클릭률 예측 데이터 처리 및 모델 교육을 통해

NetApp Solutions

NetApp
April 20, 2024

This PDF was generated from https://docs.netapp.com/ko-kr/netapp-solutions/ai/aks-anf_libraries_for_data_processing_and_model_training.html on April 20, 2024. Always check docs.netapp.com for the latest.

목차

클릭률 예측 데이터 처리 및 모델 교육을 통해 . . . 1
데이터 처리 및 모델 교육을 위한 라이브러리 . . . 1
Pandas에서 Logs day 15를 로드하여 좌골키트을 훈련합니다. 무작위 포리스트 모델을 학습하십시오 . . . 1
Dask에서 Day 15를 로드하고 Dask cuML 무작위 포리스트 모델을 교육합니다 . . . 3
기본 작업 스트림 대시보드를 사용하여 Dask를 모니터링합니다 . . . 5
교육 시간 비교 . . . 6
Prometheus 및 Grafana로 Dask 및 RAPIDS를 모니터링합니다 . . . 7
NetApp DataOps 툴킷을 사용하여 데이터 세트 및 모델 버전 관리 . . . 7
Jupyter 노트북을 참조하십시오 . . . 7

클릭률 예측 데이터 처리 및 모델 교육을 통해

데이터 처리 및 모델 교육을 위한 라이브러리

다음 표에서는 이 작업을 만드는 데 사용된 라이브러리와 프레임워크를 보여 줍니다. 이러한 모든 구성 요소는 Azure의 역할 기반 액세스 및 보안 제어와 완벽하게 통합됩니다.

라이브러리/프레임워크	설명
Dask cuML	GPU에서 작업할 ML의 경우 "cuML 라이브러리" Dask를 사용하여 RAPIDS cuML 패키지에 대한 액세스를 제공합니다. RAPIDS cuML은 고성능 GPU 기반 구축을 통해 클러스터링, 차원 축소, 회귀 접근 방식을 비롯한 인기 있는 ML 알고리즘을 구현하여 CPU 기반 접근 방식에 비해 최대 100배 빠른 속도를 제공합니다.
Dask cuDF	CuDF에는 데이터 하위 설정, 변환, 핫 인코딩 등 GPU 가속 추출, 변환, 로드(ETL)를 지원하는 다양한 함수가 포함되어 있습니다. RAPIDS 팀은 을 유지합니다 "dask-cudf 라이브러리" 여기에는 Dask 및 cuDF를 사용하는 도우미 메서드가 포함됩니다.
Scikit 학습	Scikit-Learn은 견적기라고 하는 수십 가지의 기계 학습 알고리즘과 모델을 제공합니다. 각각 "평가자" 를 사용하여 일부 데이터에 장착할 수 있습니다 "맞춤" 방법.

비교를 위해 두 대의 노트북을 사용해 ML 파이프라인을 구축했으며, 하나는 기존의 Pandas scikit-learn 접근방식이고, 다른 하나는 RAPIDS 및 Dask를 사용한 분산 훈련입니다. 각 노트북을 개별적으로 테스트하여 시간과 규모의 측면에서 성능을 확인할 수 있습니다. RAPIDS 및 DASK를 사용한 분산 훈련의 이점을 설명하기 위해 각 노트북을 개별적으로 다룹니다.

Pandas에서 **Logs day 15**를 로드하여 좌골키트을 훈련합니다. 무작위 포리스트 모델을 학습하십시오

이 섹션에서는 Pandas 및 Dask DataFrames를 사용하여 Criteo Terabyte 데이터 세트에서 Click Logs 데이터를 로드하는 방법을 설명합니다. 이 사용 사례는 광고 교환을 위한 디지털 광고에서 광고 클릭 여부를 예측하여 사용자의 프로필을 작성하는 데 사용됩니다. 또한 교환에서 자동화된 파이프라인에서 정확한 모델을 사용하지 않는 경우도 해당됩니다.

Click Logs 데이터 세트에서 15일차 데이터를 로드하여 총 45GB를 기록했습니다. Jupyter Notebook CTR-PandasRF-Collated에서 다음 셀을 실행하면 처음 5000만 개의 행이 포함된 Pandas DataFrame을 생성하고 좌골키트학습 무작위 포리스트 모델을 생성합니다.

```
%%time
import pandas as pd
import numpy as np
header = ['col'+str(i) for i in range (1,41)] #note that according to
criteo, the first column in the dataset is Click Through (CT). Consist of
40 columns
first_row_taken = 50_000_000 # use this in pd.read_csv() if your compute
resource is limited.
# total number of rows in day15 is 20B
# take 50M rows
"""
Read data & display the following metrics:
1. Total number of rows per day
2. df loading time in the cluster
3. Train a random forest model
"""
df = pd.read_csv(file, nrows=first_row_taken, delimiter='\t',
names=header)
# take numerical columns
df_sliced = df.iloc[:, 0:14]
# split data into training and Y
Y = df_sliced.pop('col1') # first column is binary (click or not)
# change df_sliced data types & fillna
df_sliced = df_sliced.astype(np.float32).fillna(0)
from sklearn.ensemble import RandomForestClassifier
# Random Forest building parameters
# n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
rf_model = RandomForestClassifier(max_depth=max_depth,
n_estimators=n_trees)
rf_model.fit(df_sliced, Y)
```

훈련된 무작위 포리스트 모델을 사용하여 예측을 수행하려면 이 전자 필기장에서 다음 단락을 실행하십시오. 중복을 피하기 위해 15일째부터 마지막 100만 행을 테스트 세트로 테스트했습니다. 또한 이 셀은 예측의 정확도를 계산하고, 사용자가 광고를 클릭하는지 여부를 모델이 정확하게 예측한 발생 비율로 정의됩니다. 이 노트북에서 잘 모르는 구성 요소를 검토하려면 을 참조하십시오 "공식 좌골 키트 - 학습 문서".

```
# testing data, last 1M rows in day15
test_file = '/data/day_15_test'
with open(test_file) as g:
    print(g.readline())

# dataFrame processing for test data
test_df = pd.read_csv(test_file, delimiter='\t', names=header)
test_df_sliced = test_df.iloc[:, 0:14]
test_Y = test_df_sliced.pop('col1')
test_df_sliced = test_df_sliced.astype(np.float32).fillna(0)
# prediction & calculating error
pred_df = rf_model.predict(test_df_sliced)
from sklearn import metrics
# Model Accuracy
print("Accuracy:",metrics.accuracy_score(test_Y, pred_df))
```

Dask에서 Day 15를 로드하고 Dask cuML 무작위 포리스트 모델을 교육합니다

이전 섹션과 비슷한 방식으로, Pandas에서 Criteo Click Logs day 15 를 로드하고 좌골키트학습 무작위 포리스트 모델을 훈련합니다. 이 예에서는 Dask cuDF를 사용하여 DataFrame 로드를 수행하고 Dask cuML에서 임의의 포리스트 모델을 교육했습니다. 이 섹션에서는 교육 시간과 규모의 차이를 비교했습니다 ""교육 시간 비교.""

criteo_dask_rf.ipynb입니다

이 노트북은 다음 예와 같이 'numpy', 'cuml', 필요한 'dask' 라이브러리를 가져옵니다.

```
import cuml
from dask.distributed import Client, progress, wait
import dask_cudf
import numpy as np
import cudf
from cuml.dask.ensemble import RandomForestClassifier as cumlDaskRF
from cuml.dask.common import utils as dask_utils
```

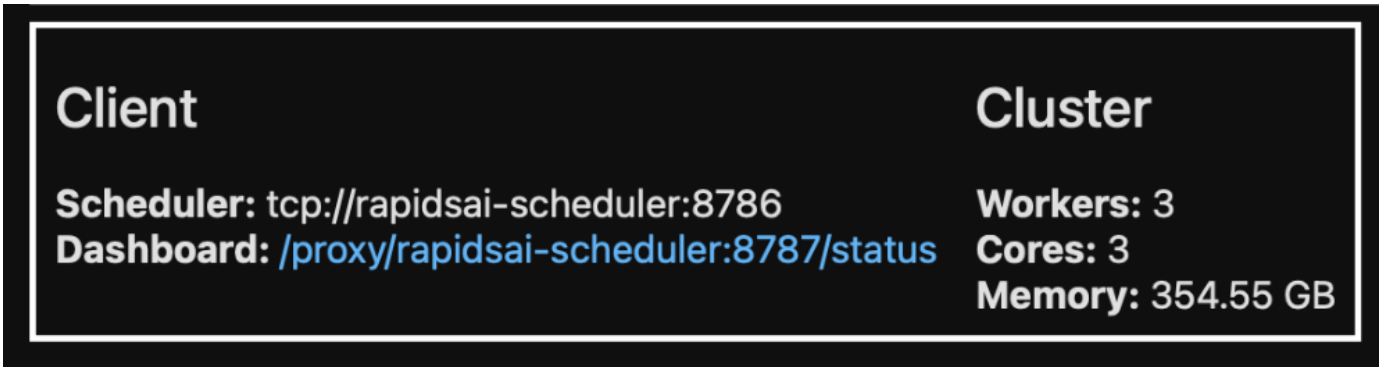
Dask 클라이언트()를 시작합니다.

```
client = Client()
```

클러스터가 올바르게 구성된 경우 작업자 노드의 상태를 확인할 수 있습니다.

```
client
workers = client.has_what().keys()
n_workers = len(workers)
n_streams = 8 # Performance optimization
```

AKS 클러스터에서 다음 상태가 표시됩니다.



DASK는 처리 코드를 즉시 실행하는 대신 실행 대신 실행 대신 대상 지정 DAG(Acyclic Graph)를 생성합니다. DAG에는 각 작업자가 실행해야 하는 일련의 작업과 상호 작용이 포함되어 있습니다. 이 레이아웃은 사용자가 Dask에서 한 가지 방식 또는 다른 방식으로 작업을 실행하도록 지시할 때까지 작업이 실행되지 않음을 의미합니다. Dask를 사용하면 다음과 같은 세 가지 주요 옵션을 사용할 수 있습니다.

- * DataFrame의 컴퓨팅()을 호출합니다. * 이 호출은 모든 파티션을 처리한 다음 결과를 스케줄러에 반환하여 최종 집계 및 cuDF DataFrame으로 변환합니다. 이 옵션은 스케줄러 노드의 메모리가 부족하지 않는 한 적은 결과에만 사용해야 합니다.
- * Call persist() on a DataFrame. * 이 호출은 그래프를 실행하지만 결과를 스케줄러 노드로 반환하는 대신 클러스터의 전체 노드를 메모리에 유지하여 사용자가 이러한 중간 결과를 다시 사용하지 않고도 파이프라인에서 재사용할 수 있도록 합니다.
- * DataFrame의 Call head(). * cuDF와 마찬가지로 이 호출은 10개의 레코드를 스케줄러 노드로 다시 반환합니다. 이 옵션을 사용하면 DataFrame 에 원하는 출력 형식이 포함되어 있는지 또는 레코드 자체가 타당한지 여부를 처리 및 계산에 따라 빠르게 확인할 수 있습니다.

따라서 사용자가 이러한 작업을 호출하지 않는 한 작업자는 스케줄러가 처리를 시작할 때까지 유휴 상태로 있습니다. 이러한 게으른 실행 패러다임은 Apache Spark와 같은 오늘날의 병렬적이고 분산된 컴퓨팅 프레임워크에서 흔히 볼 수 있습니다.

다음 단락에서는 분산 GPU 가속 컴퓨팅에 Dask cuML을 사용하여 임의 포리스트 모델을 교육하고 모델 예측 정확도를 계산합니다.

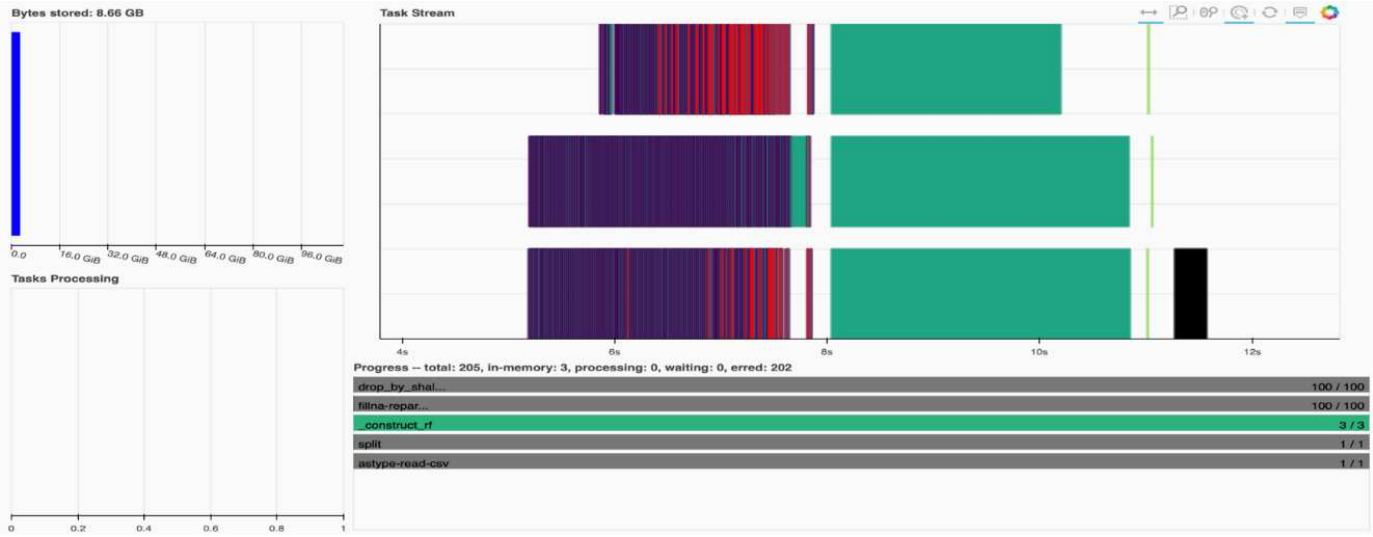
```
Adsf
# Random Forest building parameters
n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
cuml_model = cumlDaskRF(max_depth=max_depth, n_estimators=n_trees,
n_bins=n_bins, n_streams=n_streams, verbose=True, client=client)
cuml_model.fit(gdf_sliced_small, Y)
# Model prediction
pred_df = cuml_model.predict(gdf_test)
# calculate accuracy
cu_score = cuml.metrics.accuracy_score( test_y, pred_df )
```

기본 작업 스트림 대시보드를 사용하여 **Dask**를 모니터링합니다

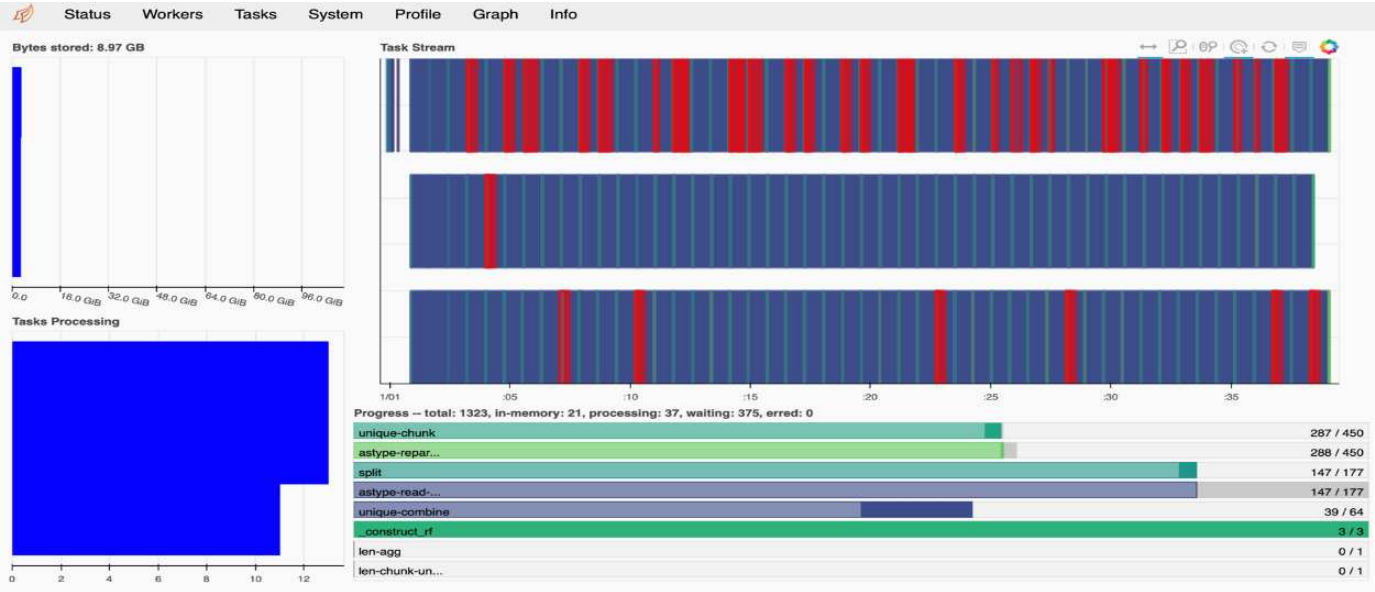
를 클릭합니다 "Dask 분산 스케줄러입니다" 다음과 같은 두 가지 형태로 실시간 피드백을 제공합니다.

- 실시간 정보가 포함된 여러 플롯과 테이블이 포함된 대화형 대시보드
- 콘솔 또는 노트북에서 대화형 사용에 적합한 진행률 표시줄

이 경우 다음 그림은 저장된 바이트, 스트림 수에 대한 자세한 분석 결과가 있는 작업 스트림, 실행된 관련 함수와 함께 작업 이름별 진행 상황 등을 포함하여 작업 진행 상황을 모니터링하는 방법을 보여 줍니다. 이 경우 작업자 노드가 3개이므로 스트림의 기본 청크가 3개 있고 색상 코드는 각 스트림 내의 서로 다른 작업을 나타냅니다.



개별 작업을 분석하고 실행 시간을 밀리초 단위로 점검하거나 장애물이나 장애물을 식별할 수 있는 옵션이 있습니다. 예를 들어 다음 그림에서는 랜덤 포리스트 모델 피팅 단계에 대한 작업 스트림을 보여 줍니다. DataFrame 프로세싱을 위한 고유한 청크, 랜덤 포리스트에 맞는 _Construct_RF 등 훨씬 더 많은 함수가 실행되고 있습니다. Criteo Click Logs에서 하루 동안 수집한 데이터의 크기가 45GB로 인해 대부분의 시간이 DataFrame 작업에 소비되었습니다.



교육 시간 비교

이 섹션에서는 기존 Pandas를 사용한 모델 교육 시간을 DASK와 비교합니다. Pandas의 경우, 메모리 오버플로를 방지하기 위해 처리 시간이 느려지기 때문에 더 적은 양의 데이터를 로드했습니다. 따라서 공정한 비교를 제공하기 위해 결과를 보간했습니다.

다음 표에서는 Pandas 무작위 포리스트 모델에 사용되는 데이터(데이터 세트의 15일 당 20억 개 중 5천만 개 행)가 상당히 적은 경우의 원시 교육 시간 비교를 보여 줍니다. 이 샘플은 사용 가능한 모든 데이터의 0.25% 이하만을 사용합니다. Dask-cuML의 경우 사용 가능한 모든 20억 행에 대해 무작위 포리스트 모델을 교육했습니다. 두 가지 접근 방식은 비슷한 훈련 시간을 낳았습니다.

접근 방식	교육 시간
Scikit-learn: 15일째에 50M 행만 교육 데이터로 사용합니다	47분 21초
RAPIDS-DASK: 15일 안에 20B 행을 모두 훈련 데이터로 사용	1시간, 12분, 11초

다음 표와 같이 교육 시간 결과를 선형적으로 보간할 경우 Dask와 함께 분산 훈련을 사용하면 큰 이점이 있습니다. 기존의 Pandas scikit-learn 접근 방식을 통해 클릭 로그를 하루에 45GB의 데이터를 처리하고 교육하는 데 13일이 걸리는 반면, RAPIDS-Dask 방식을 사용하면 같은 양의 데이터를 262.39배 더 빠르게 처리할 수 있습니다.

접근 방식	교육 시간
Scikit-learn: 15일째에 20B 행을 모두 훈련 데이터로 사용합니다	13일, 3시간, 40분, 11초
RAPIDS-DASK: 15일 안에 20B 행을 모두 훈련 데이터로 사용	1시간, 12분, 11초

이전 표에서 RAPIDS와 DASK를 사용하여 여러 GPU 인스턴스에 데이터 처리 및 모델 훈련을 분산하면, Scikit-learn 모델 훈련을 통해 기존 PandDataFrame 처리에 비해 실행 시간이 상당히 짧아진다는 것을 알 수 있습니다. 이 프레임워크를 통해 다중 노드, 다중 GPU 클러스터의 온프레미스뿐만 아니라 클라우드에서 스케일업 및 스케일아웃이 가능합니다.

Prometheus 및 Grafana로 Dask 및 RAPIDS를 모니터링합니다

모든 것이 배포된 후 새 데이터에 대한 추론을 실행합니다. 이 모델은 사용자가 검색 활동을 기반으로 광고를 클릭하는지 여부를 예측합니다. 예측 결과는 Dask cuDF에 저장됩니다. Prometheus를 사용하여 결과를 모니터링하고 Grafana 대시보드에서 시각화할 수 있습니다.

자세한 내용은 다음을 참조하십시오 "RAPIDS AI 중간 포스트".

NetApp DataOps 툴킷을 사용하여 데이터 세트 및 모델 버전 관리

Kubernetes용 NetApp DataOps 툴킷은 스토리지 리소스와 Kubernetes 워크로드를 데이터 과학 작업 공간 수준까지 추상화합니다. 이러한 기능은 데이터 과학자와 데이터 엔지니어를 위해 설계된 간단하고 사용하기 쉬운 인터페이스로 패키징되어 있습니다. 데이터 과학자와 엔지니어는 익숙한 형식의 Python 프로그램을 사용하여 JupyterLab 작업 공간을 단 몇 초 만에 프로비저닝 및 폐기할 수 있습니다. 이러한 작업 공간에는 테라바이트나 페타바이트급의 스토리지 용량이 포함될 수 있으므로 데이터 과학자는 모든 훈련 데이터 세트를 프로젝트 작업 공간에 직접 저장할 수 있습니다. 작업 영역과 데이터 볼륨을 별도로 관리하는 시대는 지났습니다.

자세한 내용은 툴킷 을 참조하십시오 "GitHub 리포지토리".

Jupyter 노트북을 참조하십시오

이 기술 보고서와 관련하여 2개의 Jupyter 노트북이 있습니다.

- "* CTR-PandasRF-Collated.ipynb. *" 이 노트북은 Criteo Terabyte Click Logs 데이터 세트에서 15일을 로드하고 Pandas DataFrame으로 데이터를 처리 및 포맷하고 Scikit-Learn 무작위 포리스트 모델을 교육하며 예측을 수행하고 정확도를 계산합니다.
- "criteo_dask_rf.ipynb. * 를 사용합니다" 이 전자 필기장은 Criteo Terabyte Click Logs 데이터 집합에서 15일을 로드하고, 데이터를 처리하여 Dask cuDF로 서식을 지정하고, Dask cuML 임의 포리스트 모델을 교육하고, 예측을 수행하고, 정확도를 계산합니다. GPU에 여러 작업자 노드를 활용함으로써 이러한 분산 데이터 및 모델 처리 및 교육 접근 방식이 매우 효율적입니다. 처리하는 데이터가 많을수록 기존 ML 방식에 비해 시간 절감 효과가 더 커집니다. 네트워킹 설정을 통해 데이터 및 모델 배포를 자유롭게 이동할 수 있는 한, 이 메모장을 클라우드, 온프레미스 또는 Kubernetes 클러스터에 다른 위치의 컴퓨팅 및 스토리지가 포함된 하이브리드 환경에 배포할 수 있습니다.

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 http://www.netapp.com/TM에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.