

데브옵스 전사적 확장 이끈다

성능·확장성·안정성 모두 갖춘 '퍼포스'에 주목할 이유



데브옵스 전사적 확장 이끈다

성능·확장성·안정성 모두 갖춘 ‘퍼포스’에 주목할 이유

한재정 수석 | 플래티어



“반도체를 생산하는 A 기업의 CIO는 최근 데브옵스를 전사적으로 확대했다. 그 과정에서 기존 사용하던 깃(Git)으로 대응되지 못하는 영역을 발견했다. 프로젝트 규모는 점점 더 커지는데 어떻게 대처할지 문제가 대두된 것. 3,000개까지 늘어난 리포지토리를 어떻게 관리할 것인가? 전 세계에 분산된 개발팀에 동일한 고성능을 어떻게 제공할 것인가? 대용량 바이너리 파일을 원활하게 관리하면서도 권한 관리, 성능까지 지원할 방안이 필요했다.”



“전장부품업체 B사는 내부 고객에서 외부 고객까지 고객 세그먼트를 넓히고 있다. 외부 고객사에 맞는 전장부품 설계를 진행하면서 케이던스 기반의 VersIC이라는 도구를 사용하게 됐고 대용량 파일 관리 문제에 직면하게 됐다. 몇 기가에 달하는 도면 설계 파일의 한 부분을 수정하고 또 다른 부분을 바꾸면서 계속 관리해야 했다. 이 회사는 대용량 파일 버전 관리가 절실한 상황이다.”

데브옵스 이니셔티브를 ‘시작’하는 것과 전사적으로 ‘확장’하는 것은 다른 문제다. 특히 형상 관리 측면에서 보자면 외부 참조 및 프로젝트 관련 리포지토리 등의 저장소가 폭발적으로 증가하기 때문이다. 또 버전 관리가 요구되는 수많은 파일이 생성될 것이고, 그 규모는 물론 크기도 엄청날 것이다.

엔터프라이즈 데브옵스(Enterprise DevOps)의 성공은 모든 유형의 자산을 ‘추적할 수 있도록 만드는 것(Trackable)’에 좌우된다고 해도 과언이 아니다. 실제로 기업들은, 그중에서도 반도체, 게임, 전장부품 업계 회사들은 엔터프라이즈 데브옵스를 도입하며 대용량 파일 관리, 원활한 성능 지원, 상세한 권한 관리 등에서 어려움을 겪고 있다.

엔터프라이즈 데브옵스의 당면 과제

데브옵스는 개발 기간을 단축하고 고품질의 제품 및 애플리케이션 등을 신속하게 배포하기 위해 개발과 운영을 통합하는 방법론이다. 데브옵스를 적절하게 그리고 성공적으로 활용한다면 기업의 디지털 트랜스포메이션을 가속화하고 비즈니스 민첩성, 탄력성, 유연성을 향상시킬 수 있다.

하지만 데브옵스를 소규모 팀 단위에서만 활용한다면 그 효과는 제한적일 수밖에 없다. 데브옵스에도 다다익선이 통한다. 더 많은 사용자와 프로젝트에 적용될 때 효과가 더 크다는 의미다. 즉 데브옵스가 전사적으로 확장돼야 그 이점을 크게 누릴 수 있다.

오픈소스 소프트웨어 형상 관리 도구 업체 퍼핏(Puppet)과 CI/CD 도구 업체 서클CI(Circle CI)가 발표한 '2020년 데브옵스 현황 보고서(2020 State of DevOps Report)'에 따르면 성공적으로 데브옵스를 확장한 기업에는 3가지 특징이 있는 것으로 드러났다. 바로 ▲내부 플랫폼 활용, ▲효과적인 변경 관리 프로세스 수립, ▲긴밀한 통합 보안이다.

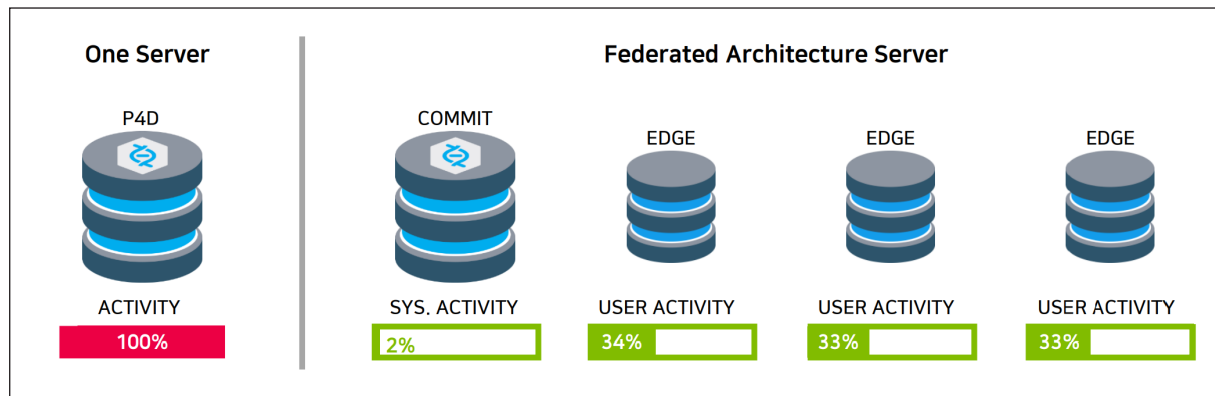
하지만 실질적으로 엔터프라이즈 레벨의 데브옵스를 적용하면서 이 3가지 특징을 구현하기란 쉽지 않다. 각종 시행착오를 겪는 기업이 많다. 그중에서도 엔터프라이즈 데브옵스에 적합한 형상 관리 플랫폼은 검토해봐야 할 주요한 부분이다.

데브옵스를 시작했고 이제 익숙해졌다고 판단한 기업이 전사적 확장을 고민하고 있다면? 형상 관리 측면에서 폭발적으로 증가하는 리포지토리, 버전 관리를 필요로 하는 수많은 파일(예: 소스 코드, 아티팩트, 구성 데이터, 바이너리 애셋, 설계 및 요구사항 문서 등)을 어떻게 관리할 것인가?

여기에 파일 크기까지 엄청나다면? 글로벌로 분산된 여러 팀이 함께 작업하는 대규모 프로젝트라면? 또 이에 부합하는 성능은 어떻게 지원할 것인가? 전사적으로 끊임없는 변화를 수용해야 하는 상황에서 변경 관리 프로세스를 일관되게 확장할 수 있게 하는 것은 엔터프라이즈 데브옵스에서 가장 어려운 부분 가운데 하나라고 할 수 있다.

데브옵스의 전사적 확장 과정에서 발생하는 이와 같은 문제들을 해결하는 '퍼포스(Perforce Helix Core)'를 알아보는 한편 이 솔루션이 성공적인 엔터프라이즈 데브옵스에서 어떤 의미를 갖는지 살펴본다.

그림 1 | 퍼포스의 페더레이티드 아키텍처 서버



'성공적인 데브옵스 확장'을 지원하다

형상 관리 측면에서 엔터프라이즈 데브옵스의 핵심 과제는 확장성이다. 대규모로 늘어난 리포지토리 및 대용량 파일, 글로벌로 분산된 사용자에게 대응하는 성능, 상세한 권한 관리, 폭넓은 호환성 등을 지원해야 하기 때문이다.

예를 들면 반도체 업체는 도면 설계를 하므로 생성되는 바이너리 파일 크기가 엄청나다. 전 세계적으로 분산돼 있는 개발팀이 함께 프로젝트를 진행하기도 한다. 대표적인 분산 버전 관리 시스템인 깃(Git)이면 충분하지 않겠나 싶지만 커버되지 못하는 영역이 명확하게 존재한다. 우선, 오픈소스 커뮤니티를 기반으로 태어난 깃(Git)에는 권한 관리 기능이 없다.

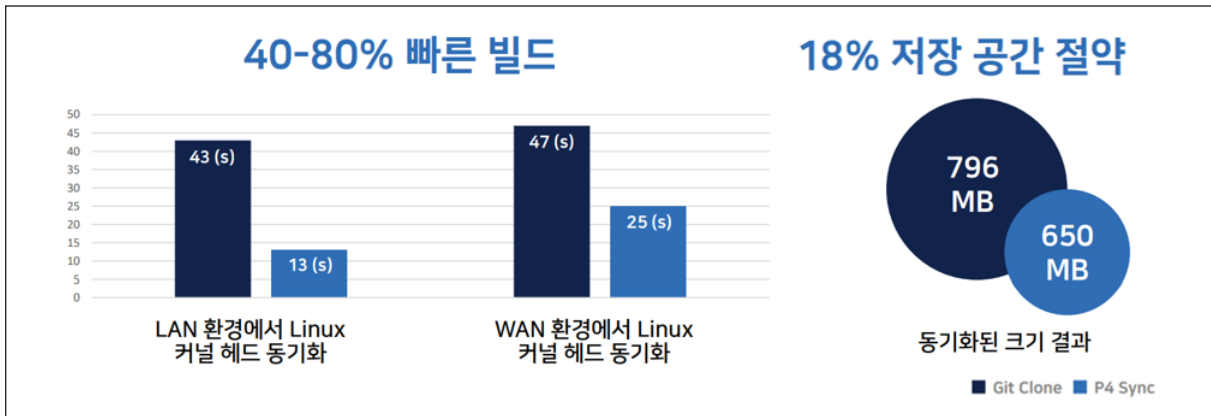
할 수 있다고 하더라도 파일 단위가 아닌 리포지토리 단위에서의 기본 접근 제어(모든 소스를 편집할 수 있거나 모든 소스를 아예 볼 수 없거나, All or Nothing)만이 가능하다. 엔터프라이즈 데브옵스에서 요구하는 '세부 파일별 상세 접근 권한 관리' 구성은 가능하지 않다.

깃(Git)은 현재 버전 상태의 Snapshot에 기반하는 구조적 특성으로 인해 **대용량 바이너리 파일 관리도 어렵다**. 또 프로젝트마다 규모를 키울 수 없어서 리포지토리를 계속 전환해야 한다. 대규모 프로젝트에 최적화된 통제가 어렵다는 이야기다.

플래티어(PLATEER)에서 제공하는 '퍼포스(Perforce Helix Core)'는 간단하게 보자면 중앙 집중형 버전 관리 시스템이자 데브옵스를 엔터프라이즈급으로 확장하려는 기업이 직면할 수 있는 문제를 해결하는 솔루션이다.

일반적으로 중앙집중형 버전 관리 시스템은 하나의 서버에 집중돼 부하 문제가 발생한다는 단점이 있다. 퍼포스에 주목해야 할 첫 번째 이유가 여기서 비롯된다. 퍼포스가 '페더레이티드 아키텍처(Federated Architecture)'를 기반으로 엔터프라이즈 데브옵스에 필요한 성능을 제공하기 때문이다.

그림 2 | 퍼포스의 CI 성능 개선 테스트 결과



* 모든 테스트는 1Gbps 링크에서 리눅스 커널의 일부 이력만 받아오는 얇은 복사(Shallow Clone)와 퍼포스 동기화의 병렬 스레드 4개를 사용해 수행됐다. WAN 테스트에서는 클라이언트와 서버 간에 200ms의 왕복 지연시간이 추가됐다.

페더레이티드 아키텍처는 워크로드를 하나의 서버가 아니라 여러 레플리카 멀티 노드로 분산시켜 모든 사용자에게 동일한 고성능을 지원한다. 조직 및 팀이 전 세계에 분산돼 있더라도 성능 이슈 없이 동기화, 파일 가져오기 등을 빠르고 원활하게 수행할 수 있다. 실제로 유럽, 아시아, 한국 등에 개발 부서를 가지고 있는 반도체 제조업체 C사는 글로벌 작업을 위해 퍼포스를 사용하고 있다.

퍼포스에서 자체적으로 테스트한 리눅스 커널 체크인/체크아웃 테스트 결과를 살펴보면 깃(Git)은 43초, 퍼포스(Perforce Helix Core)는 33초로 작업 처리 시간에 10초 이상 차이가 나는 것으로 나타났다. 동일한 조건으로 원격지에서 테스트한 결과도 퍼포스(25초)가 깃(47초)보다 더 빨랐다. 리포지토리에서의 파일 관리 크기도 퍼포스가 깃(796MB)보다 더 적은 크기(650MB)로 저장돼 대규모 CI에서 최소 40%에서 최대 80%가량 빠른 빌드, 18%의 저장공간 절약이 가능한 것으로 확인됐다.

이러한 성능 부분을 엔터프라이즈 데브옵스에 적용해 생각해보자. 퍼포스 마스터 서버를 구성하고 이를 기반으로 지역별로 분산된 복수의 복제노드(Replica), 빌드 전용 처리를 위한 노드(Build Farm), 빠른 파일 다운로드를 제공하기 위한 파일 캐시 노드(Proxy)를 추가하여 부하 분산을 시켜 놓는다.

이렇게 되면 지역에 상관없이 모든 글로벌 팀이 로컬에서 작업하는 것과 동일한 환경을 경험할 수 있다. 그러면서 사용자 관점에서는 여러 인스턴스를 조합하지 않고도 하나의 서버에서 모든 소스와 산출물을 관리할 수 있다.

또 퍼포스를 사용하면 빌드 환경과 아티팩트를 관리할 수 있다. 시스템 복잡도가 커지고

개별 구성요소가 많아질수록 개별 아티팩트 관리는 점점 더 중요해진다. 깃(Git) 사용자는 대용량 파일 지원 및 관리가 어렵다 보니 엔터프라이즈 데브옵스를 적용하려면 넥서스(Nexus), 제이프로그(JFrog) 등 별도의 아티팩트 관리 솔루션이 필요하다.

퍼포스를 사용하면 별도 솔루션을 구매할 필요 없이 모든 아티팩트를 소스 코드와 함께 동일한 리포지토리에서 관리할 수 있다. 이러한 아키텍처 부분을 퍼포스가 지원한다는 점에서 데브옵스를 전사적으로 확장해야 하는 곳이라면 이는 충분히 고려할 만한 선택지라고 할 수 있다.

쉽게 사용할 수 있는 인터페이스와 도구

성능도 성능이지만 유수의 반도체 및 게임 업체가 ‘퍼포스(Perforce Helix Core)’를 사용하는 또 다른 이유가 있다. 변경사항 등의 이력을 추적하는 건 물론이고 해당 사항을 한눈에 쉽게 파악할 수 있어서다. 이는 모든 이슈 및 변경사항, 작업 내역 등을 가시화해 상호 협업에 필요한 정보를 제공한다.

퍼포스는 수없이 많은 변경 히스토리를 그래픽을 사용해 상세하게 보여주는 기능(Revision Graph)을 제공한다. 예를 들어 낸드 플래시를 만든다고 가정해보자. 퍼포스를 사용하면 해당 설계 도면에 대해 대상 지역별, 세부 모델별, 요구 규격별로 제품 작업이 어떻게 진행되는지 그리고 변경점이 브랜치 사이에서 어떻게 이동했는지 등을 볼 수 있다. 다시 말해, 어떤 흐름으로 작업이 진행되는지, 무엇이 변경됐는지 등을 쉽게 확인할 수 있는 것이다.

전 세계에 분산된 개발팀이 함께 작업한다고 해도 현재 각 개발팀에서 작업 중인 모든 파일 상태를 확인할 수 있다. 다른 일반적인 솔루션에서는 작업이 완료된 후 서버에 Submit

업계에서 성능과 신뢰성을 이미 인정받다

게임 업계에서 널리 쓰이는 언리얼 엔진(Unreal Engine)은 공식 문서에서 퍼포스(Perforce Helix Core) 또는 SVN 사용을 가이드 하고 있다. 글로벌 게임 개발사 톱 20 가운데 무려 19곳(95%)이 퍼포스를 쓰고 있으며 닌텐도(Nintendo), 월트 디즈니(Walt Disney), 유비소프트(Ubisoft), 픽사(Pixar)가 대표적

인 고객사다. 반도체 업계에서도 케이던스(Cadence) 표준을 기반으로 칩을 만들 때 사용하는 도구(VersiC)에서 퍼포스를 기본으로 지원하며 반도체 업체 톱 10 중에 8곳(80%)이 퍼포스를 사용하고 있다. 성능 및 안정성 등을 중요시하는 업계의 특성을 감안한다면 퍼포스가 그만큼 신뢰도를 입증받았다는 뜻이다.

해야만 버전이 업데이트돼 이를 확인할 수 있지만 퍼포스는 아직 Submit 하지 않은, 로컬 PC에서 작업 중인 내용까지도 중앙 서버상에서 조회할 수 있기 때문에 각 개발팀에서 작업 중인 모든 파일의 상태를 확인할 수 있다. 즉 모든 사용자가 어떤 작업을 하는지 일목요연하게 볼 수 있고, 파일 단위까지 내려온 세부적인 권한을 관리할 수 있다.

사용자 필요에 맞춘 작업 경로 설정도 가능하다. 중앙 서버의 리포지토리에서 본인에게 필요한 일부 폴더만 선택적으로 로컬에 가져와서 작업할 수 있다. 같은 제품을 개발한다고 해도 미국, 유럽, 아시아 등 국가별 버전에 따라 워크스페이스(Workspace)를 여러 개 만들어서 작업할 수도 있다. 또 중앙집중형 버전 관리 시스템이지만 버전이나 파일 수준이 아니라 개발자의 관심사항, 즉 지금 하려고 하는 작업에 따라서 동시에 여러 체인지리스트(Changelist)를 관리하고 개발할 수 있다.

그림 3 | 퍼포스의 Revision Graph 사용 화면

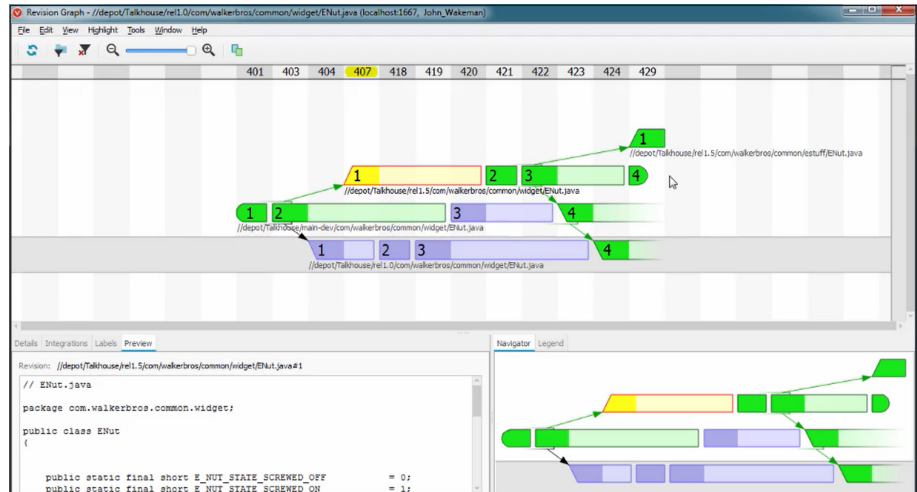
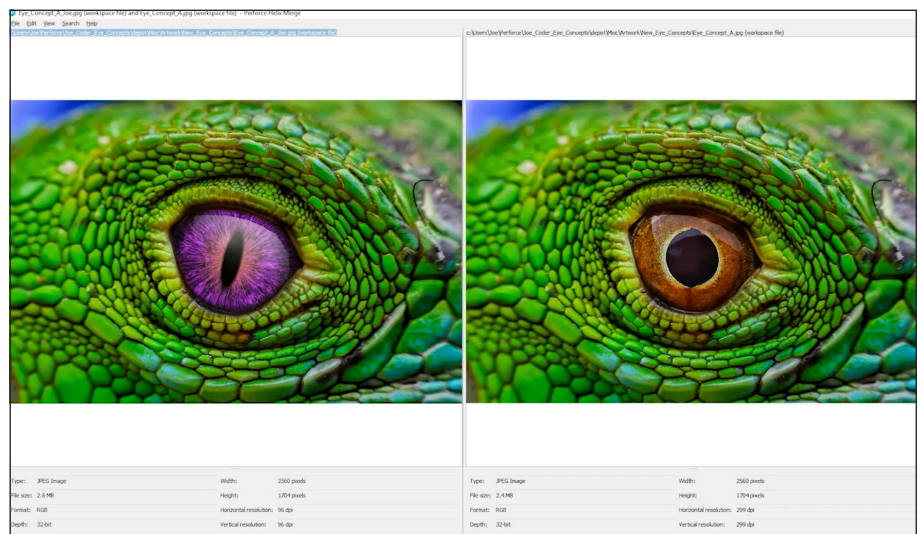


그림 4 | 퍼포스의 이미지 Diff 확인



이미지 관리와 이미지 Diff 기능을 사용하면 이전 버전과 신규 버전에서 변경된 부분을 마스킹하거나 하이라이트해서 픽셀 단위로 이미지 변경사항을 확인할 수 있다. 이는 설계 도면에도 동일하게 적용된다. 엄청나게 작은 수준의 변경사항까지 확인해야 하는 반도체 및 게임 업계에서 퍼포스가 널리 사용되는 이유이기도 하다.

퍼포스를 사용하는데 깃(Git)도 활용하고 싶다면? 퍼포스는 깃을 지원한다. 다른 오픈소스 솔루션이나 상용 깃(Git) 호스팅을 관리하지 않아도 퍼포스 자체에서 깃(Git) 호스팅을 병행해서 사용할 수 있다. 윈도우, 리눅스, 맥 전용 클라이언트, 디자이너용 도구, 개발 도구 플러그인 등도 지원된다.

이 밖에 주목해야 할 또 다른 부분이 있다면 쉽게 사용할 수 있는 인터페이스와 도구가 전문 개발 인력에만 한정되지 않아야 한다는 점이다. 데브옵스가 전사적으로 확장되면 개발자는 물론 비전문 개발자가 협력해 프로젝트를 진행하는 경우가 많기 때문에 중요한 요소다.

퍼포스에서는 그래픽 디자이너, 애니메이터 등 비전문 개발자도 시스템을 쉽게 사용할 수 있도록 버전 링크 클라이언트 도구 헬릭스 싱크(Helix Sync)를 제공한다. 새로운 도구를 쓸 필요 없이 비전문 개발자가 선호하는 도구(예: 마야, 3D 맥스, 포토샵 등)를 사용해 헬릭스 싱크에 연결하기만 하면 된다. 단순한 워크플로우인 드래그 앤드 드롭 방식으로 모든 유형의 자산을 버전화할 수 있다.

엔터프라이즈 데브옵스, 성공하려면 '확장성'에 집중하라

폐쇄적인 중앙집중형 버전 관리 시스템은 오픈된 커뮤니티 기반의 분산 버전 관리 시스템으로 넘어가기 시작하면서 도태될 것이라 예견되었었다. 실제로 많은 중앙집중형 시스템이 역사 속으로 사라지기도 했다. 하지만 분산 버전 관리 시스템은 대응하지 못하는, 중앙집중형 버전관리 시스템이 필요한 영역이 드러나기 시작했다.

깃(Git)은 오픈된 환경에서는 생산성이 좋지만 모든 버전을 로컬에 가지고 있기 때문에 통제할 수 없고 바이너리 관리에 취약하다는 단점이 있다. 엔터프라이즈급 버전 관리 및 형상 관리 차원에서는 이러한 이유로 허점이 생기게 된다. 따라서 반도체나 게임 업계에서는 이러한 문제를 퍼포스로 해소하면서 이 둘을 병행해 사용하고 있다.

결과적으로 대체라기보다는 서로 커버하는 영역이 다르기 때문에 병행해서 사용하는 형태로 자리 잡았다. 이는 엔터프라이즈 데브옵스 영역에서도 그대로 적용된다. 프로젝트 규모가 커지고 성능 이슈도 많아지며 글로벌로 작업이 이뤄지는 상황에서 '확장성'을 어떻게 원활하게 확보할 것인가? '퍼포스'가 그 적절한 해답이 될 수 있을 것이다.