storage
Computing	CPU: 11th Gen Intel(R) Core(RM) i7 @ 2.80GHz	SSD: 10GB
Storage	CPU: 64bit ARM Cortex-A53 @ 1GHz	NVMe: 256GB

표 2. 수집 대상 센서 데이터의 필드
Table 2. Fields of Collected Sensor Data

Name	Description	Type
device_id	Sensor device's ID	string
facility_name	The name of the facility with the sensor attached.	string
item_name	Sensor data item	string
item_value	Sensing data value	float
timestamp	Sensor data generation time	string

표 3. 실험 시나리오 구성
Table 3. Experimental Scenario

Scenario No.	Computing Module	Storage Module	No. of Crawlers
1	1 (Coord. & Ingest node)	3 (Data node)	1
2	1 (Coord. & Ingest node)	1 (Data node)	3
	2 (Data & Ingest node)		
3	1 (Coord. & Ingest node)	3 (Data & Ingest node)	4
4	1 (Coord. & Ingest node)	1 (Data & Ingest node)	4
	2 (Data & Ingest node)		

컴퓨팅 모듈은 스토리지 모듈에 비해 연산 속도에서 장점이 있으나 저장 장치의 용량이나 전력 소모량에서는 스토리지 모듈에 강점이 있다. 본 실험에서 BigCrawler의 수집 대상 데이터로 사용한 '전력 설비 에너지 패턴 및 고장 분석 센서' [17] 데이터는 총 24GB의 용량을 가지고 있다. 센서 데이터는 10가지 종류, 총 461대의 설비로부터 수집되었으며, 총 22개의 필드로 구조화되어 저장되어 있고, 하나의 데이터 크기는 383KB이다. 대표적으로 많이 쓰이는 필드는 표 2와 같다.

수집 대상 센서 데이터는 BigCrawler가 설치된 엣지 서버 클러스터 외부에 저장되어 있다. BigCrawler에는 최소 하나 이상의 컴퓨팅 모듈이 구성되며, 컴퓨팅 모듈에는 크롤러가 포함되어, 외부 장치로부터 지속해서 센서 데이터를 수집해 스토리지 모듈에 저장한다. 빅데이터 분석과 가시화를 위한 응용 프로그램은 크롤러를 이용하여 센서 데이터를 수집하고 일래스틱서치로 모든 필드를 검색하여 출력하는 키바나 기반의 모니터링 응용을 수행하였다.

그리고 쿠버네티스의 자원 모니터링 도구인 프로메테우스를 사용하여 실험 시나리오에 따른 클러스터 자원의 성능지표를 수집하였다.

2. 실험 시나리오 구성

본 실험은 총 4개의 모듈 구성을 표 3과 같이 네 가지 시나리오로 구성되었다. 시나리오 1은 4개의 모듈 중 하나의 컴퓨팅 모듈과 3개의 스토리지 모듈로 구성하였고 컴퓨팅 모듈에 하나의 크롤러가 데이터를 수집하는 기본 구성이며, 시나리오 2는 3개의 스토리지 모듈 중 2개의 스토리지 모듈 대신 2대의 컴퓨팅 모듈로 구성하고, 총 3개의 크롤러가 동작하는 구성이다. 시나리오 3과 시나리오 4에서는 앞선 시나리오에서 크롤러가 없는 스토리지 모듈에 크롤러를 배포하여 단위 시간 데이터 수집을 늘리는 구성을 수행하여 실험을 수행하였다.

각 시나리오별로 20분간 실험을 통해 아래의 성능지표를 측정하였다.

- CPU 점유율: 프로세서 코어의 총 작업 시간 중 실제로 데이터를 처리하는 데 사용되는 비율로 큰 값을 가질수록 실행 중인 프로그램의 컴퓨팅이 증가하고 있다는 것을 알 수 있다, 컴퓨팅 모듈의 CPU 점유율이 낮을 경우 데이터 노드로 지정해 데이터 저장 및 분산 처리에 활용할 수 있으며, 스토리지 모듈의 CPU 점유율이 낮을 경우 인제스트 노드로 지정해 크롤러를 추가로 운용할 수 있다.
- 초당 입출력 처리량 (IOPS): 저장 장치의 초당 데이터의 쓰기/읽기 처리량으로 클러스터의 빅데이터 처리 규모를 파악할 수 있다.
- 초당 네트워크 처리량 (Network Throughput): 초당 데이터 수집원과 클러스터 간 패킷 처리량으로서 클러스터의 모든 노드에서 주고받는 (Input/Output) 데이터 획득 규모를 파악할 수 있다.

3. 실험 결과

3.1. 시나리오 1

시나리오 1은 컴퓨팅 모듈이 1대이고, 크롤러가 배포되지 않은 스토리지 모듈 3대로 구성된 기본 설정 클러스터로서 구성 모듈에 대한 성능 측정 결과는 표 4와 같다.

데이터 수집원으로부터 전체 클러스터로 전송된 데이터는 초당 250개이며, 이는 1개의 크롤러에서 획득되어, 3대의 스토리지 모듈로 분산 저장되었다. 이후 1대의 컴퓨팅 모듈에서 빅데이터 분석 요청이 발생하면, 3대의 스토리지 모듈로부터 데이터 수집 및 가공이 진행된다. 이를 처리하기 위해 컴퓨팅 모듈에서는 1개의 크롤러 실행과 분석기 실행을 위해 평균 8.5% 정도의 CPU 자원을 활용하였으며, 1대의 컴퓨팅 모듈과 3대의 스토리지 모듈 간 네트워크 처리량은 초당 75 MB/s였다. 또한 3대의 스토리지 모듈에서는 데이터 저장과 빅데이터 분석 요청에 대한 처리를 위해 대당 평균 13.8% 정도의 CPU 활용과 500 IOPS의 입출력 처리량이 발생하였다.

표 4. 실험 결과 (시나리오 1)
Table 4. Experiment Result (Scenario 1)

	CPU(%)	IOPS(IOP/s)	Throughput(MB/s)
1 Computing Module	8.5	500	15
3 Storage Module	13.8	1,500	75

표 5. 실험 결과 (시나리오 2)
Table 5. Experiment Result (Scenario 2)

	CPU(%)	IOPS(IOP/s)	Throughput(MB/s)
3 Computing Module	8	2,250	99.9
1 Storage Module	24	1,600	7.5

표 6. 실험 결과 (시나리오 3)
Table 6. Experiment Result (Scenario 3)

	CPU(%)	IOPS(IOP/s)	Throughput(MB/s)
1 Computing Module	13.7	750	25
3 Storage Module	14.1	3,000	34.8

표 7. 실험 결과 (시나리오 4)
Table 7. Experiment Result (Scenario 4)

	CPU(%)	IOPS(IOP/s)	Throughput(MB/s)
3 Computing Module	8	2,700	99
1 Storage Module	28	2,000	6.7

3.2. 시나리오 2

시나리오 2는 컴퓨팅 모듈이 3대이고 크롤러가 배포되지 않는 스토리지 모듈 1개로 구성되어 있고, 성능 측정 결과는 표 5와 같다. 시나리오 1의 스토리지 모듈 2대 대신 추가된 컴퓨팅 모듈 2대에는 크롤러가 배포되어 있고, 데이터 노드로 설정되어있어 분산 빅데이터 수집 및 분석 과정에 참여한다.

시나리오 1과 비교하면 크롤러의 개수가 1개에서 3개로 증가했기 때문에 데이터 수집원으로부터 단위 시간당 획득하는 데이터의 양이 750개로 증가하였고, 이에 따라 컴퓨팅 모듈의 네트워크 처리량이 총 6.6배 정도 증가하였다. 또한 데이터 노드로 설정된 컴퓨팅 모듈에서 자체 저장소에 데이터를 일부 저장하므로, 스토리지 모듈의 대당 네트워크 처리량은 70% 감소하였고, 컴퓨팅 모듈의 IOPS는 대당 50% 증가하였으나, CPU 점유율은 큰 차이가 없었다. 스토리지 모듈은 3대를 사용한 시나리오 1에 비해 CPU 점유율이 1.73배 증가하였지만, 24% 정도의 CPU 점유율은 부담되는 수준이 아니다. 이는 컴퓨팅 모듈이 늘어남에 따라 클러스터 전체의 데이터 획득량이 늘어났으나, 시나리오 2의 구성으로 처리하는 데 문제는 없다는 것을 의미한다.

3.3. 시나리오 3

시나리오 3은 컴퓨팅 모듈이 1개이지만, 수집 대상 데이터의 증가에 대비해 크롤러가 배포된 스토리지 모듈이 3개로 구성되며 성능 측정 결과는 표 6과 같다.

시나리오 3은 시나리오 1과 비교해서 크롤러가 1개에서 4개로 증가하였으므로, 단위 시간당 획득하는 데이터의 양이

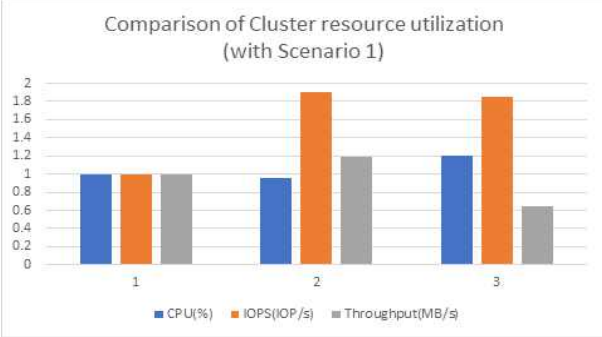


그림 8. 클러스터 자원 사용률 비교 (시나리오 1과 비교)
Fig. 8. Comparison of Cluster resource utilization (with senario 1)

총 1,000개로 증가하였고, 크롤러가 늘어남에 따라 컴퓨팅 모듈은 모두 CPU 점유율, IOPS, 네트워크 처리량이 모두 증가하였다. 이에 반해 스토리지 모듈은 CPU 점유율, IOPS는 증가하였으나 네트워크 처리량은 20.2MB/s 감소하였다.

3.4. 시나리오 4

시나리오 4는 컴퓨팅 모듈 3대와 크롤러가 배포된 스토리지 모듈 1대로 구성되어 있으며, 클러스터에서의 성능 측정 결과는 표 7과 같다. 시나리오 3의 스토리지 모듈 대신 추가된 컴퓨팅 모듈 2개에는 크롤러가 배포되어 있고, 데이터 노드로 설정되어있어 분산 빅데이터 처리 과정에 참여한다. 또한 1대의 스토리지 모듈에도 크롤러가 배포되어 있다.

시나리오 2와 비교해서 스토리지 모듈에 크롤러가 배포되어 동작함으로써 CPU 점유율이 4% 상승하였고, 수집 데이터가 증가함에 따라 IOPS가 400 IOP/s 정도 증가하였으며 네트워크 처리량은 1.2MB/s 감소하였다.

3.5. 시나리오 간 성능 비교

그림 8은 시나리오 1 대비 시나리오 2와 시나리오 3의 클러스터 자원 사용률을 비교한 그래프이다. 시나리오 1과 시나리오 2를 비교해보면 CPU 점유율은 4% 줄어들면서 IOPS는 90%, 네트워크 사용률은 19% 증가함을 볼 수 있다. 이는 성능이 좋은 컴퓨팅 모듈이 늘어남에 따라 전체 클러스터의 CPU 점유율이 줄어들면서 데이터 처리 성능이 높아짐을 알 수 있다. 그리고 3대의 컴퓨팅 모듈이 하나의 데이터 모듈에 데이터를 전송하기 때문에 네트워크 사용률은 19% 증가하였다. 이는 수집해야 하는 데이터가 적으면서 연산 처리를 실시간으로 수행해야 하는 시스템에 적합하다.

시나리오 1과 시나리오 3을 비교해보면 CPU 점유율은 20% 증가하면서 IOPS는 85% 증가하였고, 네트워크 사용률은 36% 감소함을 알 수 있었다. 이는 스토리지 모듈이 크롤링을 수행함으로써 스토리지 모듈의 CPU 점유율과 IOPS가 증가하였음을 알 수 있었고, 스토리지 모듈에 데이터가 저장됨으로써 클러스터 간 네트워크 사용률이 줄어듦을 확인할 수 있었다. 이는 연산 처리가 조금 늦어지더라도 수집해야 하는 데이터가 많아지는 시스템에 적합하다.

그림 9는 시나리오 2 대비 시나리오 4의 클러스터 자원 사용률을 비교한 그래프이다. 시나리오 2와 시나리오 4를 비교해보면 스토리지 모듈에 크롤러가 하나 추가되면서

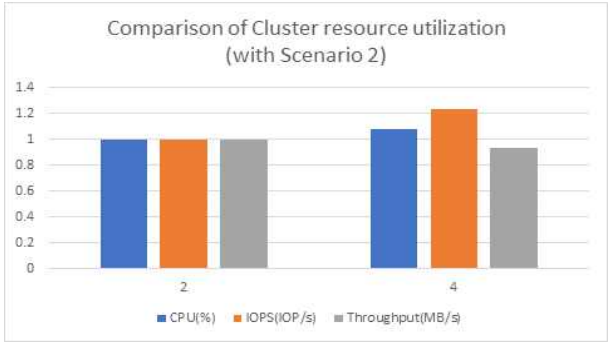


그림 9. 클러스터 자원 사용률 비교 (시나리오 2와 비교)
Fig. 9. Comparison of Cluster resource utilization (with senario 2)

CPU 점유율은 8% 증가하였는데 이는 스토리지 모듈에서만 증가한 사용률이다. 그리고 크롤링을 수행함으로 인해 IOPS는 24% 증가하였으며 크롤링 데이터의 일부가 스토리지 모듈에 저장됨으로써 데이터 전송량이 줄어들어 네트워크 사용률은 7% 감소함을 알 수 있었다. 이는 수집해야 하는 휘발성 데이터가 많아지면서 연산 처리를 실시간으로 수행해야 하는 시스템에 적합하다.

위의 특성을 바탕으로 구성한 실 엣지 환경에서의 BigCrawler의 동적 모듈 구성 시나리오는 다음과 같다.

기본 동작 방식인 시나리오 1을 수행 중 동시다발적인 이상 데이터 발생 시 실시간 처리 데이터가 늘어남을 감지하고 시나리오 2로 변환하여 데이터 수집과 저장을 줄이고, 이상 데이터 분석에 참여하는 컴퓨팅 모듈의 수를 증가시켜 데이터 처리의 연속성을 유지하고, 수집하는 데이터가 급작스럽게 늘어남을 감지하면 시나리오 3, 4로 변환하여 스토리지 모듈에도 크롤러를 배포하여 수집 대상 데이터의 증가로 인한 컴퓨팅 모듈의 과부하를 방지할 수 있으며, 수집하는 데이터의 특성에 따라 대용량의 영속성 데이터를 저장하는 경우 시나리오 3의 구성이 적합하고, 대용량의 휘발성 데이터를 처리할 때는 시나리오 4의 구성이 적합하다.

V. 결 론

본 논문의 BigCrawler 시스템은 엣지 서버를 대상으로 컴퓨팅 기능에 특화된 모듈과 스토리지 기능에 특화된 모듈 간의 구성을 변경할 수 있게 하고 빅데이터 수집을 위한 크롤러의 동적배치를 통해 산업현장의 빅데이터 수집 상황 및 컴퓨팅 자원 사용 현황에 따라 컴퓨팅 모듈과 스토리지 모듈을 최적 배치 할 수 있다. 이를 통해 다양한 빅데이터 수집 요구사항이 필요한 산업현장에서 한정된 자원으로 구성된 엣지 컴퓨팅 클러스터의 효율적인 사용이 가능하다.

이를 확인하기 위하여 다양한 엣지 컴퓨팅 서버 노드 구성에 따른 빅데이터 처리 워크로드 특성을 관찰하였다. 엣지 컴퓨팅 클러스터에서 컴퓨팅 모듈의 수가 많을 때는 빅데이터 분석과 데이터 크롤링이 분산되어 컴퓨팅 비용이 줄어드는 대신 빅데이터 전송량이 늘어나는 특징을 가지고 있었고, 스토리지 모듈의 수가 많을 때는 빅데이터를 분산 관리하여 스토리지 사용률이 올라가는 대신 빅데이터 전송량이 줄어드는 특징을 가지고 있었다. 그리고 크롤링 기능을 스토리지 모듈도 같이 수행했을 때 데이터 수집 및 분석 기능을 분산 수행하여 CPU와 스토리지 자원 사용률은 증가하지만, 빅데이터 전송량이 줄어드는 특징을 확인할 수 있었다.

향후 연구에서는 본 논문에서 스토리지 모듈의 장점인 데이터 공간 활용률뿐 아니라 스토리지 모듈이 사용하는 HW인 arm 서버와 서버에서 사용하는 NVMe 스토리지의 장점인 전력 대비 성능비도 고려한 성능 평가가 필요하다. 또한 본 논문의 연구를 바탕으로 다양한 산업현장의 빅데이터 처리 서비스들의 자원 요구사항 및 자원 간 연관 관계 등을 분석하고 이를 반영한 엣지 컴퓨팅 서버 노드 최적 배치 알고리즘에 관한 연구가 필요하다.

References

[1] J. Wang, W. Zhang, Y. Shi, S. Duan, J. Liu, "Industrial Big Data Analytics: Challenges, Methodologies, and Applications", CoRR, arXiv:1807.01016, 2018
[2] B. Duncan, M. Whittington, V. Chang, "Enterprise security and privacy: Why adding IoT and big data makes it so much more difficult," International Conference on Engineering and Technology (ICET), 2017, pp. 1-7
[3] M. Caprolu, R. Di Pietro, F. Lombardi and S. Raponi, "Edge Computing Perspectives: Architectures, Technologies, and Open Security Issues," 2019 IEEE International Conference on Edge Computing (EDGE), 2019, pp. 116-123
[4] "Technology Roadmap of SME", 2018-2020, https://www.smtech.go.kr
[5] Gigabyte, "Edge Server", 2021
[6] Dell, "Dell EMC PowerEdge XE Servers", 2021
[7] D. Kim, Y. Park, T. Chung, "Development of Big-data Management Platform Considering Docker Based Real Time Data Connecting and Processing Environments," IEMEK J. Embed. Sys. Appl., Vol. 16, No. 4, pp. 153-161, June, 2021 (in Korean).
[8] T. Kim, T, Kim, S. Jin, "Multi-access Edge Computing Scheduler for Low Latency Services," IEMEK J. Embed. Sys. Appl., Vol. 15, No. 6, pp. 299-305, December, 2020 (in Korean).
[9] S. J. Shin, J. Woo, W. Seo, "Developing a Big Data Analytics Platform Architecture for Smart Factory," Journal of Korea Multimedia Society, Vol. 19, No. 8, pp. 1516 - 1529, Aug. 2016 (in Korean).
[10] Kubernetes, https://kubernetes.io/
[11] Apache Lucene, https://lucene.apache.org/

[12] Elastic(ELK) Stack, https://www.elastic.co/
[13] J. William, "Web Data Crawling vs Web Data Scraping", Promptcloud, https://www.promptcloud.com/blog/data-scraping-vs-data-crawling.
[14] Prometheus, https://prometheus.io/
[15] Analytics end-to-end with Azure Synapse, https://docs.microsoft.com/en-us/azure/architecture/example-scenario/dataplate2e/data-platform-end-to-end
[16] AWS serverless data analytics pipeline reference architecture, https://aws.amazon.com/ko/blogs/big-data/aws-serverless-data-analytics-pipeline-reference-architecture/
[17] "Power facility energy pattern and failure analysis sensor," https://aihub.or.kr/aidata/30759 (in Korean).

Yongyeon Kim (김 용 연)

2008 Computer Engineering from Chungnam National University (B.S.)
2010 Computer Engineering from Chungnam National University (M.S.)
2010~Electronics and Telecommunications Research Institute (Senior Researcher) researcher.
Fields of interests: Embedded systems and operations, System, communication middleware, artificial intelligence SW platform
Email: yyeon@etri.re.kr

Jaeho Jeon (전 재 호)

2008 SW Engineering from Auburn University (B.S.E)
2010 SW Engineering from Auburn University (M.S.E)
2010~Electronics and Telecommunications Research Institute (ETRI), a senior researcher
Fields of interests: Edge Computing, CPS, AI Platform
Email: jeonjaeho11@etri.re.kr

Sungjoo Kang (강 성 주)

2003 Electronic Engineering from Hanyang University (B.E.)
2005 Electronic Engineering from Hanyang University (M.E.)
2011 Software Engineering from Korea Advanced Institute of Science and Technology (KAIST) and Carnegie Mellon University, respectively, (M.S.E)
2018 Computer Engineering from Chungnam National University (Ph.D.)
Field of Interests: Cyber-Physical Systems, Edge Computing, Multimodal Interactions in Metaverse, Autonomous Systems, Digital Twin, Blockchain and Cryptocurrency
Email: sjkang@etri.re.kr