

DIFS

file system with key-value store
paper review

{uni2u}{labry}@etri.re.kr
Data-Centric Network
DIF part

NDNFS

What is ...

base version: <https://github.com/named-data/NDNFS>

FUSE version: <https://github.com/remap/ndnfs-port>

metadata version: <https://github.com/remap/ndnfs-port/tree/in-memory-key>



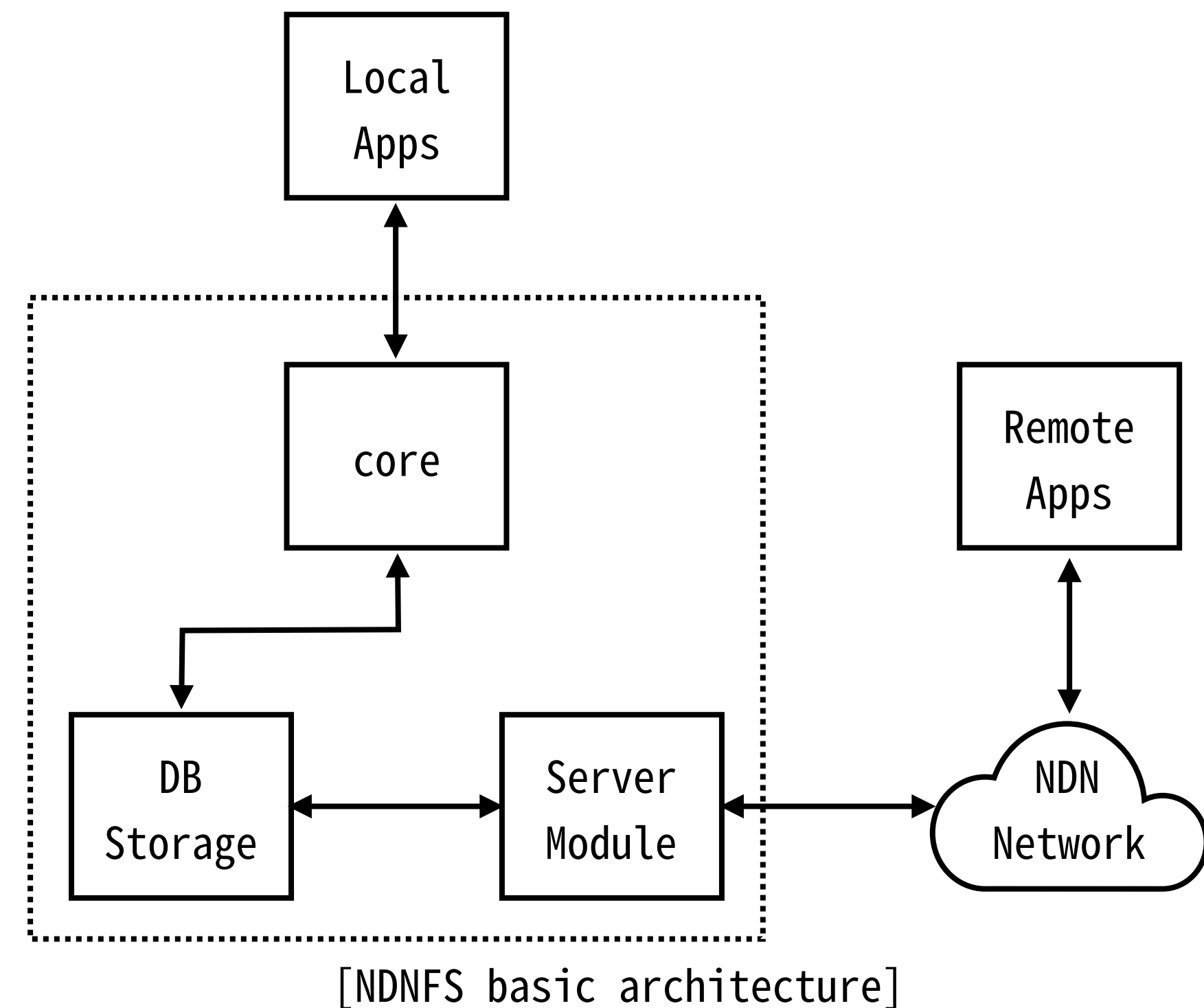
NDNFS: An NDN-friendly File System

- Basic Version (branch: master)
 - MAC OS X FUSE 2.5.6
 - MongoDB 2.4.5 (including its cpp library ver. 2.4.5)
 - ndn.cxx, boost library
- Basic Version (branch: sqlite)
 - MAC OS X FUSE
 - SQLite3
 - ndn.cxx, boost library



NDNFS architecture

- File System core
 - FUSE 커널 모듈 위에 NDNFS 구축
 - File operation: create, open, read, write, truncate, release, unlink
 - Directory operation: mkdir, rmdir, readdir
 - Meta-info operation: chmod, getattr
 - 파일 데이터 블록 및 meta 정보는 로컬 DB (SQLite) 에 저장
 - file entries table: 파일, meta 정보 저장
 - file version table: 파일 버전 정보 저장
 - file segment table: blob 저장
- Server Module
 - NDNFS core daemon 과 함께 실행 (producer 요청 처리)
 - NDNFS Global prefix 등록
 - Interest 수신: Interest name 에서 NDNFS 로컬 파일 경로 추출 후 추출한 경로를 키로 사용하여 SQLite 에 일치하는 데이터 검색
 - 일치하는 데이터 없는 경우 생성



File System view

```
./
├─ ./a
│  └─ ./a/b
│     └─ ./a/b/c.txt
```

prefix = /ndn/ucla.edu/irl/user/ndnfs

```
└─ ./a/b/c.txt/v2/s0 -> /<prefix>/a/b/c.txt/v2/s0
└─ ./a/b/c.txt/v2/s1 -> /<prefix>/a/b/c.txt/v2/s1
└─ ./a/b/c.txt/v2/s2 -> /<prefix>/a/b/c.txt/v2/s2
```



Storing file segments

- Storing file segments
 - NDN 데이터 패킷 segment 저장
 - segment 크기: 8,192 바이트
- Version control
 - 각 파일 항목에 대해 current, temporary 버전 유지 관리
 - git commit 과 비슷한 개념 (우선 temporary 버전으로 작업 후 완료되면 commit)
- Deletion
 - FreshnessSeconds 설정을 활용한 라우터 캐시 제거
- Metadata protocol
 - 파일 시스템 (파일/디렉토리 속성) 에 대한 정보

NDNFS git

- <https://github.com/named-data/NDNFS>
- master branch
 - mongoDB: metadata DB
- sqlite branch
 - sqlite: metadata DB
- directory
 - fs: file server core
 - server-js: for firefox
 - server: server module

NDN-friendly file system (based on FUSE)

96 commits

2 branches

0 releases

4 contributors

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

zhehaowang

readme: point to updated work under remap

Latest commit 0c94165 on 8 Oct 2015

fs	Allow space characters to be in the file name.	6 years ago
server-js	Capture SIGINT in server.js	6 years ago
server	Final cleanup	7 years ago
test	Final cleanup	7 years ago
.gitignore	Add waf-related files	7 years ago
README.md	readme: point to updated work under remap	4 years ago
waf	Add waf-related files	7 years ago
wscript	Move server/client to test/client	7 years ago

Branch: sqlite [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

This branch is 57 commits ahead, 1 commit behind master. [Pull request](#) [Compare](#)

qiuhanding

Fix test/client

Latest commit 81b6680 on 6 Mar 2014

fs	Switch to memory-based keychain	6 years ago
server	Switch to memory-based keychain	6 years ago
test	Fix test/client	6 years ago
waf-tools	Fix compilation errors. Not tested yet.	6 years ago
.gitignore	Fix compilation errors. Not tested yet.	6 years ago
INSTALL	INSTALL: For Max OS X, set PKG_CONFIG_PATH on the line with ./waf con...	6 years ago
README.md	INSTALL: Added download site for OSXFUSE.	6 years ago
waf	Fix waf to support protoc	6 years ago
wscript	Add pipeline tester.	6 years ago

NDNFS vs repo-ng

What is ...



Introduce and Abstract

- title: Analysis of NDN Repository Architecture and its Improvement for I/O Intensive Applications (paper; 2017)
- Analysis of NDN Repository Architectures (poster; 2016)
- Abstract
 - NDN 기본 연구와 달리 파일 시스템 연구는 부족함
 - NDNFS 및 repo-ng 를 제외하면 이렇다할 연구가 없음
 - 기존 연구는 I/O 집약적 환경의 파일 시스템에서 오버헤드 문제 발생
 - 오버헤드 발생 원인 분석과 해결책이 필요함
 - NDN 기반 분산 파일 시스템 구조 정의



Introduction

- IoT 환경에서 traffic 증가
 - IPv4 환경에서 사용하고 있는 콘텐츠 중복성 및 big-data 관리 솔루션이 필요함
 - 다양한 Device 의 모든 데이터가 대역폭 제한에 도달
 - NDN 의 cache 를 사용하여 트래픽 혼잡을 줄일 수 있음
- big file 을 repository 에 저장하면 성능 저하 발생
 - 제안된 NDN 파일 저장소에 대한 내용에서 파일 시스템에 write/read 시 성능 문제 발생
 - 최근 파일 저장소에 대한 연구보다 FileSync 형식의 연구가 진행되고 있음
- 주요 내용
 - 현재 NDN repository Architecture 및 I/O 집약적 환경에서 문제
 - I/O 집약적 환경을 위한 metadata 구성 및 관리
 - NDN 분산 파일 시스템 설계 고려 사항



Brief overview of NDN Repository

- NDNFS
 - FUSE (Filesystem in Userspace) 기반 파일 시스템
 - NDNFS 에 파일이 저장되면 세그먼트화 (NDNFS daemon 동작)
 - 특성
 - FUSE filesystem: “cp” 및 “mv” 등 OS 파일 작업 수행 가능
 - metadata management: 각 Segment 의 metadata 는 3 개의 테이블로 관리 (SQLite3)
 - extra tools: NDN.JS 를 활용한 Firefox 웹 브라우저 add-on 을 통하여 사용자가 파일 시스템을 search, download
- repo-ng
 - signature Interest 를 통한 read/insert/delete 제공 (repo-protocol 준수)
 - 모든 콘텐츠 (name, data) 는 SQLite 에 저장
 - 특성
 - data management: 데이터 및 name 은 SQLite 에 blob 형식으로 저장
 - signed Interest: write/read/delete 작업은 signature Interest 로 수행
 - extra tools: “ndnputfile” 및 “ndngetfile” 제공



Architecture analysis and comparison

- Initialization

- NDNFS

- 자체적으로 필요한 구성요소 설정



- NDNFS 실행파일 옵션: 파일 저장 디렉토리, 사용자 액세스 (마운트 지점), prefix name, 로그 파일 경로 및 위치

- metadata 의 key-chain 초기화 후 signature

- SQLite Table 생성: “file_system”, “file_version”, “file_segments”

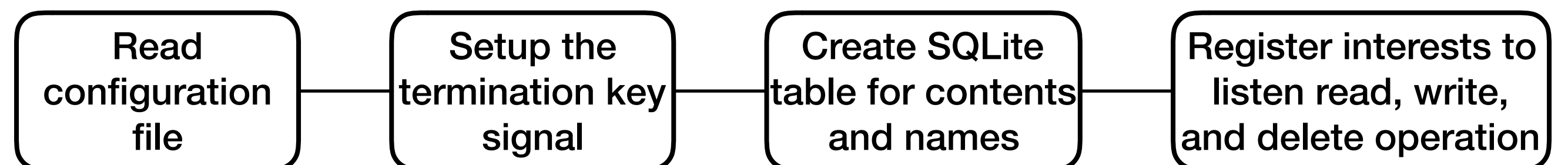
- 파일 정보 (파일 크기 등) 는 metadata table 에 기록

- segments 정보는 file_segments 테이블에 기록

- repo-ng

- config 파일을 통한 초기화

- SQLite 초기화 후 data 및 name 저장



- public key-chain 정보를 “keyLocatorHash” 에 저장



Architecture analysis and comparison

- File insertion
 - NDNFS
 - 대상 파일을 파일 시스템에 복사
 - FUSE 마운트 지점에 붙여 넣으면 NDNFS 는 동일한 파일이 있는지 검사 (file list 활용)
 - 파일 확장명 확인 후 MIME type 결정 및 SQLite 에 저장
 - 파일 write (4KB 크기 고정)
 - 버전 정보 write (SQLite)
 - 파일 write 완료 후 각 segments 에 signature 하여 SQLite 저장 (모든 데이터 패킷에 서명)
- repo-ng
 - OS 파일 write 명령이 아닌 NDN 기반 repo insert 명령 (Interest 활용)
 - insert Interest: signature 및 “StartBlockID”, “EndBlockID” 정보 또는 Selector 정보 포함
 - 소스 노드에 Interest 전송하여 데이터 패킷 수신



Architecture analysis and comparison

- Observations on SQLite tables
 - repo-ng: 3 가지 blob 유형 속성과 테이블의 primary key 컬럼
 - data attribute: 모든 데이터 패킷 reserve; blob type
 - name attribute: 데이터 name; blob type
 - keyLocatorHash: 데이터 패킷에서 signature 해독을 위한 키 정보; blob type
- NDNFS
 - repo-ng 의 name 과 동일한 path
 - signature attribute: blob type

Attribute	Type	Description
id	integer	primary key of the table
name	blob	NDN name for contents
data	blob	contents in binary format
keyLocatorHash	blob	public key information

[NDN_REPO DB schema of repo-ng]

Attribute	Type	Description
path	text	file path (equivalent to “name” in repo-ng)
version	integer	version information of each segment
segment	integer	number for each segment
signature	blob	signature of each segment

[FILE_SEGMENTS DB schema of NDNFS]



Discussion and Enhanced scheme

- SQLite 사용 (NDNFS, repo-ng)
 - Linux file system 에 비하여 매우 낮은 성능
 - 예상) SQLite 테이블에서 blob type 활용에 따른 문제로 예상함
- repo-ng 의 데이터 저장 알고리즘은 SQLite 에 의존; NDNFS 는 FUSE 활용
- repo-ng
 - SQLite 성능 향상을 위하여 contents, names, signature 등의 모든 데이터를 메모리에 저장
 - 이는 대용량 파일 insert 에 전체 시스템 성능 저하
- NDNFS
 - SQLite 에 signature 및 기타 데이터를 blob type 으로 저장
 - write 가 빈번한 경우 성능 저하
 - ext4 와 비교하여 심각한 병목 발생

Attribute	Type	Description
path	text	content path (“name” in repo-ng)
current_version	integer	content version
mime_type	text	mime type of a file
ready_signed	integer	each file;s signature state
type	integer	file type, the output of ls command

[FILE_SYSTEM DB schema of NDNFS]



Discussion and Enhanced scheme

- Problems of SQLite and its DB Tables
 - SQLite 의 알려진 문제점으로 인해 새로운 NDN 데이터 저장 scheme 필요
 - NDNFS 는 contents 의 metadata 를 저장하기 위하여 SQLite 활용
 - repo-ng 는 SQLite 에 모든 패킷 정보를 업데이트
 - write, renaming, copying, modifying 등 파일 작업 실행시 overhead 발생
 - 즉, 많은 속성 (attribute) 이 파일에 포함
 - path, version, mime_type 등의 속성 (파일 관련 속성)
 - I/O 집약적 환경에서 단점으로 작용되기 때문에 성능을 위해 SQLite 가 아닌 파일 자체로 재 배치 필요
 - repo-ng 는 contents 와 metadata 모두 blob 데이터 형식으로 저장



Discussion and Enhanced scheme

- Metadata Management with Xattr
 - NDN 프로토콜은 디스크에 contents 를 저장하기 위해 더 많은 metadata 생성
 - 파일 시스템에서 DB 사용 여부를 최소화 하여야 함
 - 파일별 metadata 는 DB 종속성을 감소 (metadata 관리 도구 필요성)
- Xattr 은 Key-Value database
 - signature 는 패킷당 발생하기 때문에 이를 제외한 나머지는 모두 metadata 파일에 저장
- Avoiding Signature Write
 - 패킷별 signature 를 대체할 새로운 metadata 구성 필요
 - 파일 시스템 (server) 과 writer (data producer) 사이에 완전한 신뢰, 인증 프로세스

Attribute	Type	Description
path	text	content path ("name" in repo-ng)
current_version	integer	content version
mime_type	text	mime type of a file
ready_signed	integer	each file;s signature state
type	integer	file type, the output of ls command
certificate	binary	producer's certificate for each content

[NEW METADATA Configuration for each content]





Performance Evaluation

- 실험 환경
 - Intel Xeon CPU 3.4Ghz (8코어)
 - 14GB Ram, 50GB HDD
 - ubuntu 15.10
- NDN 환경
 - 2015.11.21. 기준
 - NFD v0.4.0, ndn-cxx v0.4.0-beta2, NDNFSv0.3, NDN-CPP v0.4.0
 - repo-ng
- 실험 내용
 - 200MB, 700MB, 1GB, 2GB, 3GB 업로드
 - ndnputfile 시 repo-ng 데이터 전송 크기: 8K (ndn-cxx 기본 기본 패킷 크기)



Performance Evaluation

- repo-ng 및 NDNFS insert 수집
- 일반적으로 repo-ng 는 NDNFS 보다 성능이 떨어짐
- 1GB 파일 insert 의 경우 repo-ng 는 NDNFS 보다 많은 메모리, CPU 점유
- 메모리 소비량은 repo-ng 가 contents, name, signature 같은 모든 데이터를 읽기가 끝날 때 까지 메모리에 저장하기 때문
- NDNFS 메모리 소비가 적은 이유는 각 주기마다 파일 시스템에 데이터를 쓰는 크기 때문
 - 4KB 로 고정

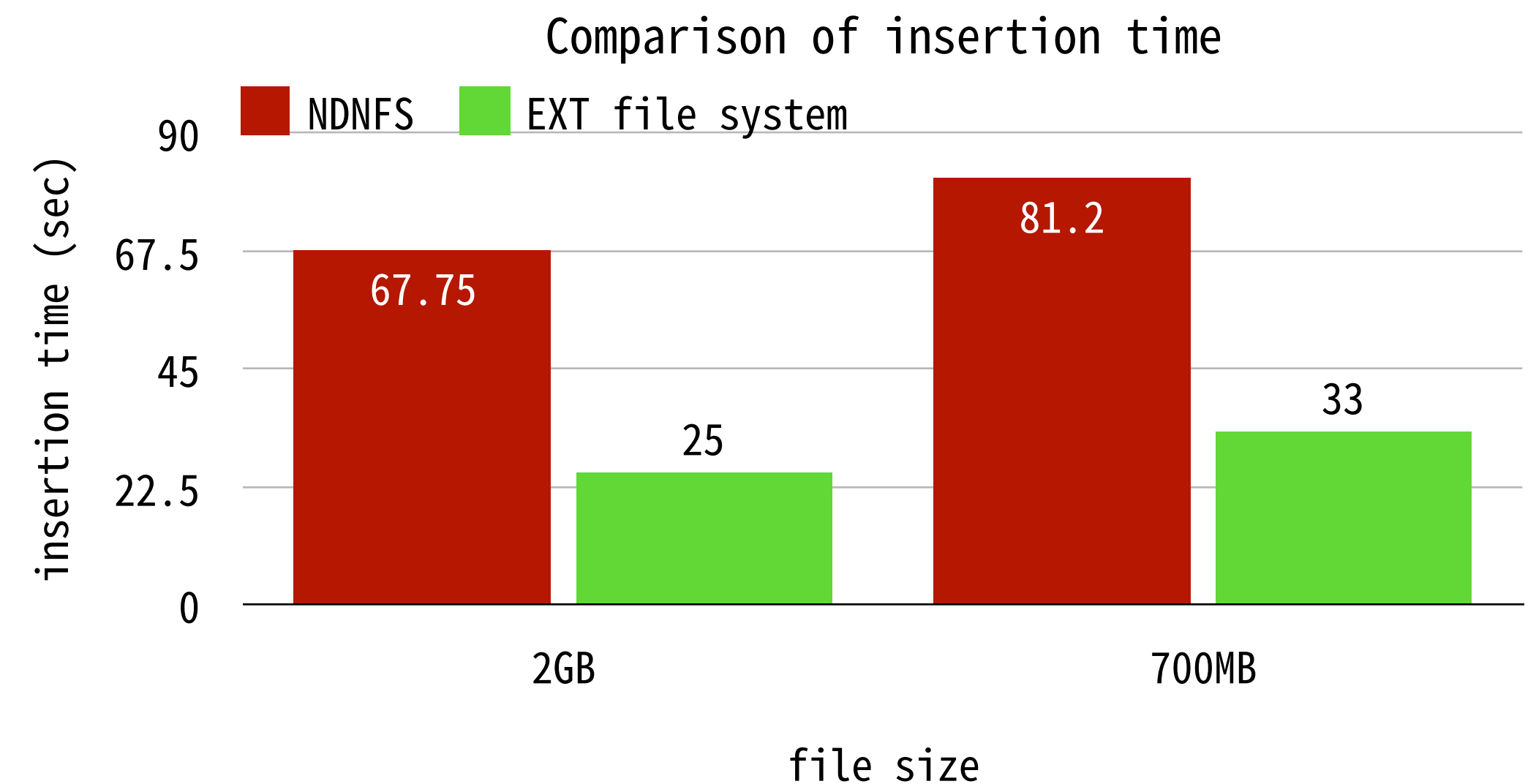
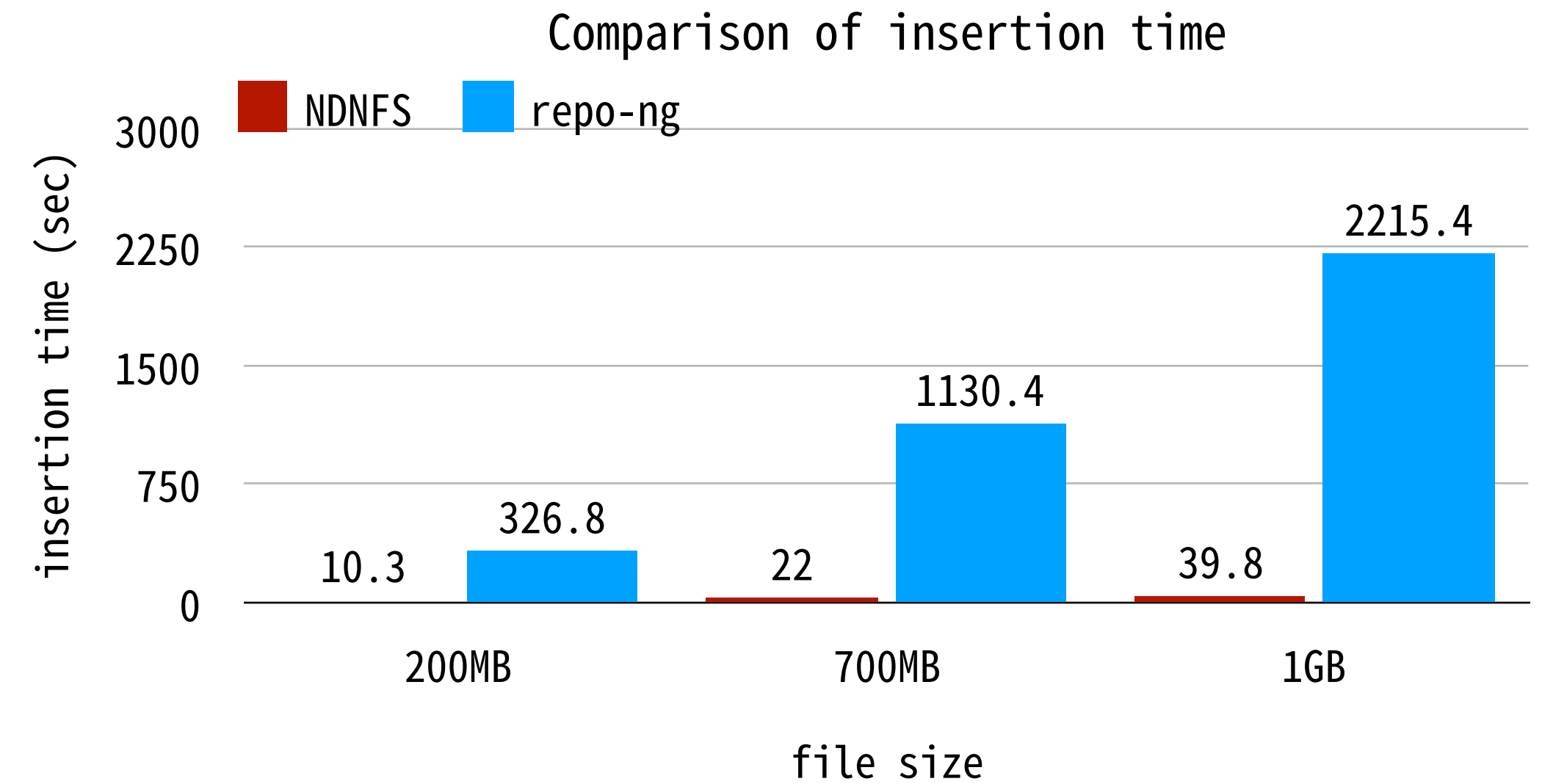
metrics	200MB		700MB		1GB	
	NDNFS-port	repo-ng	NDNFS-port	repo-ng	NDNFS-port	repo-ng
CPU	60%	100%	59.3%	100%	63%	100%
memory	0.1%	16.3%	0.1%	55.3%	0.1%	77%

[INSERTION performance comparison]



Performance Evaluation

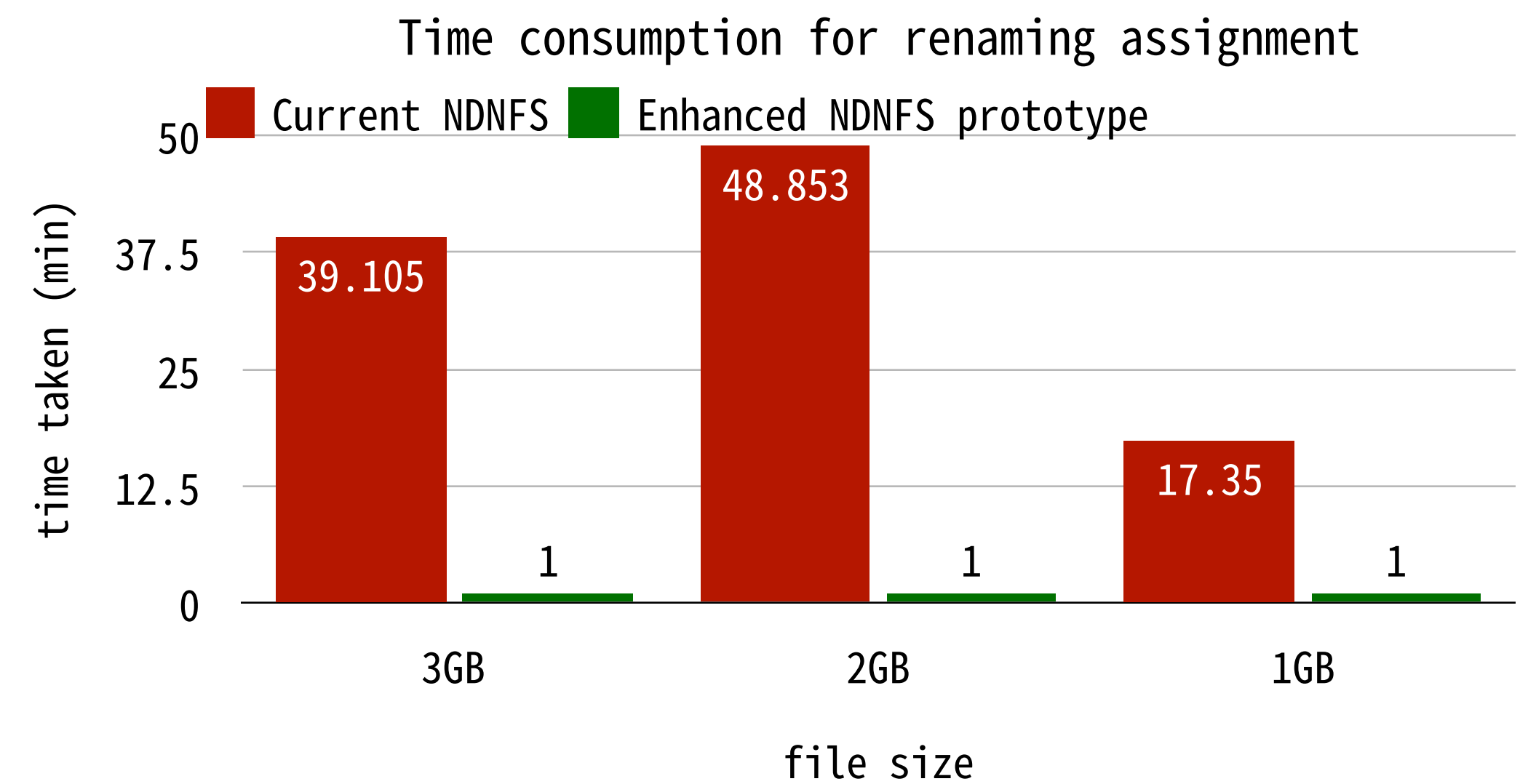
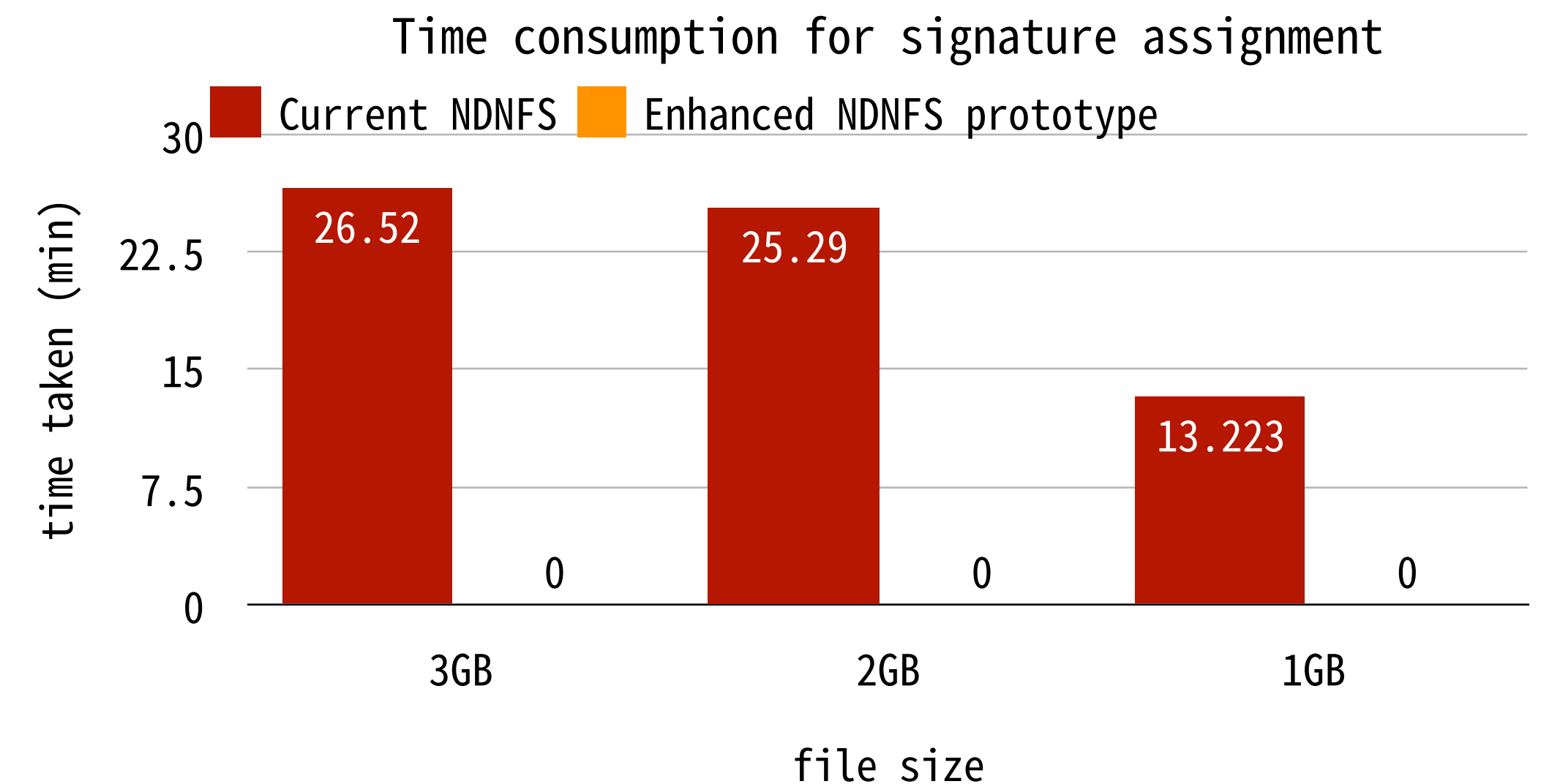
- comparison of insertion time
 - repo-ng
 - 스왑 파일에서 시스템 메모리 사용
 - 14GB RAM과 가상 메모리 모두 사용하여 시스템 메모리 공간 부족 발생
 - NDN 방식의 프로세스 사용으로 인한 시간 소비 발생
 - 1 Interest : 1 Data
 - NDNFS
 - repo-ng 보다 빠르지만 Linux ext4 에 비해 느림
 - 각 디스크 작업 주기마다 SQLite 작업 발생





Performance Evaluation

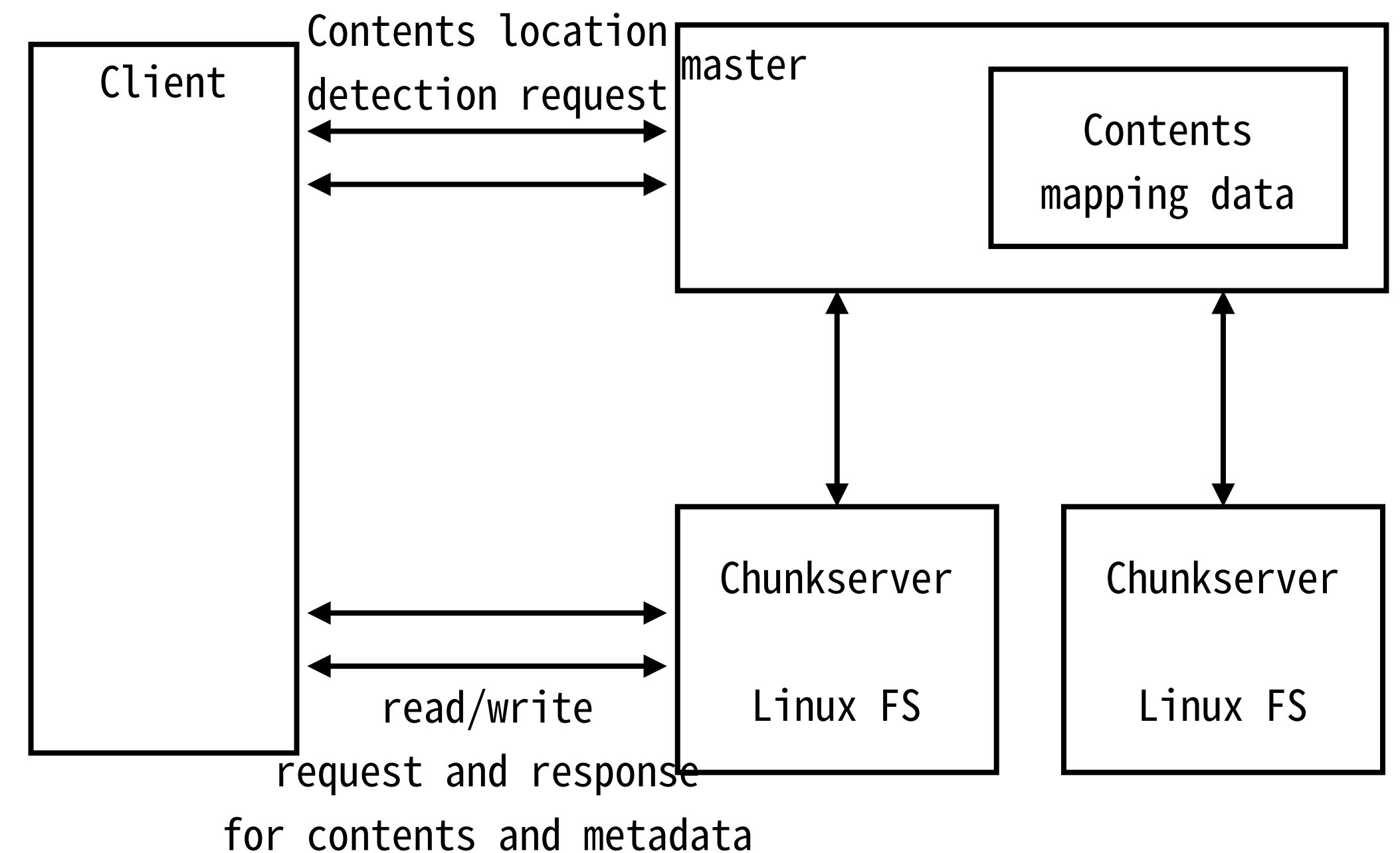
- comparison of signature time
 - SQLite
 - signature write 수행으로 인한 성능 저하
 - SQLite 작업 중에 다른 파일 작업 성능 저하
 - Enhanced NDNFS prototype
 - SQLite 작업 중에 다른 파일 작업 성능 저하 (동일 문제)
 - signature 작업이 없어 I/O 집약적 환경에서 성능 향상
- comparison of renaming time
 - SQLite
 - 모든 signature 및 metadata name 교체로 인한 성능 저하
 - Enhanced NDNFS prototype
 - SQLite 에 데이터 및 signature write 동작이 없음





Consideration on its Evolution to DFS

- 분산 파일 시스템은 client/server 모델, 데이터베이스를 사용한 metadata 유지
 - 데이터베이스 사용의 현실적 한계
 - I/O 집약적 시스템에서 발생하는 비용
 - 안정성/처리량을 고려하면 유료 DBMS 선택
 - 분산 파일 시스템 metadata 적용이 반드시 필요함
- 디자인
 - 중앙 metadata 서버
 - 기존 client/server 모델, 데이터베이스로 운영
 - 전체 내용의 위치를 매핑하기 위한 최소 데이터 구성
 - content 파일
 - 파일에서 파생된 모든 metadata 기록





NDNFS port git

- <https://github.com/remap/ndnfs-port>
 - master branch
 - server: NDN 을 사용하여 원격에서 NDNFS read 액세스 지원 (NFD 활용)
 - client: NDN-JS 라이브러리의 NDN-JS firefox 애드온 실행
 - NEW features
 - network-ready 패킷 대신 SQLite3 DB signature 만 저장
 - 새로운 metadata 브랜치에 mime_type 추가
 - NDN-JS firefox 애드온 및 ndn-cxx 버전 업데이트
 - In-File_Metadata branch
 - 논문 내용 구현

Port of ndnfs with CCL.

143 commits

6 branches

0 releases

4 contributors

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

jefft0

README: For ndnpeek, added the CanBePrefix and Verbose options.

Latest commit 82cf63e on 7 Jun

fs

misc: Fix bugs.

9 months ago

server

misc: Fix bugs.

9 months ago

test

misc: Fix bugs.

9 months ago

waf-tools

waf: Update boost.py.

5 months ago

.gitignore

server: add logging function

5 years ago

INSTALL.md

install: add note on boost lib path

3 years ago

README.md

README: For ndnpeek, added the CanBePrefix and Verbose options.

5 months ago

waf

build: update waf version for python3

5 years ago

wscript

wscript: no longer build out-of-date tests; change boost depency for ...

4 years ago

Branch: In-File_Metadata...

New pull request

Create new file

Upload files

Find file

Clone or download

This branch is 1 commit ahead, 11 commits behind master.

Pull request

Compare

neogeoss

In-file metadata

Latest commit 059cd69 on 11 Apr 2016

fs

In-file metadata

4 years ago

server

In-file metadata

4 years ago

test

wscript: no longer build out-of-date tests; change boost depency for ...

4 years ago

waf-tools

Initial commit: server module ported

5 years ago

.gitignore

server: add logging function

5 years ago

INSTALL.md

install.md: udpatate install instructions

4 years ago

README.md

ndnfs: suggest the merged firefox addon instead

4 years ago

waf

build: update waf version for python3

5 years ago

wscript

In-file metadata

4 years ago

THANKS

감사합니다