

이벤트 처리 대기열 차단됨

적용 가능성

Splunk Enterprise

설명

일부 데이터는 인덱서에 직접 온보딩되며 일부 소스는 무거운 전달자를 통해 데이터를 전송합니다.

조사

1. splunkd.log에는 몇 가지 소스 유형에 대한 줄 바꿈과 AggregatorMiningProcessor 오류가 있습니다.

```
WARN AggregatorMiningProcessor - Breaking event because limit of 256 has been exceeded
data_sourcetype="sourcetype_name"WARN AggregatorMiningProcessor - Breaking event because
data_sourcetype="other_sourcetype_name"
```

2. [Indexing Performance Dashboard\(인덱싱 성능 대시보드\)](#)에는 "Aggregator" Splunk Processor에서 CPU가 많이 사용되고 있는 것으로 표시됩니다.

3. 오류를 발생시킨 소스 유형을 수정해도 문제가 해결되지 않았습니다(오류는 사라지고 문제는 남아 있음).

4. 합산 대기열에 병목 현상이 있습니다. pstack은 합산 프로세서가 라인을 하나의 이벤트로 병합하여 병목 현상을 생성하느라 바쁘다는 것을 보여줍니다.

이 문제는 집계 프로세서가 많은 행을 단일 이벤트로 병합할 때 발생합니다. 코드에서 현재 이벤트로 병합해야 하는 행이 하나 있을 때마다 문자열::reserve와 다음 코드로 메모리 이동이 발생합니다. 따라서 한 이벤트에 대해 1000개의 라인이 있는 경우 1000배의 예약 및 memmovecalled가 발생합니다.

```
void CowPipelineData::appendAndReset(CowPipelineData& other)
{
```

```

...
    _pd->_rawAsStr.reserve(_pd->_rawSeg.size() + other._pd->_rawSeg.size() + 1);
    append_other_raw:
    _pd->_rawAsStr.push_back('\n');
    _pd->_rawAsStr.append(other._pd->_rawSeg);
...
}

```

라인 수가 많은 샘플 이벤트 중 일부를 수집하여 어떤 이벤트가 문제를 일으키는지 파악하고 총 라인 수가 많은 상위 3개 sourcetype을 식별합니다.

```

| tstats count where index=* OR index=_* by linecount sourcetype
| where linecount > 2
| stats sum(eval(linecount*count)) as totallines by sourcetype
| sort -totallines

```

개념:

그 이유는 다음과 같습니다.

- 일반적으로 구문 분석 대기열을 첫 번째 대기열로 사용하고 집계 대기열을 두 번째 대기열로 사용합니다.
- 구문 분석 대기열은 LINE_BREAKER가 정의되지 않은 경우 스트리밍 데이터를 기본 문자 \r\n에 따라 행으로 구문 분석하는 것입니다.
- 그러면 합산 대기열은 TIME_FORMAT 등의 구성을 기준으로 행을 병합합니다
- LINE_BREAKER를 설정하면 Splunk에게 스트리밍 데이터를 구문 분석 단계에서 행으로 구문 분석하는 방법을 알려주고
- SHOULD_LINEMERGE = false로 설정하면 집계 프로세스에서 행을 이벤트로 병합하지 않아야 합니다.
- 따라서 기본적으로 LINE_BREAKER 및 SHOULD_LINEMERGE = false가 고객의 문제를 해결할 수 있습니다.

해결

1. limits.conf를 다음과 같이 변경하여 디버깅을 위해 agg_processor에 대한 CPU 시간 메트릭을 활성화합니다.

```
.../limits.conf
[default]
agg_cpu_profiling = true
```

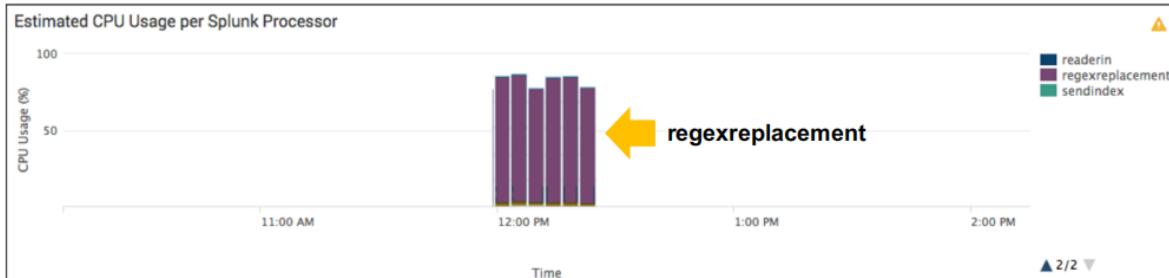
2. 아래 검색을 실행하여 어그리게이터 프로세서에 큰 부담을 주는 소스 유형을 식별합니다.

```
index=_internal source=*metrics.log* per_sourcetype_agg_cpu Metrics
| timechart span=10s avg(cpu) by series
```

3. SHOULD_LINEMERGE를 "false"로 설정하고 검색 그래프에 들어오는 각 소스 유형에 대해 "LINE_BREAKER"에 대한 regex를 작성합니다. Exp는 트러블슈팅 교육 문서에서 가져온 아래와 같습니다.

참고: 스크린샷은 regex 프로세서에 대한 검색을 보여줍니다.

MC > Indexing > Performance > Indexing Performance: Instance



- Search for specific hosts / sources / sourcetypes to find duplicate stanzas for possible anomalies and precedence issues
- Enable CPU time metrics for **RegexProcessor** for debugging purposes

limits.conf

```
[default]
regex_cpu_profiling = true

[metrics]
maxseries = 100
```

4. 모든 소스 유형을 수정하면 대기열 채우기 문제가 해결됩니다.

추가 리소스:

[커뮤니티 사무실](#) 시간에 나오는 다음 비디오에서는 차단된 대기열을 해결하는 방법에 대해 자세히

설명합니다.

관련 링크

Limits.conf 수동

<https://docs.splunk.com/Documentation/Splunk/7.3.0/Admin/Limitsconf>

이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.