

볼보 자동차와 오토리브(Autolive)가 합작 설립한 **제누이티(Zenuity)**는 딥 러닝기술을 이용하여 자율주행 소프트웨어를 개발하는 회사입니다. 제누이티는 NVIDIA® DGX-1™과 퓨어스토리지 플래시블레이드(FlashBlade™)에 기반한 AI 인프라스트럭처를 구축하여, 자사의 데이터 과학자와 연구자들이 더 스마트하고 안전한 자율주행차를 만들 수 있도록 지원하는 궁극적인 플랫폼을 마련했습니다.



비즈니스 혁신

제누이티는 AI 인프라스트럭처를 혁신하여 딥 러닝을 이용한 자율주행 분야에서 선두의 자리에 올랐습니다. 데이터 과학자들은 제품의 타임라인에 쫓기지 않고 안전 운전을 보장하는데 필요한 대용량 데이터를 처리할 수 있게 되었습니다.

지역

전 세계

산업

자동차

“ 전 세계 자율주행차 시장에서는 가장 먼저, 가장 안전한 자율주행 솔루션을 제공하기 위한 경쟁이 벌어지고 있다. NVIDIA DGX-1 과 퓨어스토리지 플래시블레이드 (FlashBlade)의 결합은 혁신적인 AI 인프라스트럭처를 제공해 우리가 선두를 차지할 수 있도록 해주었다.”

베니 닐슨(Benny Nilsson), 딥 러닝 관리자

미래의 자동차 개발

전 세계의 여러 기업이 우리가 여행하는 방식이나 상품이 전 세계로 배송되는 방식, 그리고 도시가 설계되는 방식까지 완전히 바꾸어 놓을 자동차 개발 경쟁을 하고 있으며, 기업들은 개발 과정에서 커다란 과제에 당면해 있습니다.

자율주행차를 성공적으로 개발하기 위해서는 인간만큼 정확한 의사결정을 실시간으로 내릴 수 있는 소프트웨어가 핵심적입니다. 기업에는 탑승자의 안전을 보장하는 것도 물론 중요하지만, 출시를 가속화하여 경쟁력을 확보하는 일 또한 중요합니다. 2025년까지 400억 달러에 달할 것으로 예상되는 자율주행차 시장에는 막대한 기회가 있습니다.

스웨덴에 본사를 두고 독일과 미국에 지사를 둔 제누이티는 인공 지능(AI)과 딥 러닝을 중심으로 한 소프트웨어 개발 전략을 통해 자동차 업계가 직면한 가장 큰 도전과제들을 해결해 나가고 있습니다. 볼보 자동차와 오토리브가 합작 설립한 제누이티는 자동차의 소프트웨어 ‘두뇌’를 개발하고, 오토리브는 이 기술을 전 세계 시장에 공급할 예정입니다.

DGX-1과 플래시블레이드(FLASHBLADE)를 통한 자율주행 가속화

제누이티의 딥 러닝 작업은 2015년 오토리브에서 시작된 프로젝트에 그 뿌리를 두고 있습니다. 제누이티의 딥 러닝 관리자인 베니 닐슨(Benny Nilsson)은 이 프로젝트에 참여했습니다. 닐슨은 “오토리브에 근무하며 딥 러닝을 위한 최고의 아키텍처에 대해 많은 토론을 하고 테스트했다” 며 “첫 번째 시스템은 GPU(그래픽처리장치)를 갖춘 상용 서버들과 AI 워크로드를 고려하지 않은 엔터프라이즈 스토리지 시스템이었다. 어떤 것이 가능하고 어떤 것이 불가능한지의 측면에서, 그 시스템으로부터 많은 것을 배울 수 있었다”고 회고했습니다.

GPU 기술에 대한 이전 경험은 제누이티가 최적의 딥 러닝 훈련 플랫폼을 선택할 때 많은 도움이 되었습니다. 제누이티는 자율주행 자동차를 개발하기 위해서 자동차에서 기하급수적으로 수집되는 각종 센서 데이터를 자사 모델 내에서 처리하고 훈련시키려면 획기적인 성능이 필요하다는 것을 인지하게 되었습니다. 그래야 연구자들이 중요한 개발 사이클을 놓치지 않고 신속하게 개발된 모델의 훈련을 반복할 수 있기 때문입니다. 또한, 자율주행 개발자들이 자율주행 소프트웨어 개발 이외에 개발을 빠르게 하기 위한 하드웨어와 소프트웨어를 설계 및 구축하고 문제를 진단하는 일에 시간을 낭비할 수 없다는 이유도 있었습니다.

제누이티는 이러한 니즈에 가장 적합한 연산 플랫폼을 발견했습니다. 바로 NVIDIA가 만든 DGX-1 딥 러닝 플랫폼이었습니다. 닐슨은 “AI와 딥 러닝에 관한 한 DGX-1은 완벽한 기준”이라고 말했습니다.

고객명:

Zenuity
<https://www.purestorage.com/kr/customers.html>

활용 사례:

- AI/딥 러닝 – NVIDIA DGX-1™, 퓨어스토리지 플래시블레이드 (FlashBlade™), Tensorflow, PyTorch

도전 과제:

- 중대한 결정을 실시간으로 내릴 수 있는 자율 주행 소프트웨어 개발
- 폭발적으로 증가하는 센서 데이터 처리
- 안전 주행 개선에 핵심적인 대량의 데이터 처리 및 시장 출시 가속화라는 두 가지 목표 달성

IT 혁신:

- 플래시블레이드(FlashBlade)의 소형 폼팩터와 간결성을 통해 8U에서 페타바이트 제공
- 훈련 실행 간격의 획기적인 감소
- 훈련 시간 감소 - IT 투자 효율성 및 데이터 과학자들의 생산성 향상

“ DGX-1과 플래시블레이드 (FlashBlade) 시스템의 무한한 확장성을 통해, 제누이티의 딥 러닝 인프라스트럭처는 미래에 대한 준비를 갖추었다.”

베니 닐슨(Benny Nilsson), 딥 러닝 관리자

DGX-1은 최고의 딥 러닝 훈련 성능을 제공함과 동시에 플러그 앤 플레이 방식으로 빠르게 딥 러닝 인프라스트럭처를 구성할 수 있으며, 간편하게 사용 및 관리할 수 있도록 설계되었습니다.

연산 플랫폼을 결정한 연구팀은 스토리지로 눈을 돌렸습니다. 닐슨은 “이전의 시스템에서는 스토리지의 중요성을 완전히 이해하지 못했다”며, “DGX-1을 테스트할 때, 기존 스토리지 시스템이 GPU에 충분한 데이터를 공급해주지 못했다. 데이터 병목현상이 발생한 후에야 AI 워크로드에 적합한 스토리지 시스템이 필요하다는 사실을 깨달았다”고 말했습니다.

제누이티의 인프라 전문가 올라 에릭슨(Ola Eriksson)이 스토리지 선정 절차를 감독했습니다. 그는 스토리지 공급업체 10곳을 선정하고, 제품의 성능을 비교하기 위해 실제 딥 러닝 워크로드를 모방한 엄격한 테스트 절차를 수립했습니다. 에릭슨은 “최우선 과제는 모든 DGX-1이 지속적으로 구동되도록 만드는 것이었다”고 말했습니다.

딥 러닝 팀의 연구 엔지니어인 사무엘 슈바이데거(Samuel Scheidegger)는 “DGX-1의 활용도는 데이터 과학자들의 생산성과 직결되기 때문에 스토리지 시스템을 구매하기 전, 실습 환경에서 테스트를 해 볼 수 있어야 했다”고 덧붙였습니다.

그는 핵심적인 선정 기준은 DGX-1의 활용률이었는데 이 활용률을 높이기 위해서는 충분한 대역폭을 제공할 수 있어야 한다는 것이다. “그래서 우리는 스토리지 시스템이 성능 저하(throttling) 없이 얼마나 많은 DGX-1들에 데이터를 공급할 수 있는가를 평가했다. 이러한 측면에서, 퓨어스토리지의 플래시블레이드(FlashBlade)가 단연 앞섰다”고 말했습니다.

플래시블레이드(FlashBlade)는 방대한 스케일아웃 기반의 I/O 처리를 통하여, 기존 스토리지 시스템의 스케일업 기반의 I/O처리 대비 성능의 병목이 제거되었습니다. 플래시블레이드(FlashBlade)는 AI와 딥 러닝에 필요한 성능과 간편성을 제공하도록 설계되어, 데이터 과학자들이 인프라스트럭처가 아니라 데이터에 집중할 수 있도록 해줍니다.

슈바이데거는 “소형 폼팩터와 간편성을 갖춘 플래시블레이드(FlashBlade)는 단 8U로 1페타바이트의 용량을 제공한다”고 말했습니다.

자동차를 더 스마트하고 안전하게 만들어 주는 역량 제공

딥 러닝은 엄청난 연산 능력과 방대한 양의 데이터가 요구됩니다. 이는 특히 자율주행차처럼 안전 운전을 위한 전방위 주시와 주시의 정확성을 요구하는 시스템에 해당됩니다. 이러한 측면에서, 제누이티의 IT 인프라스트럭처는 상당한 연산 능력이 필요했습니다.

주요 연산능력이 필요한 곳은 신경망 학습 과정으로, 이 과정에서 GPU는 테스트 차량에서 수집된 방대한 양의 데이터를 소비합니다. 데이터는 이미지, 레이더, LIDAR 데이터, 펄스 레이저 형태의 라이트와 관련된 탐색 방법으로 이루어집니다. 제누이티의 데이터 과학자들은 원하는 프레임워크를 자유롭게 선택할 수 있었으며, 그 중에서 가장 보편적인 프레임워크는 TensorFlow와 PyTorch였습니다.

증가하는 데이터를 더 많이 처리함으로써, 자율주행 소프트웨어는 지속적으로 학습하고, 시간이 지날수록 정확도가 높아집니다. 이에 따라 업데이트된 새로운 소프트웨어 버전이 자동차로 다운로드 됩니다. 그러나 이러한 신경망 훈련을 연속적으로 반복하는 데는 시간이 많이 걸립니다. 각 훈련을 위한 데이터세트는 테라바이트 규모에 달할 수 있으며, 환경이 적절하게 최적화되지 않으면 단일한 신경망 훈련에 몇 주가 소요될 수도 있습니다. 제누이티는 DGX-1과 플래시블레이드(FlashBlade)를 결합하여 이러한 문제를 해결하고 있습니다. 두 제품의 조합은 훈련 간격을 획기적으로 감소 시켜주어 데이터 과학자들이 모델에서 신속하게 훈련을 반복할 수 있도록 해주었습니다.

“ 플래시블레이드(FlashBlade)는 우리의 엄격한 딥 러닝 벤치마크를 충족한 유일한 스토리지 시스템이다.”

베니 닐슨(Benny Nilsson),
딥 러닝 관리자

닐슨은 “기업들은 뛰어난 안전성 향상을 위한 핵심적인 데이터 처리 역량과 제품의 시장 출시 가속화 사이에서 계속해서 줄다리기고 있다”며 “NVIDIA DGX-1과 퓨어스토리지 플래시블레이드(FlashBlade)를 결합함으로써, 데이터 과학자들은 이 두 제품의 장점만 취해, 제품 타임라인에 구애받지 않고 안전 주행을 위한 모델에 필요한 대량의 데이터를 처리할 수 있게 되었다”고 말했습니다.

제누이티는 현재 플래시블레이드(FlashBlade) 시스템 2대를 가동하고 있습니다. 각 시스템에는 52TB 블레이드 15개가 여러 대의 DGX-1에 연결되어 있으며, 세계에서 가장 앞선 데이터센터용 GPU인 Tesla V100 8대가 단일한 NVLink 기반의 하이브리드 큐브-메시 토폴로지로 설정되어 놀랄 만큼 빠른 GPU 간에 통신을 제공합니다. 플래시블레이드(FlashBlade)는 모든 DGX-1에 끊임없이 데이터를 제공하고도 성능이 남아, 제누이티 팀은 NVIDIA와 퓨어스토리지의 기술이 향후의 니즈도 충족해줄 것이라고 확신하고 있습니다.

닐슨은 “훈련 데이터 세트는 이미 수백 테라바이트 수준이며, 더 많은 데이터가 수집되고 더 많은 차량이 운행되면서 그 양은 폭발적으로 증가할 것이다”며 “DGX-1과 플래시블레이드(FlashBlade) 시스템의 무한한 확장성 덕분에, 제누이티의 딥 러닝 인프라스트럭처는 미래에 대한 준비를 갖추었다. 간편하게 스토리지 블레이드를 추가하기만 하면 용량이 확장되어 DGX의 지속적인 가동에 필요한 데이터를 문제없이 공급할 수 있다”고 설명했습니다.

미래를 향한 가속

DGX-1과 플래시블레이드(FlashBlade)로 구성된 AI 인프라스트럭처는 IT 투자를 완전하게 활용한다는 제누이티의 최우선 목표를 충족해주었습니다. 닐슨은 “GPU에 필요한 데이터를 공급할 수 없으면 가장 가치 있는 리소스를 데이터 과학자들을 제대로 활용할 수 없게 된다”며 “NVIDIA DGX-1과 퓨어스토리지 플래시블레이드(FlashBlade)는 훈련 시간을 단축해준다. 이는 IT 투자가 더 효과적으로 활용되고, 데이터 과학자들의 생산성이 보다 향상되어, 궁극적으로 더 행복해진다는 의미”라고 말했습니다.

제누이티가 중요하게 생각한 또 다른 고려사항은 시스템의 설치와 구현이 간단해야 한다는 것이었습니다. 에릭슨은 “DGX-1, 특히 플래시블레이드(FlashBlade)는 설치가 아주 간편하고 평소에 크게 신경을 쓸 필요가 없다”며 “이 둘은 신속하게 구현이 되며, 구현 후에는 별다른 지원이 필요 없다”고 말했습니다.

시간이 지나면서, 제누이티는 DGX-1 환경을 확장할 예정입니다. 증가하는 데이터 세트의 규모에 맞게 대처하기 위함입니다. 더 많은 양의 데이터를 처리하려면 다수의 GPU 연산 능력을 활용해야 하고, 더 빠르게 반복 학습을 통하여 새롭고 더 스마트한 자율주행 신경망 모델을 개발하려는 비즈니스의 부담도 늘어납니다. 제누이티의 DGX-1 기반 플랫폼이 성장함으로써, 더 광대한 인프라스트럭처는 페이스를 지속해야 할 것 입니다. 특히, 스토리지에 더 신경을 써야 할 것 입니다

닐슨은 “자율주행차를 위해 가장 안전한 솔루션을 제일 먼저 제공하기 위한 글로벌 경쟁이 벌어지고 있다”며 “NVIDIA DGX-1과 퓨어스토리지 플래시블레이드(FlashBlade)는 혁신적인 AI 인프라스트럭처를 제공해 제누이티가 선두를 차지할 수 있도록 해준다”고 말했습니다.



korea@purestorage.com

https://www.purestorage.com/kr/customers.html

