**Iotracer 설치 및 실행 매뉴얼**

**(Centos8, multi-stream patch - kernel 4.9.216)**

**-----------------------------------------------------------------------------------------------------**

**1. Multi-Stream 커널 패치**

**2. iotracer 설치**

**3. iotracer 실행**

**-----------------------------------------------------------------------------------------------------**

**1. Multi-Stream 커널 패치**

**# multi-stream 커널 패치를 위한 kernel compile 패키지 설치 (커널 컴파일 패키지가 이미 설치된 경우 생략 가능)**

* **dnf upgrade -y**
* **dnf groupinstall "Development Tools" -y**
* **dnf install tar openssl-devel bc elfutils-libelf-devel ncurses-devel -y**
* **reboot**

**# multi-stream 커널 패치 (kernel ver. 4.9.216)**

* **git clone --depth=1 <https://github.com/multi-stream/multi-stream>**
* **curl -OJL <https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-4.9.216.tar.xz>**
* **tar -xJf linux-4.9.216.tar.xz**
* **cd linux-4.9.216**
* **patch -p1 < ../multi-stream/patches\_kernel/0001-consolidated\_initial-support-for-nvme-directive-streams-and-sas-multistream.patch**

**# Kernel compile**

* **make mrproper**
* **make menuconfig**
* **sed -ri '/CONFIG\_SYSTEM\_TRUSTED\_KEYS/s/=.+/=""/g' .config**
* **make -j$(nproc)**
* **make modules\_install**
* **make install**

**2. iotracer 설치**

**# iotracer 빌드에 필요한 라이브러리 설치 (이미 설치된 경우 생략 가능)**

* **sudo dnf upgrade**
* **dnf install epel-release**
* **sudo dnf install cmake llvm-devel clang-devel llvm-static elfutils-libelf-devel ncurses-devel libbsd-devel libblkid-devel python3**

**# python 링크 생성 (이미 되어 있는 경우 생략 가능)**

* **sudo ln -s /usr/bin/python3 /usr/bin/python**

**# bcc 설치**

* **git clone --depth=1 <https://github.com/iovisor/bcc.git>**
* **mkdir bcc/build; cd bcc/build**
* **cmake .. -DCMAKE\_INSTALL\_PREFIX=/usr**
* **make -j$(nproc)**
* **sudo make install**

**# procps 설치**

* **git clone --depth=1 <https://gitlab.com/procps-ng/procps>**
* **cd procps**
* **./autogen.sh**
* **./configure**
* **make -j$(nproc)**
* **make install**
* **echo "/usr/local/lib/" > /etc/ld.so.conf.d/procps.conf**
* **ldconfig**

**# iotracer 설치**

**tar -zxf iotracer.tar.gz**

**⋇ 가장 최근 버전인 20200826\_iotracer\_1.tar.gz 사용 할 것**

>  **cd iotracer**

* **mkdir build; cd build # build 폴더 존재 시 삭제 후 진행**
* **cmake ..**
* **make**

**3. iotracer 실행**

**Iotracer로 IO를 추적해서 생성한 파일에 대해 ioparser 를 실행해서 성능 요약 파일 생성**

**1) iotracer 옵션 및 파라미터**

**Usage: [OPTION...] <watch directory>**

**-o, --output path iotracer 추적결과 저장 경로**

**-m, --dm path skhynix device manager 설치 경로**

**-d, --disable\_dm skhynix device manager 없이 실행**

**-i, --interval time\_interval 요약파일 생성 window의 크기(Seconds) 지정**

**-D, --daemon 데몬 프로세스로 실행**

**-b, --buffer buffer\_size 추적결과 저장 buffer 크기 지정**

**-p, --parse\_size file\_size ioparser 동작 주기**

**(추적 결과 파일의 특정 크기)**

**-w, --parse\_window time ioparser 동작 주기 (시간)**

**-t, --parse\_output path ioparser 결과 파일 저장 경로**

**<watch directory> IO 추적 대상 경로**

**2) iotracer 옵션 세부 내용**

**- output 옵션 : trace한 정보와 iotracer의 로그를 저장하는 \_\_디렉토리\_\_**

**⋇생성되는 파일들**

* **dmdu.bin : 디스크 사용량 및 드라이버 매니저 데이터 (바이너리 형태). 시간 및 크기에 따라 여러 개의 파일로 나뉘어짐**
* **trace.bin : tracepoint 에서 추출한 데이터 (바이너리 형태). 역시 여러 파일에 분할되어 저장됨**
* **iotrace.log : iotracer 상태 정보 (텍스트 형태)**

**- dm 옵션 : skhynix에서 제공하는 driver manager의 절대 경로**

* **dmdu.bin에 드라이버 매니저를 통해 수집한 데이터 저장**
* **disable\_dm 옵션을 사용해 driver manager 기능 비활성화 가능**

**- disable\_dm 옵션 : driver manager 비 활성화**

**- interval 옵션 : driver manager 및 디스크 사용량 측정 시간 간격, 초단위**

**- buffer : dmdu.bin, trace.bin 에 수집한 데이터를 쓰기전에 사용하는 버퍼 크기**

**- parse\_size : ioparser를 주기적으로 동작 할 크기 (단위 : 바이트)**

**⋇ 주의 : 1GB 이상 사용 할 것**

**- parse\_window : ioparser를 주기적으로 동작 할 시간**

**- parse\_output : ioparser 동작 결과를 저장할 디렉토리**

**- <watch directory> : 추적하려는 NVMe 장치의 mount point, 즉, 디렉토리**

**3) ioparser 옵션 및 파라미터**

**Usage: [OPTION...] <input> <output>**

**-t, --threshold value set threshold**

**-w, --window value window size**

**-a, --append enable append**

**<input> log directory**

**<output> output directory**

**3) ioparser 옵션 세부 내용**

**- threshold 옵션 : 99.99, 99.9999 등 설정**

**- window : 요약 정보를 만드는 윈도우 크기, 초 단위**

**⋇주의 : iotracer의 -i 옵션과 유사하나 다른 옵션**

**- append : 기존 결과 파일에 이어서 저장**

**- <input> : iotracer 에서 지정한 output 경로**

**- <output> : input directory를 해석해서 저장하는 output 경로**

**⋇생성되는 파일들**

* **dmdu.csv : 디바이스 매니저를 통해 수집한 데이터 및 디스크 사용량 데이터**
* **flow.csv : BIO 이벤트 추적 결과 저장**
* **vfs\_read\_window.csv : vfs 계층 읽기 연산 추적 결과 요약 저장**
* **vfs\_write\_window.csv : vfs 계층 쓰기 연산 추적 결과 요약 저장**
* **r\_window.csv : 블록 계층 읽기 연산 추적 결과 요약 저장**
* **d\_window.csv : 블록 계층 discard 연산 추적 결과 요약 저장**
* **w\_window.csv : 블록 계층 쓰기 연산 추적 결과 요약 저장**
* **process.csv : IO 발생 프로세스 기록**
* **vfs\_read.csv : vfs 계층 읽기 연산 추적 결과 저장**
* **vfs\_write.csv : vfs 계층 쓰기 연산 추적 결과 저장**