ISSN : 1226-7244 (Print)
ISSN : 2288-243X (Online)
j.inst.Korean.electr.electron.eng.Vol.28,No.3,458~464,September 2024
논문번호 24-03-31
https://doi.org/10.7471/ikeee.2024.28.3.458
216

동적 이기종 캐시 서버 네트워크에서의 내결함성 캐싱 시스템 설계 및 구현

Design and Implementation of a Fault-Tolerant Caching System for Dynamic Heterogeneous Cache Server Networks

김 현 기*, 함 규 식*, 김 진 우*, 장 수 영*, 최 창 범*★

Hyeon-Gi Kim*, Gyu-Sik Ham*, Jin-Woo Kim*, Soo-Young Jang* and Chang-Beom Choi*★

Abstract

This study proposes a fault-tolerant caching system to address the issue of caching content imbalance caused by the dynamic departure and participation of cache servers in a heterogeneous cache server network, and validates it in both real and virtual environments. With the increase of large-scale media content requiring various types and resolutions, the necessity of cache servers as key components to reduce response time to user requests and alleviate network load has been growing. In particular, research on heterogeneous cache server networks utilizing edge computing and low-power devices has been actively conducted recently. However, in such environments, the irregular departure and participation of cache servers can occur frequently, leading to content imbalance among the cache servers deployed in the network, which can degrade the performance of the cache server network. The fault-tolerant caching algorithm proposed in this study ensures stable service quality by maintaining balance among media contents even when cache servers depart. Experimental results confirmed that the proposed algorithm effectively maintains content distribution despite the departure of cache servers. Additionally, we built a network composed of seven heterogeneous cache servers to verify the practicality of the proposed caching system and demonstrated its performance and scalability through a large-scale cache server network in a virtual environment.

요 약

본 연구는 이기종 캐시 서버 네트워크에서 캐시 서버의 동적 이탈과 참여로 인한 캐싱 콘텐츠 불균형 문제를 해결하기 위해 내결함성 캐싱 시스템을 제안하고, 이를 실제 및 가상 환경에서 검증하였다. 다양한 유형과 해상도가 요구되는 대용량 미디어 콘텐츠의 증가로 인해, 캐시 서버는 사용자 요청에 대한 응답 시간을 단축하고 네트워크 부하를 줄이는 핵심 요소로서 필요성이 증대되고 있다. 특히, 최근 엣지 컴퓨팅 및 저전력 장치를 활용한 이기종 캐시 서버 네트워크에 관한 연구가 활발히 진행되고 있다. 그러나 이러한 환경에서는 캐시 서버의 불규칙한 이탈과 참여가 빈번하게 발생할 수 있으며, 이로 인해 캐시 서버 네트워크에 배포되어 있는 콘텐츠 간 불균형이 발생하여 캐시 서버 네트워크의 성능을 저하키실 수 있다. 본 연구에서 제안한 내결함성 캐싱 알고리즘은 캐시 서버 이탈 시에도 미디어 콘텐츠 간 균형을 유지함으로써 안정적인 서비스 품질을 보장한다. 실험 결과, 제안된 알고리즘은 캐시 서버의 이탈에도 불구하고 콘텐츠 분포를 효과적으로 유지함을 확인하였다. 또한, 실제 7대의 이기종 캐시 서버로 구성된 네트워크를 구축하여 제안 캐싱 시스템의 실용성을 검증하였고, 가상 환경에서 대규모 캐시 서버 네트워크를 통해 시스템의 성능 및 확장성을 입증하였다.

Key words : Heterogeneous Cache Server Network, Edge Computing, Fault-Tolerant Caching System, Resource-Aware Caching, Content Management Algorithm

* Dept, of Computer Engineering, Hanbat National University(Researcher, Professor)
★ Corresponding author
E-mail : cbchoi@hanbat.ac.kr, Tel : +82-42-821-1144

※ Acknowledgment
This results was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-004)
Manuscript received Sep. 19, 2024; revised Sept. 24, 2024; accepted Sep. 30, 2024.
This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/licenses/by-nc/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ⅰ. 서론

캐시 서버는 대규모 데이터 전송에서 중요한 역할을 담당하는 요소로, 사용자 요청에 대한 응답 시간을 단축하고 네트워크 및 원본 서버의 부하를 줄이는 데 사용된다. 특히, 다양한 유형과 해상도가 요구되는 대용량 미디어 콘텐츠의 증가로 인해, 캐시 서버는 사용자 요청에 대한 응답 시간을 단축하고 네트워크 부하를 줄이는 핵심 요소로서 필요성이 증대되고 있다. 캐시 서버는 요청되는 데이터를 임시로 저장하여, 사용자가 동일한 콘텐츠에 반복적으로 접근할 때마다 원본 서버에 요청을 보내는 대신 캐시 서버에서 빠르게 응답할 수 있도록 한다. 이를 통해 전체 시스템 성능과 네트워크의 효율성을 높일 수 있다.

최근 증가하는 미디어 콘텐츠 수요와 사용자 기기의 다양성에 대응하기 위해, 엣지 컴퓨팅과 저전력 장치를 활용한 이기종 캐시 서버 네트워크에 관한 연구가 활발히 진행되고 있다. 이기종 캐시 서버 네트워크는 다양한 특성과 성능을 가진 다양한 서버들이 하나의 네트워크에서 협력하여 캐싱 기능을 수행하는 구조를 의미한다. 여기서 “이기종”은 저장 용량, 처리 능력, 네트워크 속도 등이 서로 다른 장치들이 함께 운영되고 있음을 의미하며, 특히 네트워크 엣지에서 다양한 사용자 기기의 요청에 신속하게 응답하기 위해 캐싱 작업을 수행한다. 예를 들어, 라즈베리파이와 같은 저전력 장치가 캐시 서버로 사용되며, 이를 통해 컴퓨팅 자원과 네트워크 자원의 효율성을 극대화하는 다양한 시도가 이루어지고 있다[1][2][3][4]. 그러나 이러한 환경에서는 캐시 서버의 불규칙한 참여와 이탈이 빈번하게 발생할 수 있으며, 이 때문에 캐시 서버 네트워크에 배포된 콘텐츠 간 불균형이 발생하여 캐시 서버 네트워크의 성능이 저하될 수 있다. 캐시 서버 이탈은 특히 중요한 문제로, 특정 서버에 저장된 콘텐츠가 갑작스럽게 소실될 경우 나머지 캐시 서버에 과도한 부하가 가중되거나 사용자에게 제공되는 서비스 품질이 저하될 수 있다. 따라서 캐시 서버의 이탈 상황에서도 미디어 콘텐츠 간 균형을 유지할 수 있는 메커니즘이 필요하다. 이는 네트워크 대역폭, 서버 간 자원의 불균형을 고려한 효율적인 캐싱 알고리즘을 통해 가능하다.

본 논문은 이기종 캐시 서버 네트워크에서 캐시 서버 이탈 시에도 미디어 콘텐츠 간 균형을 유지함으로써 안정적인 서비스 품질을 보장하는 내결함성 캐싱 시스템을 제안한다. 내결함성이란 시스템의 일부 구성 요소에 장애가 발생해도 시스템이 계속 정상적으로 동작할 수 있는 능력을 말하며, 서비스에 장애가 발생하더라도 외부적으로는 아무 문제가 없는 것처럼 보이거나, 신속히 복구되는 특성을 말한다[5]. 제안된 캐싱 시스템은 각 캐시 서버의 컴퓨팅 자원, 네트워크 상태, 그리고 미디어 콘텐츠의 중요도를 종합적으로 고려하여, 이탈한 캐시 서버의 역할을 다른 서버가 신속히 대체할 수 있도록 설계되었다. 이를 통해 캐시 서버 네트워크 전반의 성능 저하를 최소화하고 시스템의 안정성을 높이는 것을 목표로 한다. 실제 7대의 이기종 기기를 기반으로 캐시 서버 네트워크를 구축하여 제안된 캐싱 시스템의 실용성을 검증하였으며, 가상 환경에서 대규모 캐시 서버 네트워크를 통해 시스템의 성능 및 확장성을 입증하였다. 이를 통해 이기종 캐시 서버 기반의 네트워크에서 캐싱 효율성을 극대화하고, 캐시 서버 이탈 시에도 안정적인 콘텐츠 제공을 보장함으로써, 자원 관리의 효율성을 높이고 최적화된 캐싱 시스템을 구현하는 데 기여하고자 한다.

Ⅱ. 관련연구

2.1. 이기종 환경에서의 내결함성 동적 스케줄링 연구

Mei 외 연구는 이기종 시스템에서 내결함성을 보장하는 동적 재스케줄링 기법을 제안하였다[6]. 이 연구는 서버 간 부하를 효율적으로 분산하여 시스템 신뢰성을 유지하고, 작업 중단이나 시스템 장애 시 신속한 복구가 가능하도록 설계되었다. 이기종 시스템에서 각 서버의 자원 상태를 실시간 모니터링하며, 장애 발생 시 작업을 동적으로 재분배하여 성능 저하를 방지한다. 그러나 이 연구는 캐시 서버의 불규칙한 참여와 이탈로 인한 콘텐츠 불균형 문제를 깊이 다루지 않았으며, 캐싱 시스템에서의 내결함성 유지 방안은 포함되지 않는다. 본 연구는 캐시 서버 이탈 시 미디어 콘텐츠 간 균형을 유지하기 위한 내결함성 캐싱 알고리즘을 제안함으로써 이기종 환경에서의 신뢰성 유지 문제를 보완하고자 한다.

2.2. 다중 에이전트 기반 협력 캐싱 연구

X. Fan 외는 동적 이기종 네트워크에서 다중 에이전트를 활용한 협력 캐싱 시스템을 제안하였다[7]. 이 시스템은 네트워크 내 다양한 노드 간 협력을 통해 콘텐츠 전송을 최적화하고, 대역폭 및 자원 사용 효율을 극대화하는 데 중점을 둔다. 연구에서는 네트워크가 변동되는 상황에서도 다중 에이전트가 각 노드 상태를 분석하고 실시간으로 최적의 콘텐츠 전송 경로를 선택하여 사용자에게 신속하게 콘텐츠를 제공하는 것을 목표로 한다. 그

러나 이 연구는 캐시 서버의 이탈로 인한 콘텐츠 불균형 문제나 내결함성 유지에 대한 고려가 부족하다. 본 연구는 캐시 서버 이탈 시 콘텐츠 간 균형을 유지하여 안정적인 서비스 품질을 보장하는 알고리즘을 제안하고자 한다.

2.3. 이기종 캐시 시스템에서의 동적 작업 병렬 처리 연구

Wang 외는 이기종 캐시 시스템에서 에너지 효율과 작업 병렬성을 최적화하는 방안을 제시하였다[8]. 이 연구는 캐시 서버 간 작업 부하를 분산하고 자원 사용을 효율화하여 성능을 극대화하는 데 중점을 두고 있다. 대규모 작업을 병렬로 처리해 시스템 응답 시간을 줄이고, 에너지 소비를 줄이는 방향으로 설계되었다. 각 서버의 성능을 고려하여 작업을 동적으로 분배하고, 실시간으로 작업 처리 상황을 모니터링하며 시스템 상태에 맞게 자원을 재분배하는 것이 주요 목표이다. 그러나 이 연구는 캐시 서버의 동적 참여와 이탈로 인한 콘텐츠 불균형 문제를 다루지 않았다. 본 연구는 이기종 캐시 서버 네트워크에서 캐시 서버 이탈 시에도 미디어 콘텐츠 간 균형을 유지함으로써 안정적인 콘텐츠 제공을 보장하는 알고리즘을 제안함으로써 기존 연구의 한계를 보완하고자 한다.

Ⅲ. 내결함성 캐싱 시스템

3.1. 내결함성 캐싱 시스템 설계

본 연구에서 제안하는 내결함성 캐싱 시스템은 캐시 서버의 동적 이탈과 참여로 인한 콘텐츠 불균형 문제를 해결하기 위해 설계되었다. 시스템은 이기종 디바이스들로 구성된 캐시 서버 네트워크 환경을 가정하며, 네트워크에 참여 중인 서버의 상태를 실시간으로 모니터링한다. 캐시 서버가 이탈할 경우, 시스템은 이탈한 서버에 저장된 미디어 콘텐츠의 종류, 해상도, 포맷, 수량 등을 즉시 확인하고, 해당 콘텐츠를 잔여 캐시 서버들의 가용 자원을 분석하여 균형 있게 재배치하는 데 초점을 맞춘다. 이 과정에서 CPU, 메모리 용량, 네트워크 대역폭, 스토리지 용량 등 다양한 하드웨어 자원을 고려하여 최적화된 방식으로 콘텐츠를 분산시킨다. 이후 콘텐츠가 성공적으로 재배포되었는지 확인하여 네트워크의 안정성을 유지한다.

그림 1은 동적 이탈과 참여에 따른 캐시 서버 복구를 위한 내결함성 캐싱 알고리즘을 보여준다. [의사코드 라인 10-12] 이 알고리즘은 네트워크 상의 캐시 서버 상태를 주기적으로 모니터링하며, 서버가 정상적으로 네트워크에 참여하고 있는지 확인한다. [의사코드 라인 13-32]

Algorithm 1 Cache Server Recovery Algorithm

```
1: function DistributeContent(C)
2:     Sort servers by S_Memory (descending)
3:     for each server do
4:         if server can host C then
5:             Assign C to the server
6:             break
7:         end if
8:     end for
9: end function
10: while monitoring cache servers do
11:     Check server status
12: end while
13: if a cache server leaves then
14:     Identify media content C on the server
15:     Extract C_type, C_resolution, C_format
16:     if new server joins within T_out then
17:         Prioritize and DistributeContent(C)
18:     else
19:         Analyze resources S of existing cache servers
20:         S ←S_CPU, S_Memory, S_Bandwidth, S_Storage
21:         DistributeContent(C)
22:         while C is not accommodated and reducible do
23:             Lower C_resolution DistributeContent(C)
24:         end while
25:         if no server can host any C then
26:             Wait for new server to join
27:             if new server joins then
28:                 Prioritize and DistributeContent(C)
29:             end if
30:         end if
31:     end if
32: end if
33: Confirm C redistribution
34: Return system to stable state
```

Fig. 1. Fault-Tolerant Caching Algorithm for Cache Server Recovery.
그림 1. 캐시 서버 복구를 위한 내결함성 캐싱 알고리즘

캐시 서버의 이탈이 감지되면, 이탈한 서버에 저장된 미디어 콘텐츠를 확인하고, 콘텐츠의 종류, 해상도, 포맷 등의 세부 정보를 추출한다(라인 14-15). 새로운 캐시 서버가 타임아웃 내에 네트워크에 참여할 경우, 이탈한 서버의 콘텐츠를 우선적으로 새 서버에 배포하여 콘텐츠 비율을 신속하게 복구한다(라인 16-17). 타임아웃 내에 새로운 서버가 참여하지 않으면, 네트워크에 남아 있는 기존 캐시 서버들의 CPU, 메모리, 네트워크 대역폭, 스토리지 등의 가용 자원을 분석한다(라인 19-20). 이후, 가장 많은 가용 자원을 보유한 서버 순으로 배포를 시도한다(라인 21, 1-9). 이 과정에서 콘텐츠를 수용할 충분한 자원을 가진 서버가 없을 경우, 콘텐츠 해상도를 낮춰 재배포를 시도하며(라인 22-24), 더 이상 해상도를 낮출 수 없을 경우에는 새로운 서버가 참여할 때까지 기다렸다가 배포를 시도한다(라인 25-30). 이 과정을 통해 이탈한 서버의 콘텐츠를 다른 서버에 재배포함으로써 콘텐츠 비

율을 유지하고 네트워크의 안정성을 보장할 수 있다. 마지막으로, 콘텐츠가 성공적으로 재배포되었는지 확인한 후 시스템을 안정된 상태로 복귀시킨다(라인 33-34).

3.2. 내결함성 캐싱 시스템 구현

본 연구에서 제안하는 캐싱 시스템은 다양한 성능의 이기종 서버들로 구성된 캐시 서버 네트워크 환경을 가정한다. 이기종 서버는 일반적으로 네 가지 유형으로 분류할 수 있다. 첫 번째 유형의 서버는 고성능 CPU, 대용량 메모리, 빠른 네트워크 대역폭과 대용량 저장 공간을 제공한다. 두 번째 유형은 중간 성능의 CPU, 메모리와 네트워크 대역폭을 갖추고 있으나, 대용량 저장 공간을 제공한다. 세 번째 유형은 낮은 성능의 CPU와 메모리, 적당한 네트워크 대역폭을 가진 서버이며, 네 번째 유형은 저전력으로 작동하며, 제한된 CPU와 메모리 성능을 갖춘 서버이다. 이에 대응하기 위하여 워크스테이션, 노트북, 미니 PC와 라즈베리파이로 이기종 캐시 서버 네트워크를 구성하였다. 또한, 이기종 캐시 서버 클러스터를 구축하고 운영하기 위해 쿠버네티스(Kubernetes) 기술을 활용한다. 쿠버네티스는 다양한 서버들로 이루어진 캐시 서버 네트워크를 효과적으로 관리하고, 서버의 참여와 이탈을 실시간으로 모니터링하는 역할을 수행한다. 또한, 제안하는 알고리즘이 탑재된 서비스를 Pod 형식으로 배포하여, 캐시 서버의 이탈과 참여 시에 신속한 대응이 가능하도록 구현되었다.

Ⅳ. 실험 및 결과

4.1. 이기종 캐시 서버 네트워크 테스트베드

실제 7대의 이기종 기기를 기반으로 캐시 서버 네트워크를 구축하여 제안된 시스템을 실험하였다. 이기종 기기들은 워크스테이션, 노트북, 미니 PC, 라즈베리파이와 같은 다양한 하드웨어 성능의 장치들로 구성되었으며, 각 기기의 사양과 이기종 테스트베드 환경은 표 1과 그림 2에 나타나 있다. 그림 2의 숫자는 표 1에 표시된 기기들의 순서에 따른 번호이다. CPU, 메모리, 스토리지, 네트워크 대역폭 등 하드웨어 자원이 상이한 7대의 기기들로 이루어진 이기종 테스트베드 환경에서 제안된 시스템의 안정성과 확장성을 평가함으로써 더욱 정확한 검증을 수행할 수 있었다. 캐싱 시스템은 쿠버네티스(Kubernetes) 클러스터로 운영되며, 이를 통해 캐시 서버의 동적 관리와 자원 할당을 처리하였다.

Table 1. Heterogeneous Testbed Specifications.
표 1. 이기종 테스트베드 사양

Device	CPU	Memory	Storage	Network Speed
612Workstation	12th Gen Intel(R) Core(TM) i7-12700	62Gi	468G	1000Mb/s
612notebook	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz	7.4Gi	234G	100Mb/s
612notebook2	13th Gen Intel(R) Core(TM) i7-13700H	30Gi	938G	1000Mb/s
612odyssey1	Intel(R) Celeron(R) J4125 CPU @ 2.00GHz	7.6Gi	27G	100Mb/s
612rasp4-1	ARM Cortex-A72 1.5GHz	7.6Gi	29G	100Mb/s
612rasp4-2	ARM Cortex-A72 1.5GHz	7.6Gi	29G	10Mb/s
612rasp3-1	ARM Cortex-A57 1.4GHz	905Mi	29G	100Mb/s

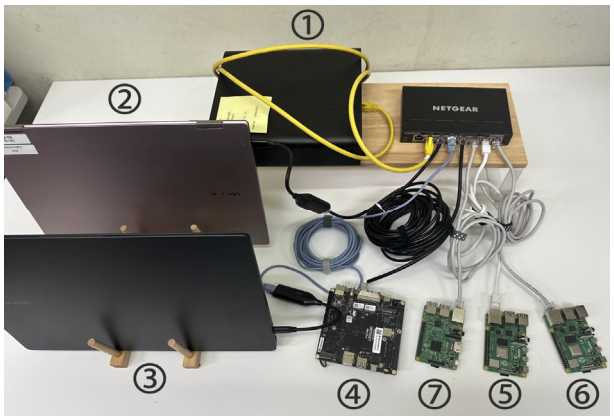

Fig. 2. Heterogeneous testbed environment.
그림 2. 이기종 테스트베드 환경

4.2. 내결함성 캐싱 시스템 실험 및 결과

실험은 이기종 캐시 서버 네트워크(4.1 참고)에서, 임의의 캐시 서버가 동적으로 이탈하거나 참여할 때 클러스터의 총 콘텐츠 종류와 수를 모니터링하여 미디어 콘텐츠 간 균형성을 확인하는 방식으로 진행되었다. 클러스터 내 가용 자원이 가장 높은 장치가 내결함성 캐싱 시스템이 실행되는 장치로 선정된다. 그림 3과 4는 캐싱 시스템 구현을 위해 사용한 쿠버네티스 구성을 나타낸다. ROLES 열을 통해서 각 캐시 서버의 역할을 확인할 수 있으며, 가용 자원이 가장 높은 612workstation 이 control-plane, 즉, 내결함성 캐싱 시스템이 실행되는 장치로 선정되어 동작하고 있는 것을 알 수 있다. 내결함성 캐싱 시스템은 클러스터 내 캐시 서버의 상태를 주기적으로 모니터링하여, 각 캐시 서버가 가용한 상태인지와 관리 중인 콘텐츠의 종류를 확인한다. 캐시 서버의 상태는 가용 여부에 따라 “Ready”와 “NotReady”로 구분

되며, 특정 캐시 서버가 요청에 응답하지 않거나 내부 오류가 발생하면 해당 서버는 “NotReady” 상태로 전환된다. 그림 3은 모든 캐시 서버가 가용 상태인 쿠버네티스 클러스터를 나타내고 있으며, 그림 4는 612notebook 캐시 서버가 비가용 상태로 전환된 후의 쿠버네티스 클러스터를 나타내고 있다. 내결함성 캐싱 시스템은 “NotReady” 상태로 전환된 캐시 서버가 관리하던 콘텐츠 목록을 확인하여 클러스터 내 다른 캐시 서버에 재배포한다. 이때, 미디어 콘텐츠 간 균형을 유지하도록 콘텐츠를 분산시키며, 각 서버의 가용 자원을 고려한다.

NAME	STATUS	ROLES	AGE	VERSION
612notebook	Ready	cacheserver,worker	105d	v1.29.5
612notebook2	Ready	cacheserver,worker	97d	v1.29.5
612odyssey1	Ready	cacheserver	35d	v1.29.5
612rasp3-1	Ready	cacheserver	34d	v1.29.5
612rasp4-1	Ready	cacheserver	34d	v1.29.5
612rasp4-2	Ready	cacheserver	35d	v1.29.5
612workstation	Ready	control-plane	105d	v1.29.5

Fig. 3. Kubernetes cluster with all cache servers available.
그림 3. 모든 캐시 서버가 가용 상태인 쿠버네티스 클러스터

NAME	STATUS	ROLES	AGE	VERSION
612notebook	NotReady	cacheserver,worker	106d	v1.29.5
612notebook2	Ready	cacheserver,worker	98d	v1.29.5
612odyssey1	Ready	cacheserver	36d	v1.29.5
612rasp3-1	Ready	cacheserver	35d	v1.29.5
612rasp4-1	Ready	cacheserver	35d	v1.29.5
612rasp4-2	Ready	cacheserver	36d	v1.29.5
612workstation	Ready	control-plane	106d	v1.29.5

Fig. 4. Kubernetes cluster with an unavailable cache server.
그림 4. 한 캐시 서버가 비가용 상태인 쿠버네티스 클러스터

실험에 사용된 콘텐츠는 각각 1080p, 720p, 480p 해상도를 가진 세 가지 유형의 미디어 콘텐츠로 구성되었으며, 총 9가지 종류의 콘텐츠를 통해 실험이 진행되었다. 실험은 실제 테스트베드와 캐시 서버를 도커(Docker)로 대체한 가상 환경에서 수행되었다. 실제 테스트베드에서는 5분마다 임의로 장치 상태를 변경하고, 1시간 후 캐시 서버 내 콘텐츠 분포가 어떻게 변했는지를 확인하였다. 가상 환경에서는 캐시 서버의 수를 100개로 확장하여, 총 100개의 도커 컨테이너로 실험을 진행하였다. 이 컨테이너들을 25개 단위로 서로 다른 자원 스펙을 가지는 그룹으로 구성하였다. 첫 번째 그룹은 CPU 2, 8GB 메모리, 1000Mb/s 네트워크 대역폭을, 두 번째 그룹은 CPU 1, 4GB 메모리, 100Mb/s 네트워크 대역폭을, 세 번째 그룹은 CPU 0.5, 2GB 메모리, 10Mb/s 네트워크 대역폭을, 네 번째 그룹은 CPU 0.2, 1GB 메모리, 10Mb/s 네트워크 대역폭을 갖추고 있다. 가상 환경에서도 실제 테스트베드와 동일하게 5분마다 캐시 서버 역할을 수행하는 임의의 도커 컨테이너의 상태를 변경하고, 1시간 후

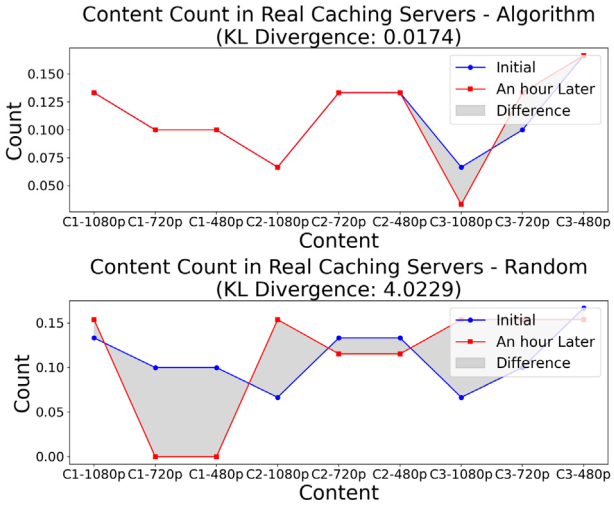

Fig. 5. Graph of number of contents in a real environment.
그림 5. 실제 환경에서의 콘텐츠 수 그래프

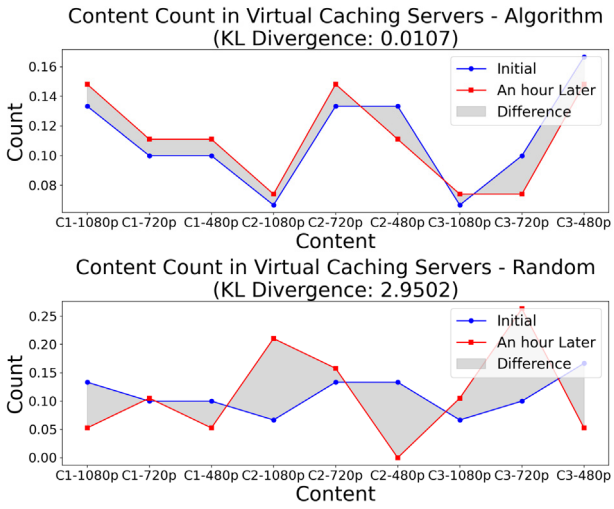

Fig. 6. Graph of number of contents in a Virtual environment.
그림 6. 가상 환경에서의 콘텐츠 수 그래프

캐시 서버 내 콘텐츠 분포 변화를 분석하였다.

그림 5와 6은 각각 실제 환경과 가상 환경에서의 실험 결과를 나타낸다. 상단에 위치한 그래프는 제안 내결함성 캐싱 시스템에 따른 결과이며, 하단에 위치한 그래프는 랜덤 콘텐츠 배포에 따른 결과이다. 각 그래프에서 파란 선은 캐시 서버들이 실험 초기에 관리하고 있던 콘텐츠의 분포를 나타내며, 빨간 선은 1시간의 실험 후 캐시 서버가 관리하고 있는 콘텐츠의 분포를 나타낸다. 그림 5에서는 일부 캐시 서버의 이탈로 인해 C3-1080p 콘텐츠를 수용할 서버가 부족하여 해당 콘텐츠 대신 저해상도 버전인 C3-720p 콘텐츠가 배포된 사례를 확인할 수 있다. 제안된 내결함성 캐싱 알고리즘이 적용된 경우, 캐시 서버의 동적 이탈 및 참여에도 불구하고 미디어 콘텐츠 간

균형이 안정적으로 유지되는 것을 확인할 수 있다. 반면, 랜덤 방식으로 콘텐츠를 관리한 환경에서는 초기 콘텐츠 분포에서 크게 벗어나는 것을 확인하였다. 이는 제안된 알고리즘이 콘텐츠 불균형 문제를 효과적으로 해결함으로써 안정적인 서비스 품질을 제공할 수 있음을 보여준다.

V. 결론

본 연구는 이기종 캐시 서버 네트워크에서 캐시 서버의 동적 이탈과 참여로 인한 콘텐츠 불균형 문제를 해결하기 위해 내결함성 캐싱 시스템을 제안하고, 이를 실제 및 가상 환경에서 실험을 통해 검증하였다. 제안된 시스템은 캐시 서버의 상태를 실시간으로 모니터링하여, 서버 이탈 시 미디어 콘텐츠 간 균형을 유지하기 위한 효율적인 알고리즘을 제공한다. 특히, 이기종 환경에서 서버 간의 자원 불균형과 네트워크 대역폭을 고려한 콘텐츠 배포 메커니즘을 통해, 캐시 서버 이탈 및 참여에 따른 성능 저하를 최소화하고 네트워크의 안정성을 유지하는 데 중점을 두었다.

실험 결과, 제안된 캐싱 시스템은 캐시 서버의 동적 이탈과 참여에도 불구하고 콘텐츠 분포를 효과적으로 유지함을 확인하였다. 이는 제안된 시스템이 이기종 환경에서 효율적으로 동작하며, 콘텐츠 불균형 문제를 해결하는 데 효과적임을 보여준다. 또한, 실제 7대의 이기종 캐시 서버로 구성된 네트워크를 구축하여 시스템의 실용성을 검증하였고, 가상 환경에서 100개의 캐시 서버로 실험한 결과에서도 제안된 시스템은 확장성 있게 동작하여 대규모 네트워크 환경에서도 성능 저하 없이 콘텐츠 관리가 가능함을 입증하였다.

본 연구에서 제안된 내결함성 캐싱 시스템은 향후 엣지 컴퓨팅 및 분산 캐싱 환경에서 안정적이고 효율적인 콘텐츠 제공을 보장할 수 있는 중요한 기술적 기여를 할 것으로 기대된다. 특히, 다양한 기기로 구성된 이기종 네트워크에서의 적용 가능성을 고려할 때, 저전력 장치를 활용한 캐시 서버 네트워크의 구축 및 운영에도 실질적인 성과를 얻을 수 있을 것이다. 향후 연구에서는 제안된 시스템을 더욱 확장하여, 캐시 서버 간의 콘텐츠 인기도를 반영한 배포 전략 및 동적 자원 할당 알고리즘을 추가로 연구할 계획이다.

References

[1] Jazaeri, S.S., Asghari, P., Jabbehdari, S. et al. Toward caching techniques in edge computing over SDN-IoT architecture: a review of challenges, solutions, and open issues. *Multimed Tools Appl* 83, 1311-1377, 2024.

[2] A. R. Jolly, M. K. Chakravarthi, N. K. Jindal, and D. Birlasekaran, "Transparent Proxy Cache Server using Raspberry Pi," *Indian Journal of Science and Technology*, vol.9, no.44, pp.1-6, 2016. DOI: 10.17485/ijst/2016/v9i44/105312

[3] Y. Liu, P. Zhong, Z. Yang, W. Li, and S. Li, "Computation Offloading Based on a Distributed Overlay Network Cache-Sharing Mechanism in Multi-Access Edge Computing," *Future Internet*, vol.16, no.4, article 136, pp.1-18, 2024. DOI; 10.3390/fi16040136

[4] X. Xu, C. Feng, S. Shan, T. Zhang, and J. Loo, "Proactive Edge Caching in Content-Centric Networks with Massive Dynamic Content Requests," *IEEE Access*, vol.8, pp.59906-59921, 2020. DOI: 10.1109/ACCESS.2020.2983068

[5] Gao, Z., Cecati, C., & Ding, S. X., A survey of fault diagnosis and fault-tolerant techniques-Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE transactions on industrial electronics*, 62(6), 3757-3767, 2015.

[6] J. Mei, K. Li, X. Zhou, and A. Y. Zomaya, "Fault-Tolerant Dynamic Rescheduling for Heterogeneous Computing Systems," *Journal of Grid Computing*, vol.13, no.4, pp.507-525, 2015. DOI: 10.1007/s10723-015-9331-1

[7] X. Fan, H. Chen, Z. Ni, G. Li, H. Sun and J. Yu, "Multi-Agent Collaborative Caching Strategies in Dynamic Heterogeneous D2D Networks," *IEEE Transactions on Vehicular Technology*, vol.73, no.5, pp.7204-7217, 2024. DOI: 10.1109/TVT.2023.3345863

[8] M. Wang, T. Ta, L. Cheng and C. Batten, "Efficiently Supporting Dynamic Task Parallelism on Heterogeneous Cache-Coherent Systems," *in Proceeding of 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, Valencia, Spain, pp.173-186, 2020. DOI: 10.1109/ISCA45697.2020.00025

BIOGRAPHY

Hyeon-Gi Kim (Member)

2024 : BS degree in Computer Engineering, Hanbat University.
2024~present : MS course in Computer Engineering, Hanbat National University

Gyu-SiK Ham (Member)

2023 : BS degree in Computer Engineering, Hanbat University.
2023~present : MS course in Computer Engineering, Hanbat National University

Jin-Woo Kim (Member)

2024 : BS degree in Computer Engineering, Hanbat University.
2024~present : MS course in Computer Engineering, Hanbat National University

Soo-Young Jang (Member)

2006 : BS degree in Industrial Engineering, Korea Advanced Institute of Science and Technology (KAIST).
2008 : MS degree in Industrial & Systems Engineering, Korea Advanced Institute of Science and Technology(KAIST).
2014 : PhD degree in Industrial & Systems Engineering, Korea Advanced Institute of Science and Technology (KAIST).
2014~2017 : Senior Researcher, Samsung Electronics.
2017~2023 : Senior Researcher, Electronics and Telecommunications Research Institute(ETRI).
2023~present : Professor, Dept, of Computer Engineering, Hanbat National University

Chang-Beom Choi (Member)

2005 : BS degree in Computerl Engineering, Kyunghee University.
2007 : MS degree in Computer Science, Korea Advanced Institute of Science and Technology(KAIST).
2014 : PhD degree in Electronic and Electrical Engineering, Korea Advanced Institute of Science and Technology(KAIST).
2014~2021 : Professor. Dept, of Global Entrepreneurship and Information Communication Technology, Handong University
2021~present : Professor, Dept, of Computer Engineering, Hanbat National University