Performance Co-Pilot, Bpftrace, Grafana 기반 슈퍼컴퓨터 모니터링 및 성능 분석 시스템 구축 방안 연구

곽재혁*
*한국과학기술정보연구원
jhkwak@kisti.re.kr

A study on how to build a supercomputer monitoring and performance analysis system based on Performance Co-Pilot, Bpftrace and Grafana

Jae-Hyuck Kwak*
*Korea Institute of Science and Technology Information

요 약

슈퍼컴퓨터는 수백~수천 노드의 컴퓨팅 자원이 연결되어 복잡한 계산이나 대규모 데이터를 병렬 처리하며 일부 노드에서 발생하는 예상치 못한 문제는 전체적인 시스템 성능 저하로 이어질 수 있기 때문에 슈퍼컴퓨터 모니터링과 성능 분석은 슈퍼컴퓨터를 구축하고 운영하는데 필수적인 요소로 볼 수 있다. 본 논문에서는 오픈소스 소프트웨어인 Performance Co-Pilot, Bpftrace, Grafana를 활용한 슈퍼컴퓨터 모니터링 및 성능분석 시스템 구축 방안을 제안하였으며 이를 통해서 확장가능하면서도 유연한 구조의 슈퍼컴퓨터 모니터링 및 성능 분석이 가능함을 보였다.

1. 서론

슈퍼컴퓨터는 계산 시뮬레이션을 목적으로 주로 과학계산 분야에서 사용되었으나 최근 인공지능 및 빅데이터 응용 분야에서 대규모 계산 및 데이터 처리가 요구됨에 따라 슈퍼컴퓨터의 활용 분야가 점차적으로 확대되고 있다.[1][2] 슈퍼컴퓨터는 주로 MPI와 같은 병렬라이브러리를 사용하여 수백~수천 노드의 컴퓨팅 자원을 고성능 네트워크로 연계하여 복잡한 계산이나 대규모 데이터를 병렬 처리하는데 일부 노드에서 발생하는 예상치 못한 문제는 전체적인 시스템의 성능 저하로 이어질 수 있어서 슈퍼컴퓨터에서 모니터링과 성능 분석은 슈퍼컴퓨터를 구축하고 운영하기 위한 필수적인 요소로 볼 수 있다.

본 논문에서는 오픈소스 소프트웨어를 활용한 확장가능하면서 유연한 구조의 슈퍼컴퓨터 모니터링 및 성능 분석 시스템을 제안하였다. 시스템 모니터링 및 성능 분석 프레임워크로서 Performance Co-Pilot, 커널 수준의 모니터링 및 성능 분석 프레임워크로서 Bpftrace, 모니터링 가시화 프레임워크로서 Grafana를 사용하였으며 이를 유기적으로 통합하여 슈퍼컴퓨터에서 요구되어지는 유연하면서도 확장가능한 모니터링 및 성능 분석 시스템 구축이 가능함을 보였다.

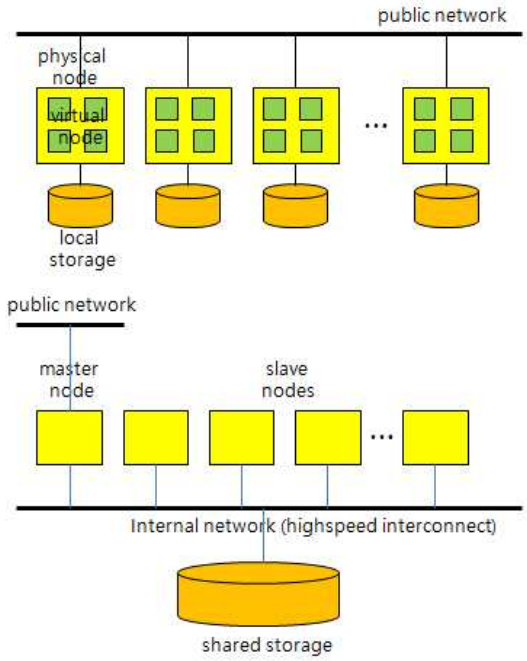
본 논문의 구성은 다음과 같다. 2장에서는 슈퍼컴퓨터 모니터링 및 성능 분석 시스템의 요구사항을 분석하였고 3장에서는 이를 충족하기 위해서 본 논문에서 활용한 관련 기술에 대해서 분석하였다. 4장에서는 이를 기반으로 슈퍼컴퓨터 모니터링 및 성능 분석 시스템 구축 방안을 제안하였다. 5장에서는 결론과 향후 계획을 제시하였다.

2. 슈퍼컴퓨터 모니터링 및 성능 분석 시스템의 요구사항

슈퍼컴퓨터 모니터링 및 성능 분석 시스템을 구축하기 위해서는 다음과 같은 요구사항이 충족될 필요가 있다.

(1) 슈퍼컴퓨터 아키텍처 대응

슈퍼컴퓨터 아키텍처는 클라우드 아키텍처와 차이점이 있으며 그림 1과 같다. 클라우드는 가상 머신 기반으로 한 서비스 목적으로 활용되어 왔다면 슈퍼컴퓨터는 대규모 계산 작업의 병렬 처리를 목적으로 활용되어 왔기 때문에 인프라 구조에 있어 차이점이 있다. 슈퍼컴퓨터는 마스터-슬레이브 구조를 가지는데 모든 노드는 Infiniband와 같은 초고속 인터커넥트로 연결되어 있고 PBS, Slurm과 같은 큐잉시스템에 의해서 관리되며 대부분 로컬 스토리지 없이 Lustre와 같은 병렬파일시스템을 공유 스토리지로 사용한다. 따라서 슈퍼컴퓨터 모니터링 및 성능 분석 시스템 구축 시에는 이와 같은 슈퍼컴퓨터 아키텍처가 고려되어야 한다.



(그림 1) 클라우드 아키텍처와 슈퍼컴퓨터 아키텍처 비교

(2) 모니터링 및 성능 분석 메트릭 확장성

슈퍼컴퓨터는 다수의 사용자가 공유하기 때문에 사용자가 요구하는 다양한 응용 소프트웨어가 설치될 수 있고 MPI와 같은 병렬 라이브러리를 사용하므로 일부 노드의 문제 상황이 전체 시스템 성능에 영향을 미칠 수 있다. 따라서 문제 상황이 발생했을 때 사용자 수준과 커널 수준에서 시스템의 현재 상황을 정확하게 파악할 수 있어야 한다. 이를 위해서 인프라 관리자가 필요로 하는 모니터링 및 성능 분석 메트릭에 대한 동적인 관리가 가능해야 하며 모니터링 및 성능 분석 메트릭을 추가하고 제거할 수 있는 확장성이 요구된다.

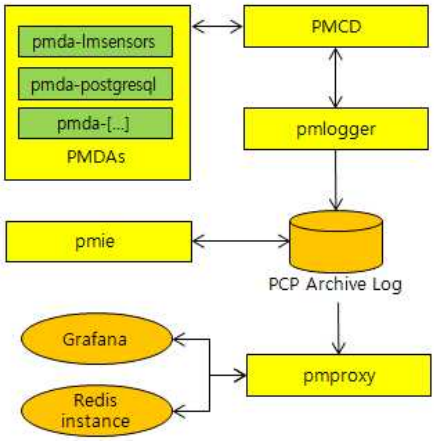
(3) 모니터링 및 성능 분석 메트릭 시각화 유연성

모니터링 및 성능 분석을 위한 시각화는 인프라 운영 시에 관리자가 시스템 전반적인 상황을 신속하게 파악하는데 도움을 주며 관리자가 요구하는 모니터링 및 성능 분석 메트릭에 대한 시각화에 유연성이 요구된다. 관리자는 라이브 데이터 및 히스토리 데이터로부터 인프라 운영 관리에 필요한 모니터링 및 성능 분석 메트릭을 시각화할 수 있어야 하며 문제 상황에 따라서 시각화되는 정보를 유연하게 변경할 수 있어야 한다.

3. 관련 기술 분석

(1) Performance Co-Pilot

Performance Co-Pilot(PCP)[3]은 시스템 성능 분석 툴킷으로 시스템의 성능 데이터를 수집, 통합, 분석할 수 있는 확장 가능한 프레임워크를 제공한다. 그림 2는 PCP의 구조를 보여준다.

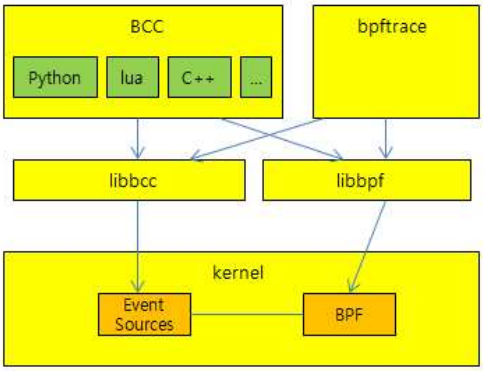


(그림 2) PCP 구조

모니터링 대상이 되는 호스트는 pmcd(Performance Metrics Collector Daemon)을 실행하며 동일한 호스트에서 하나 이상의 pmda(Performance Metrics Domains Agent)를 제어하는데 각각의 pmda는 목적에 맞는 성능 정보를 수집하는 역할을 수행한다. pmlogger는 pmcd로부터 성능 정보를 기록한다. pmproxy는 모든 PCP서비스에 대한 REST API를 제공하며 pmlogger에 의해 생성된 아카이브를 redis 데이터베이스에 저장하여 빠르고 확장 가능한 시계열 쿼리를 지원한다. pmie(Performance Metrics Inference Engine)는 성능에 관한 자동화된 필터링과 추론 기능을 제공한다.

(2) Bpftrace

Bpftrace[4]는 BCC(BPF Compiler Collection), BPF(Berkeley Packet Filter)[5] 기반 고수준 트레이싱 언어이다. 그림 3은 Bpftrace의 구조를 보여주고 있다. Bpftrace는 프로브(probe)에 부착하고 프로그램을 로드한 후에 계측(instrumentation)을 수행하기 위해서 libbcc, libbpf를 사용하며 스크립트를 BPF 바이트코드로 컴파일하기 위해서 LLVM을 백엔드로 사용한다. Bpftrace는 kprobe, uprobe와 같은 동적 계측부터 tracepoints, USDT와 같은 정적 계측까지 다양한 이벤트 소스를 지원하여 시스템 이상 현상이 발생할 경우 시스템 내부를 상세하게 들여다 볼 수 있는 방법을 제공한다.



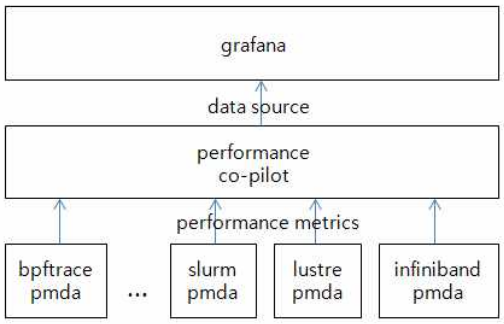
(그림 3) Bpftrace 구조

(3) Grafana

Grafana[6]는 관측 데이터를 시각화 하는데 최적화된 대시보드를 제공하는 데이터 분석 및 대화형 시각화 웹 프레임워크로서 다양한 데이터 소스에서 데이터를 수집하고 이 데이터를 쿼리하고 쉽게 분석할 수 있는 기반을 제공한다. Grafana는 PCP를 데이터 소스로 사용할 수 있어서 PCP에서 수집된 다양한 메트릭을 사용자 정의 기반으로 시각화하는 것이 가능하다.

4. PCP, Bpftrace, Grafana 기반 슈퍼컴퓨터 모니터링 및 성능 분석 시스템 구축 방안

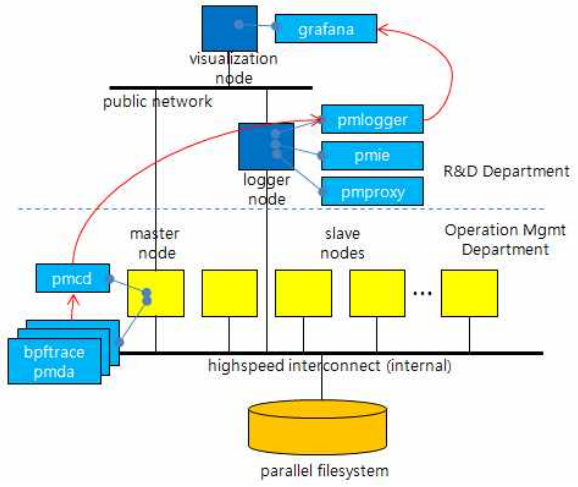
본 논문에서 제안한 슈퍼컴퓨터 모니터링 및 성능 분석 구조는 그림 4와 같다.



(그림 4) PCP, Bpftrace, Grafana 기반 슈퍼컴퓨터 모니터링 및 성능 분석 구조

PCP는 슈퍼컴퓨터 모니터링 및 성능 분석 시스템을 위한 코어 프레임워크로 사용하였다. Bpftrace는 커널 수준의 모니터링 및 성능 분석 프레임워크로서 PCP의 pmda로서 통합하였다. 슈퍼컴퓨터에서 운영되는 큐잉시스템, 병렬파일시스템, 고성능 인터커넥트에 대한 모니터링 및 성능 분석 메트릭은 pmda로서 PCP에 연동될 수 있다. Grafana는 PCP를 데이터소스로 사용하며 모니터링 및 성능 분석 가시화 프레임워크로서 구성하였다. pmda에서 수집되는 모니터링 및 성능 분석 매트릭은 Grafana를 통해 표출되며 Grafana의 동적인 대시보드 및 패널 생성 기능으로 인해 인프라 관리자의 요구 사항에 최적화된 모니터링 및 성능 분석 시스템 구성이 가능하다.

슈퍼컴퓨터 아키텍처에서 PCP, Bpftrace, Grafana 기반 슈퍼컴퓨터 모니터링 및 성능 분석 시스템을 적용하면 그림 5와 같다.

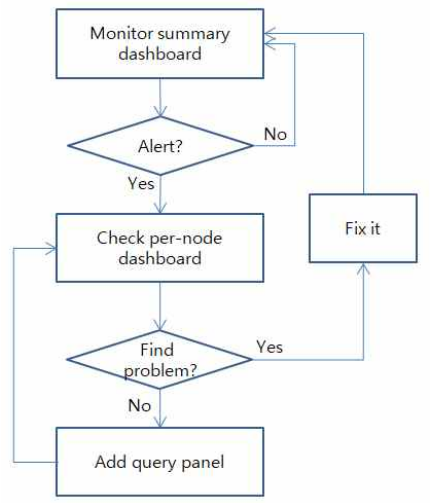


(그림 5) PCP, Bpftrace, Grafana 기반 슈퍼컴퓨터 모니터링 및 성능분석 시스템 적용

슈퍼컴퓨터의 마스터 노드와 슬레이브 노드에는 pmda, pmcd가 설치되어 모니터링 및 성능 분석 메

트릭을 수집한다. 로거 노드는 별도로 구성되어 pmlogger를 통해 슈퍼컴퓨터로부터 수집된 모니터링 및 성능 분석 메트릭을 통합하고 pmie를 통해 관리자가 요구하는 성능에 관한 자동화된 필터링과 추론 기능을 수행할 수 있다. 또한, pmproxy를 통해서 Grafana를 위한 데이터 소스로서 연동된다. 로거 노드는 외부 네트워크에 연결되어 있지만 슈퍼컴퓨터 IP주소 범위 이외에 모니터링 및 성능 분석 메트릭 데이터를 요청하는 가시화 노드 IP주소에서만 접근되도록 방화벽을 설정하면 된다. 가시화 노드는 Grafana를 통해 PCP에서 수집된 모니터링 및 성능 분석 데이터를 표출한다. 앞서 언급한 것처럼 Grafana의 유연한 구조로 인해 인프라 관리자의 요구 사항에 맞는 최적화된 모니터링 및 성능 분석 메트릭 데이터 가시화가 가능할 것이다.

마지막으로 본 논문에서 제안한 슈퍼컴퓨터 모니터링 및 성능 분석 시스템에 기반한 인프라 관리자의 작업 흐름을 정리하면 그림 6과 같다.



(그림 6) 슈퍼컴퓨터 모니터링 및 성능분석 시스템에 기반한 인프라 관리자 작업 흐름

인프라 관리자는 pmie를 통해 슈퍼컴퓨터의 이상 상황에 대한 알람 규칙을 생성하고 Grafana에 이를 포함하여 시스템 전반적인 모니터링 및 성능 분석 데이터를 표출하는 요약 대시보드를 구성하고 모니터링한다. 알람이 발생하게 되면 알림이 리턴한 노드 정보를 기반으로 노드별 대시보드에서 노드의 현재 상황에 대한 보다 상세한 파악이 가능하다. 만약, 노드별 모니터링 및 성능 분석 메트릭을 분석하여 문제 해결이 안된다고 하면 저수준에서 문제점을 파악하기 위해서 Bpftrace 스크립트를 통해 커널 수준의 쿼리를 구성하고 Grafana에 패널을 동적으로 추가함으로서 계속해서 문제 원인을 파악하기 위해서 시도한다. 문제 상황이 파악되었다면 문제점을 해결하고 알람이 오프된 것을 확인 한 후에 요약 대시보드를 모니터링하는 상태로 복귀하면 된다.

5. 결론 및 향후 계획

본 논문에서는 PCP, Bpftrace, Grafana를 통합하여 유연하면서도 확장 가능한 슈퍼컴퓨터 모니터링 및 성능 분석 시스템 구축 방안에 대해서 알아보았다. 슈퍼컴퓨터 아키텍처는 클라우드 아키텍처와 차이가 있으며 이를 고려하여 최적화된 모니터링 및 성능 분석 시스템을 구축할 필요가 있다. PCP와 Bpftrace는 확장 가능한 모니터링 및 성능 분석 프레임워크를 제공하며 Grafana는 인프라 관리자가 필요로 하는 유연하면서도 최적화된 시각화를 제공할 수 있다. 본 논문에서는 이를 활용하여 슈퍼컴퓨터 아키텍처 기반으로 각각의 프레임워크를 통합 구성하는 방안을 보이고 이에 기반한 인프라 관리자의 작업 흐름을 정리함으로서 효율적인 슈퍼컴퓨터 모니터링 및 성능 분석이 가능함을 확인하였다.

슈퍼컴퓨터는 다양한 소프트웨어가 설치되고 다수의 사용자에 의해서 공유되는 시스템이기 때문에 신속하면서 정교한 모니터링 및 성능 분석이 요구된다. 이를 위해서 슈퍼컴퓨터 아키텍처에 최적화된 pmda 개발이나 알람 규칙 개발이 필요할 수 있으며 이것은 향후 연구 과제로 남아 있다.

※ 본 연구는 2021년도 한국과학기술정보연구원(KISTI) 주요사업 과제(K-21-L02-C08-S01, 초고성능컴퓨팅 공동활용을 위한 통합 환경 개발 및 구축)로 수행한 결과임

참고문헌

[1] Yutong Lu, Depei Qian, Haohuan Fu, Wenguang Chen, “Will supercomputers be super-data and super-AI machines?”, Communications of the ACM, 61(10), pp.82-87, 2018.

[2] 곽재혁, 변은규, “누리온의 HPC 융합 기술과 HPC 융합 기술 개발 사례”, 정보과학회지 제37권 10호, pp.9-16, 2019.

[3] Performance Co-Pilot, https://pcp.io/

[4] Bpftrace, https://github.com/iovisor/bpftrace

[5] Brendan Gregg, “BPF Performance Tools”, Addison-Wesley, 2019.

[6] Grafana, https://grafana.com/