기술

# 설비 데이터 수집 서버가 느려지면 메시지 큐부터 넣어야 할까, 웨이스가 수집기 200대로 재 보니 원인은 값마다 저장하는 방식이었습니다

**방동걸** (주)웨이스 대표이사
2026년 10월 29일

설비 데이터를 모으는 서버가 느려지면 가장 먼저 나오는 처방은 메시지 큐입니다. Kafka 같은 큐를 앞에 두면 몰려드는 값을 받아 두었다가 차례로 저장할 수 있으니, 느린 수집 서버에는 큐가 답이라는 이야기가 자연스럽게 나옵니다. 하지만 큐는 서버와 운영 인력이 하나씩 더 붙는 구조라 넣기 전에 무엇이 실제로 막히는지부터 확인해야 합니다.

제조 AI 운영체제 VEXPLOR를 만드는 웨이스는 2026년 10월 이 질문을 실측으로 확인했습니다. 공장 설비 값을 읽어 서버로 보내는 수집기 코드를 그대로 띄워 1대에서 200대까지 늘리면서 서버가 어디서 멈추는지 쟀습니다. 막힌 원인은 큐가 없어서가 아니었습니다. 서버가 값 하나마다 데이터베이스에 따로 쓰고 따로 확정하는 방식이 상한을 정하고 있었습니다. 이 쓰기 방식을 묶음 단위로 바꾸자 같은 기계에서 받을 수 있는 양이 4배 넘게 늘었습니다.

01

## 시험은 실제 수집기 코드와 작은 공장 서버 한 대 규모로 했습니다

시험 대상은 웨이스가 오픈소스(Apache-2.0)로 공개한 VEXPLOR AAS Studio의 원격 수집 기능입니다. 공장에 놓인 수집기(VEXPLOR Edge)가 설비 값을 읽어 5초마다 묶음으로 서버에 보내고, 서버는 그 값을 시계열 데이터베이스(TimescaleDB)에 저장합니다.

- **부하 발생기** — 실제 수집기 코드를 그대로 띄우고 설비 쪽만 값을 만들어 주는 가짜 장치로 바꿨습니다. 수집기 등록, 서명, 디스크 대기열, 재전송 규칙은 운영과 같습니다.
- **서버 규모** — 서버와 데이터베이스가 가상 머신 하나(vCPU 4개, 메모리 4GiB)를 함께 사용합니다. 작은 공장에 서버 한 대를 둔 경우와 비슷합니다.
- **단계** — 수집기를 1대, 10대, 50대, 100대, 200대로 늘렸습니다. 단계마다 180초를 돌리고 앞 30초를 뺀 150초를 쟀으며, 같은 단계를 두 번씩 반복했습니다.
- **판정 기준** — 응답 시간 상위 5%(p95)가 1초 안인지, 그리고 보낸 값을 서버가 다 받아 수집기 대기열이 늘지 않는지 함께 봤습니다.

태그 수와 읽기 주기를 바꾼 설정 4가지를 돌렸습니다. 태그 100개를 1초마다 읽는 수집기는 초당 100값을, 태그 1,000개를 1초마다 읽는 수집기는 초당 1,000값을 보냅니다.

02

## 수집기를 늘려도 서버는 초당 1만 1천에서 1만 2천 값에서 멈췄습니다

묶음 크기가 다른 설정 3가지가 모두 같은 지점에서 더 받지 못했습니다.

| 설정 | 다 받은 마지막 단계 | 처음 못 받은 단계 |
|---|---|---|
| **태그 100 × 1초** | 100대, 초당 1만 값 | 200대, 9,341~11,948값 |
| **태그 1,000 × 5초** | 50대, 초당 1만 값 | 100대, 11,629~12,001값 |
| **태그 1,000 × 1초** | 10대, 약 9,800값 | 25대, 11,131~11,450값 |

묶음 하나에 500값을 담든 5,000값을 담든 상한은 초당 약 1만 1천~1만 2천 값이었습니다. 묶음 크기가 아니라 **값의 개수가 상한을 정한다**는 뜻입니다. 상한을 넘자 서버는 요청을 거절하지 않고 느려졌고, 그 몫은 수집기 대기열에 쌓였습니다. 그러자 화면에서 값이 조회되기까지 걸리는 시간이 평소 몇 초에서 1분 넘게, 길게는 90초 넘게까지 늘었습니다.

CPU는 차지 않았습니다. 상한에서 서버는 0.36코어, 데이터베이스는 약 2코어를 썼습니다. 두 몫을 더해도 약 2.4코어로 4코어가 다 차지 않았으니, 장비가 모자라서 멈춘 것은 아닙니다.

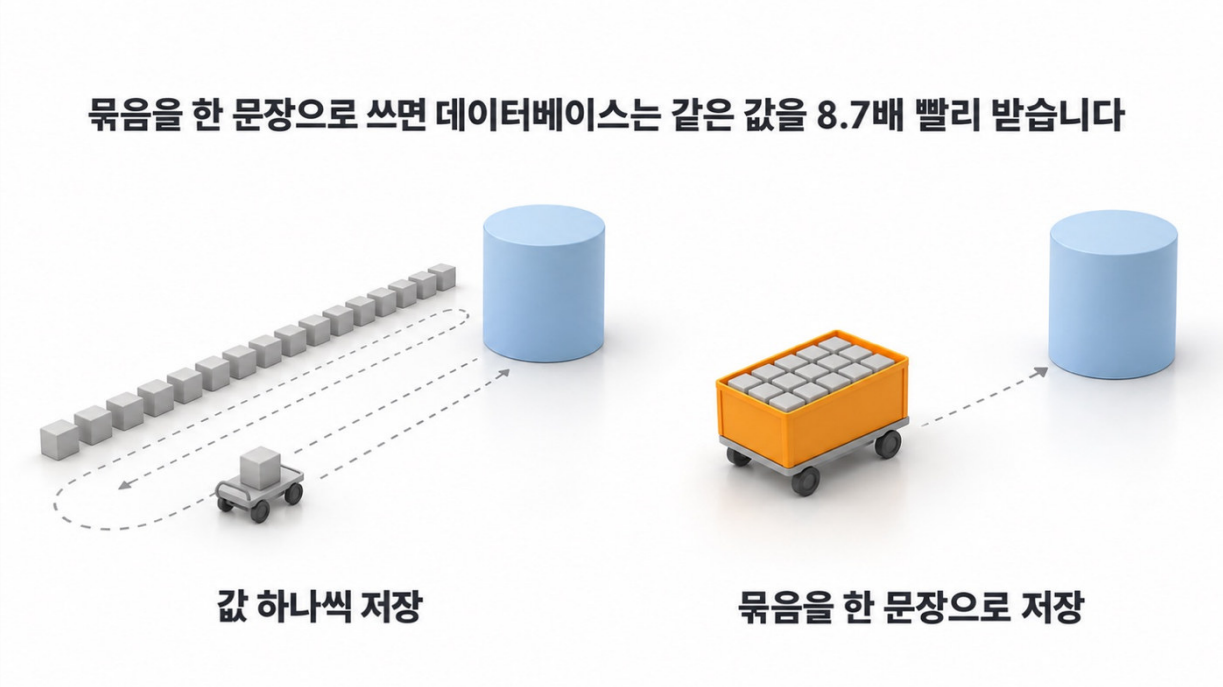
03

## 상한을 정한 것은 값 하나마다 저장하고 확정하는 코드였습니다

서버 코드를 따라가 보니 수집기가 보낸 묶음을 풀어 값 하나마다 INSERT 문을 따로 보내고 있었습니다. 트랜잭션으로 묶지 않아 값마다 커밋도 따로 일어났습니다. 5,000값짜리 묶음 하나를 저장하려면 데이터베이스를 5,000번 오가야 했고, 값 하나에 약 0.26ms가 걸려 묶음 하나에 1.3초가 들었습니다.

같은 데이터베이스와 같은 표 구조에서 두 방식을 따로 쟀습니다.

| 쓰기 방식 | 초당 저장한 값 | 5,000값 묶음 저장(동시 연결 10개) |
|---|---|---|
| **값 하나씩 저장** | 1만 9,800값 | 2.5초 |
| **묶음을 한 문장으로 저장** | 17만 2,000값 | 0.28초 |
| **차이** | 8.7배 | 약 9분의 1 |




값을 하나씩 저장하면 데이터베이스를 5,000번 오가고, 묶음으로 저장하면 한 번에 끝납니다

묶음을 한 트랜잭션 안에서 한 문장(`INSERT ... SELECT FROM unnest(...)`)으로 쓰면 데이터베이스는 같은 값을 8.7배 빨리 받습니다. 큐 없이 코드 몇 군데만 고쳐서 얻을 수 있는 여유였습니다. 비교하려고 데이터베이스 없이 메모리에만 담는 서버로도 돌려 봤는데, 초당 2만 5천 값을 받으면서 서버가 쓴 CPU는 0.26코어였습니다. 서명 확인과 형식 검사 같은 서버 쪽 처리는 상한의 원인이 아니었습니다.

04

## 쓰기 방식을 고치자 같은 기계에서 상한이 초당 약 5만 값이 됐습니다

웨이스는 쓰기 방식을 묶음 단위로 바꾸고 묶음 하나의 모든 값과 「이 묶음을 받았다」는 기록을 한 트랜잭션으로 저장하도록 고친 뒤 VEXPLOR AAS Studio 2.8.1 버전으로 냈습니다. 같은 가상 머신, 같은 부하 발생기로 다시 쟀습니다.

| 같은 단계 | 2.8.0 응답 p95 | 2.8.1 응답 p95 |
|---|---|---|
| **수집기 100대, 초당 1만 값** | 1.1~1.3초 | 0.03~0.04초 |
| **수집기 200대, 초당 2만 값** | 다 못 받음 | 0.04~0.05초 |
| **수집기 400대, 초당 4만 값** | 시험 안 함 | 0.2~0.4초 |

| 구분 | 2.8.0 | 2.8.1 |
|---|---|---|
| **받을 수 있는 상한** | 초당 약 1만 1,500값 | 초당 약 5만 값 |
| **여유 있게 쓰는 선** | 초당 5,000값 | 초당 4만 값 |
| **5,000값 묶음 저장** | 1.28초 | 0.15~0.19초 |

이제 상한을 정하는 것은 서버 코드가 아니라 데이터베이스입니다. 빈 표에서 시작하면 초당 6만 값을 다 받았고, 표에 1,300만 행 넘게 쌓인 뒤에는 5만 값 근처에서 멈췄습니다. 연결 수를 10개에서 20개로 늘려 보기도 했지만 오히려 느려져 기본값 10개로 두었습니다.

시험 중에 큐와 상관없는 병목도 하나 찾았습니다. 수집기 200대가 태그 1,000개씩 등록된 상태에서 관리 화면의 수집기 목록을 한 번 불러오는 데 114초가 걸렸습니다. 태그가 100개씩일 때도 관리자가 그 화면을 열어 두기만 하면 수집 응답 p95가 0.27초에서 2.9초로 10배 넘게 느려졌습니다. 장치 수만큼 같은 계산을 되풀이하던 부분을 한 번만 하도록 고친 뒤 목록은 0.1초 안에 열렸습니다. 수집이 느리다는 증상의 원인이 수집 기능 밖에 있기도 합니다.

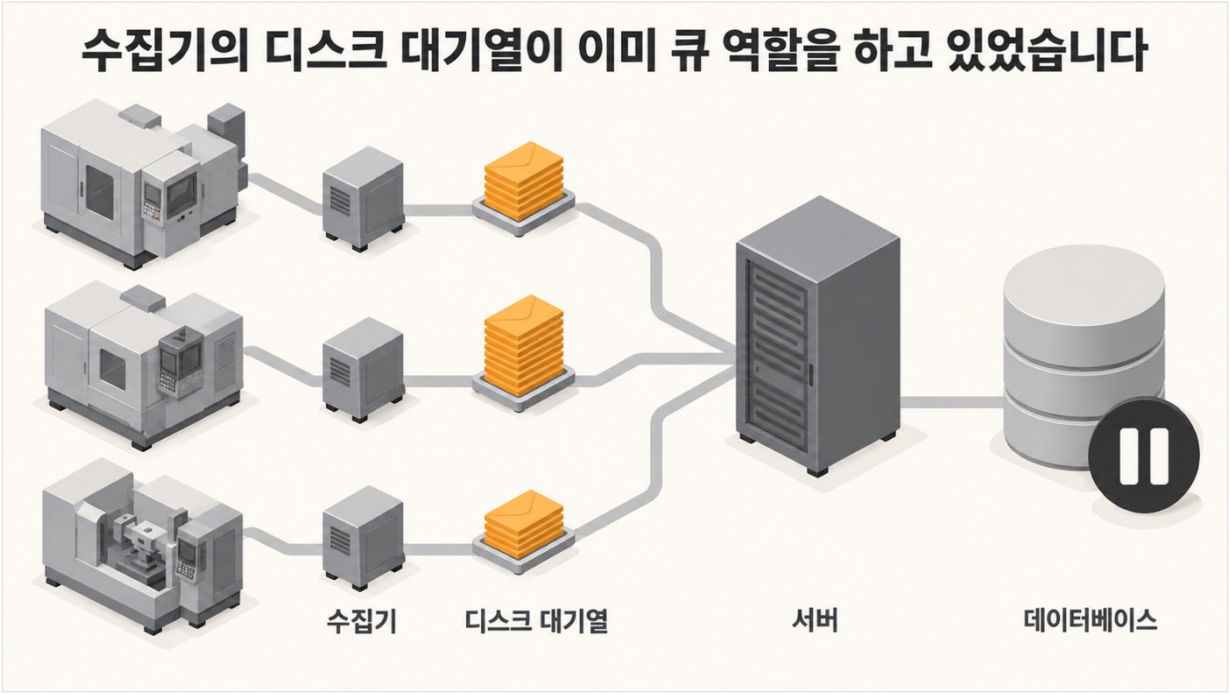
05

## 큐가 없어도 데이터베이스가 멈춘 동안 값은 하나도 빠지지 않았습니다

큐를 넣자는 또 하나의 이유는 장애 대비입니다. 데이터베이스가 멈추거나 망이 끊기면 값을 잃지 않겠느냐는 걱정입니다. 그래서 장애 2가지를 일부러 만들었습니다.

| 장애 시험 | 만든 값과 저장된 값 | 평소로 돌아오기까지 |
|---|---|---|
| **데이터베이스 30초 멈춤** | 117만 9,000 / 117만 9,000 | 재개 뒤 9.7초 |
| **수집기 50대 10분 단절 뒤 동시 재전송** | 627만 9,400 / 627만 9,400 | 풀린 뒤 35초 |

두 시험 모두 빠진 값 0, 중복 저장 0, 서버 오류 0이었습니다(2.8.1 기준, 2.8.0에서도 빠진 값과 중복은 0이었고, 재전송 시험의 회복에는 405초가 걸렸습니다). 수집기가 보내지 못한 묶음을 자기 디스크에 쌓아 두고, 서버가 같은 묶음을 다시 받으면 중복으로 알아보는 규칙이 있어서입니다. **수집기의 디스크 대기열이 이미 큐 역할을 하고 있었습니다.**




데이터베이스가 멈추는 동안 값은 수집기 디스크 대기열에 쌓였다가, 재개 뒤 순서대로 저장됩니다

06

## 메시지 큐는 이 세 경우에 넣습니다

시험 결과로 웨이스가 정한 기준은 이렇습니다. 이 시험 기계 기준으로 초당 4만 값까지는 큐 없이 지금 구조로 운영합니다. 수집기 400대가 태그 100개를 1초마다 보내거나, 수집기 200대가 태그 1,000개를 5초마다 보내는 규모입니다. 그 위로 가면 큐보다 데이터베이스의 코어와 메모리를 먼저 늘립니다. 이제 상한을 정하는 쪽이 데이터베이스이기 때문입니다.

큐가 필요한 경우는 다음 셋입니다.

- **데이터베이스를 키워도 받을 양을 넘을 때** — 수집기 수백 대가 태그 수천 개를 1초마다 보내는 규모입니다.
- **같은 값을 여러 곳이 실시간으로 받아야 할 때** — MES 연동, 알람 판정, 분석 서비스가 같은 흐름을 각자 읽어야 하면 큐를 가운데 두고 각자 꺼내 가는 편이 낫습니다. 지금 구조에서는 모두 데이터베이스를 다시 읽어야 합니다.
- **데이터베이스를 자주 멈춰야 하는데 그동안 응답이 늦어지면 안 될 때** — 다만 위 시험처럼 값은 이미 잃지 않으므로, 이것만으로는 큐를 도입할 이유가 약합니다.

큐를 넣더라도 수집기와 서버 사이는 HTTPS와 서명, 같은 묶음 재전송 규칙을 그대로 둡니다. 서버가 서명과 형식만 확인해 큐에 넣고, 저장과 그 밖의 용도는 각자 큐에서 읽어 가는 구조입니다.

07

## 수집 서버가 느릴 때 큐를 넣기 전에 확인할 5가지

같은 증상을 겪는 공장이라면 다음 순서로 확인하면 큐가 정말 필요한지 가릴 수 있습니다.

1. **상한이 묶음 크기와 상관없이 같은가** — 묶음 크기를 바꿔도 같은 값 개수에서 멈추면 쓰기 방식을 의심합니다.
2. **상한에서 CPU가 남는가** — 서버와 데이터베이스 코어가 남는데 멈추면 장비보다 코드 문제일 가능성이 큽니다.
3. **같은 데이터베이스에서 일괄 쓰기를 따로 재 봤는가** — 지금 방식과 묶음 한 문장 방식을 같은 표에서 비교하면 코드로 얻을 수 있는 여유가 바로 나옵니다.
4. **수집 밖의 요청이 수집을 느리게 하지 않는가** — 관리 화면, 보고서 조회처럼 같은 서버를 쓰는 다른 요청을 끄고 다시 잽니다.
5. **장애 때 값이 빠지거나 겹치는가** — 데이터베이스를 잠시 멈추고, 만든 값과 저장된 행 수, 중복 행 수를 맞춰 봅니다. 여기서 문제가 없으면 장애 대비만으로 큐를 넣을 이유는 없습니다.

이 시험이 보지 못한 것도 적어 둡니다. HTTPS 처리 비용은 앞단 프록시가 맡아 시험에 들어 있지 않고, 실제 서버처럼 코어와 디스크 속도가 다른 기계에서는 재지 않았습니다. 수천만 행이 넘는 큰 표에서 압축과 집계 작업이 장시간 더하는 부하도 아직 확인하지 않았습니다. 실제 서버에서는 같은 부하 발생기로 한 번 더 재서 그 기계의 상한을 정해야 합니다.




수집 서버가 느려진 원인과 고친 뒤 수치, 메시지 큐를 넣는 세 경우를 한 장으로 정리했습니다

느린 수집 서버에 큐를 먼저 넣으면 상한은 그대로인 채 관리할 서버만 늘어납니다. 큐는 쓰기 방식을 고치고 데이터베이스를 키운 뒤에도 양이 넘치거나, 같은 값을 여러 곳이 실시간으로 받아야 할 때 넣는 도구입니다.

WACE(웨이스)는 공장이 스스로 판단하고 멈추지 않게 하는 제조 AI 운영체제 VEXPLOR를 만드는 회사다.

08

## 함께 읽어보세요

- 웨이스는 왜 AAS 도구를 오픈소스로 공개했는가


| | |
|---|---|
| **원문** | blog.vexplor.com/post/data-collection-server-queue-or-batch-insert |
| **발행** | 2026년 10월 29일 · 방동걸 (주)웨이스 대표이사 |



| | |
|---|---|
| **회사** | 주식회사 웨이스 (WACE Inc.) · 대표이사 방동걸 · 사업자등록번호 439-87-02852 |
| **주소** | (18462) 경기 화성시 동탄구 동탄영천로 150, 에이동(A동) 1702·1706호 (영천동, 실리콘앨리동탄) |
| **연락처** | 전화 070-4730-0629 · contact@wace.me |
| **홈페이지** | wace.me · linkedin.com/company/wace-inc |


원문 홈페이지 링크드인