



ADA: An Application-Conscious Data Acquirer for Visual Molecular Dynamics

Tongliang Deng¹, **Hanpei Wu**², Yanliang Zou²,
Si Chen³, Tao Xie⁴, Shu Yin^{2,5,6,7}

¹SenseTime Research; ²ShanghaiTech University; ³West Chester University; ⁴San Diego State University;

⁵Shanghai Engineering Research Center of Intelligent Vision & Imaging;

⁶CAS, Shanghai Institute of Microsystem & Information Technology;

⁷State Key Lab of High Performance Computing



LABORATORY OF
I/O SYSTEMS
& DATA SCIENCE
并行与分布式I/O系统实验室



Outline

- **Background**
- Motivation
- Design
- Evaluation
- Conclusion

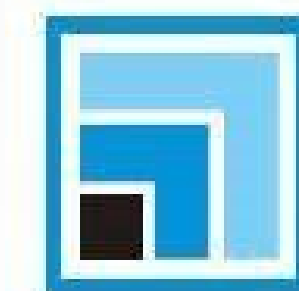


Background: Molecular Dynamics(MD)

- MD is a computer simulation technique that simulate motion of molecular systems, widely used in biology, physics, chemistry and materials domains.
- For serving different needs, there are many MD applications, including NAMD, GROMACS, BIOVIA, MATERIALS STUDIO...









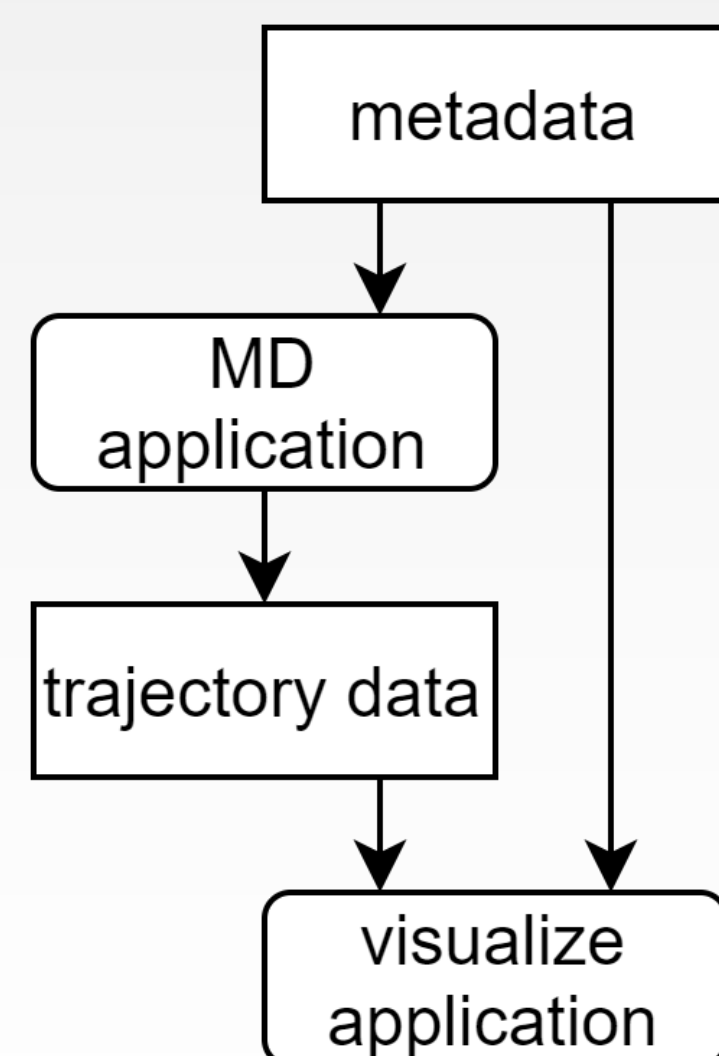
Background: Molecular Dynamics(MD)

- MD applications accept initial state and environment from input **metadata**, calculate position and speed for next moment using force field in last moment, write the simulation result of each moment into a **trajectory data**.
- Trajectory data records space coordinates of each particle in each frame. Large trajectory data often be compressed for storage and transmission
- Scientist use MD result to observe changes in the microsystem, but they cannot directly observe or analysis from coordinates numbers which may even be compressed.



Background: Visualization for MD Result

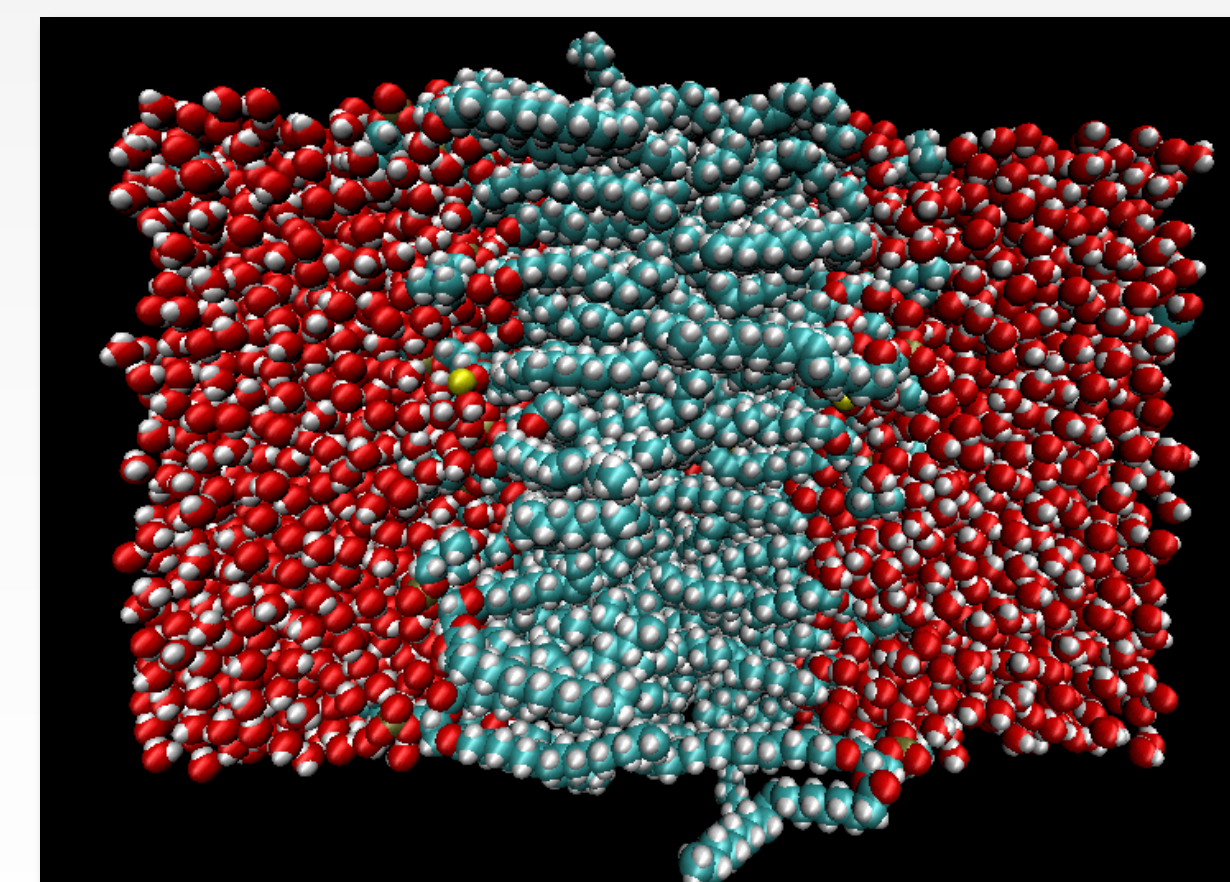
- Visualization application for MD decompress and render coordinates to 3D animations, help scientists observe the trend or the key time point of the changes.
- Workflow, Visualization App and animation example:



PyMOL by Schrödinger

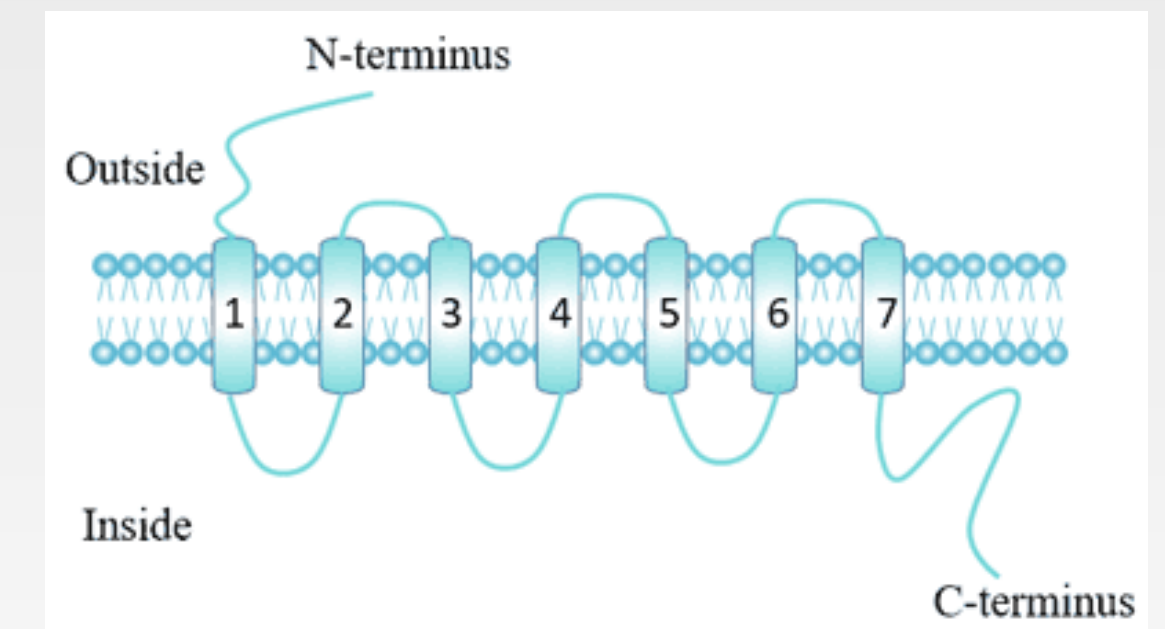
VMD
Visual Molecular Dynamics

UCSF CHIMERA
an Extensible Molecular Modeling System



Background: GPCR Research

- G-protein coupled receptors(**GPCR**) is a group of protein on cell membrane. GPCR research reveal the drug target and physiological mechanism.
- G-protein coupled receptors(GPCR) is the of about 40% modern medicine.
- So GPCR become a frontier focus of biological research. ShanghaiTech iHuman Institute published many important breakthroughs about GPCR on **Science**, **Cell** and **Nature**.



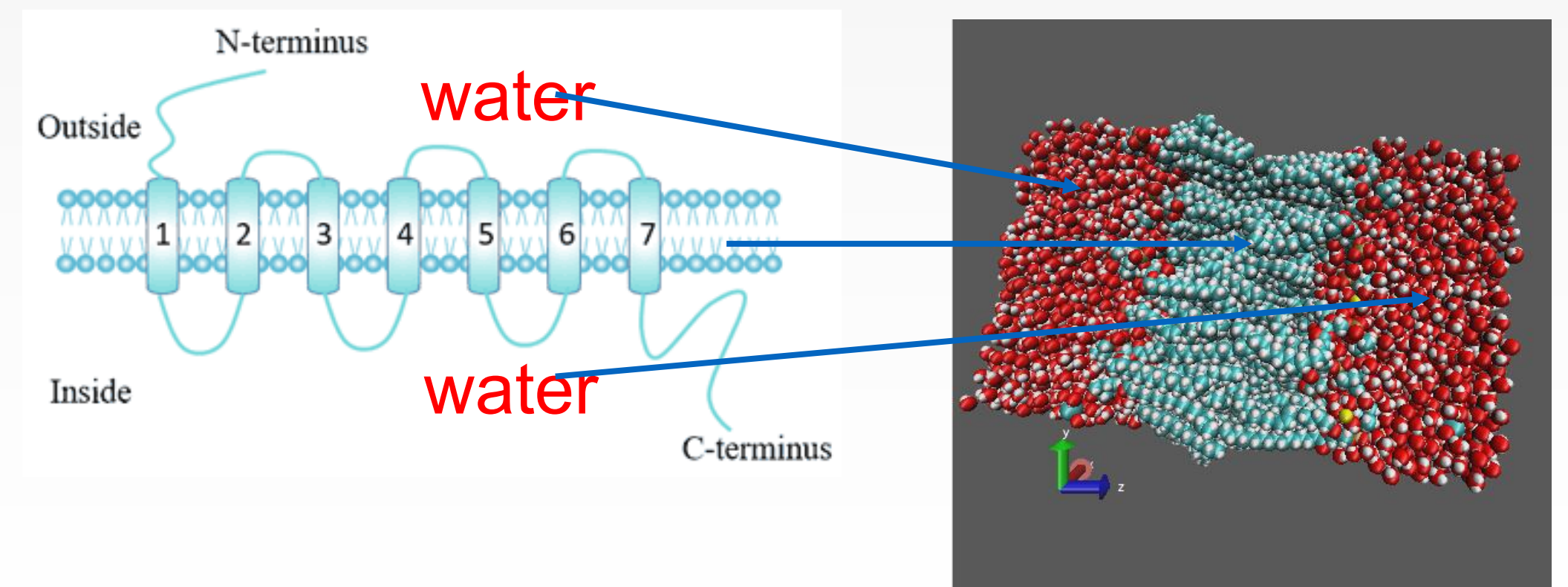
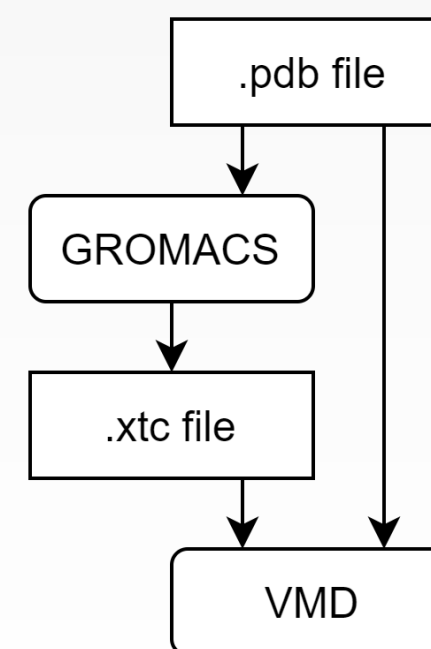
Outline

- Background
- **Motivation**
- Design
- Evaluation
- Conclusion



GPCR MD

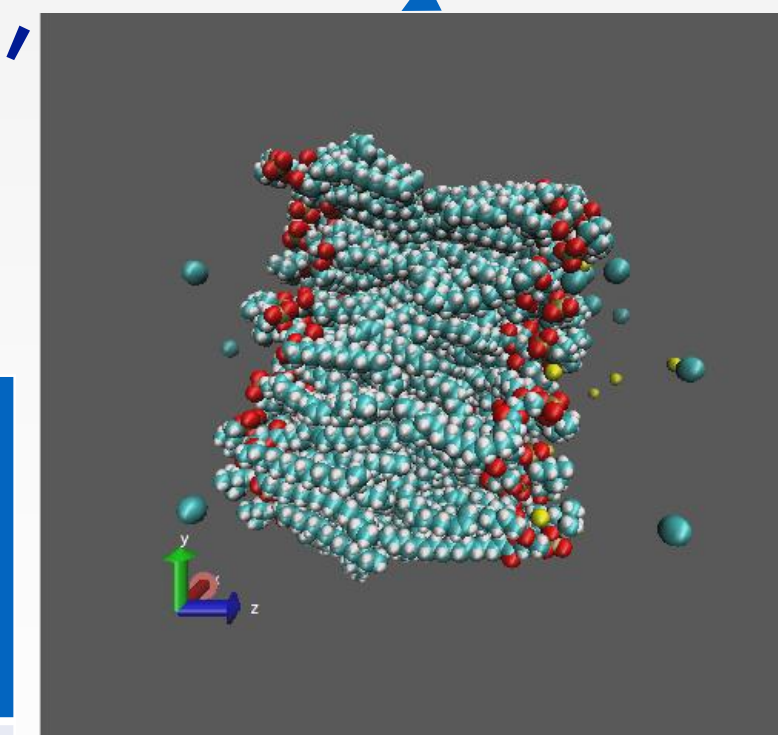
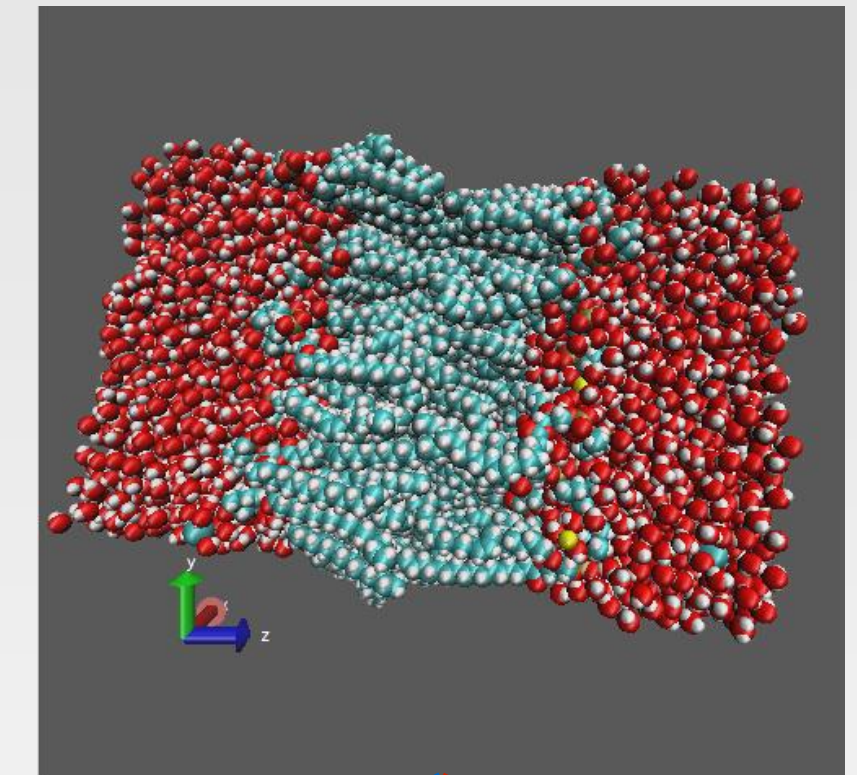
- To research principles of drug action on GPCR, researchers use GROMACS to simulate, use VMD to render.
- GPCR is on cell membrane, both sides of the membrane are water. MD need to reappear GPCR environment for simulation, so metadata and trajectory both contain **water**.
- VMD render received trajectory, so VMD also load and render **water**.



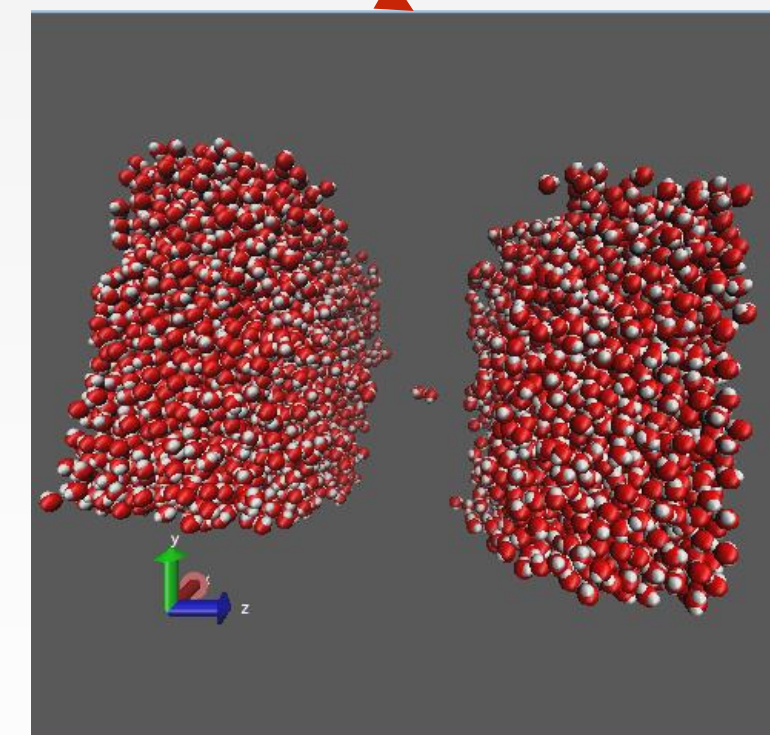
GPCR MD

- Water trajectory is not the focus of observe.
- We define trajectory data into 2 types:
 - Water
 - Non-water (protein, ligand, phospholipid, ...)
- We only need to visualize non-water data, non-water data only accounts for less than 50%.
 - Load water data requests more memory.

Number of frames	Compressed file size (MB)		Non-water data fraction (%)
	Complete data	Non-water data	
626	100	44	44
1251	200	98	49
5006	800	348	43.5



Non-water

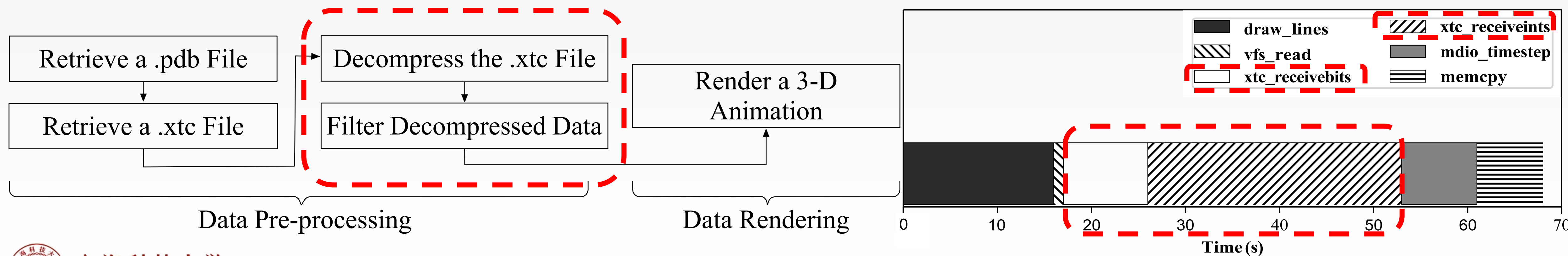


water



Motivation: VMD dataflow

- VMD need Data Pre-processing before Data Rendering.
- Time-consuming **redundant** effort over repeatedly study in Data Pre-processing. **Redundant** time accounted for over 50% CPU burst time.



Outline

- Background
- Motivation
- **Design**
- Evaluation
- Conclusion



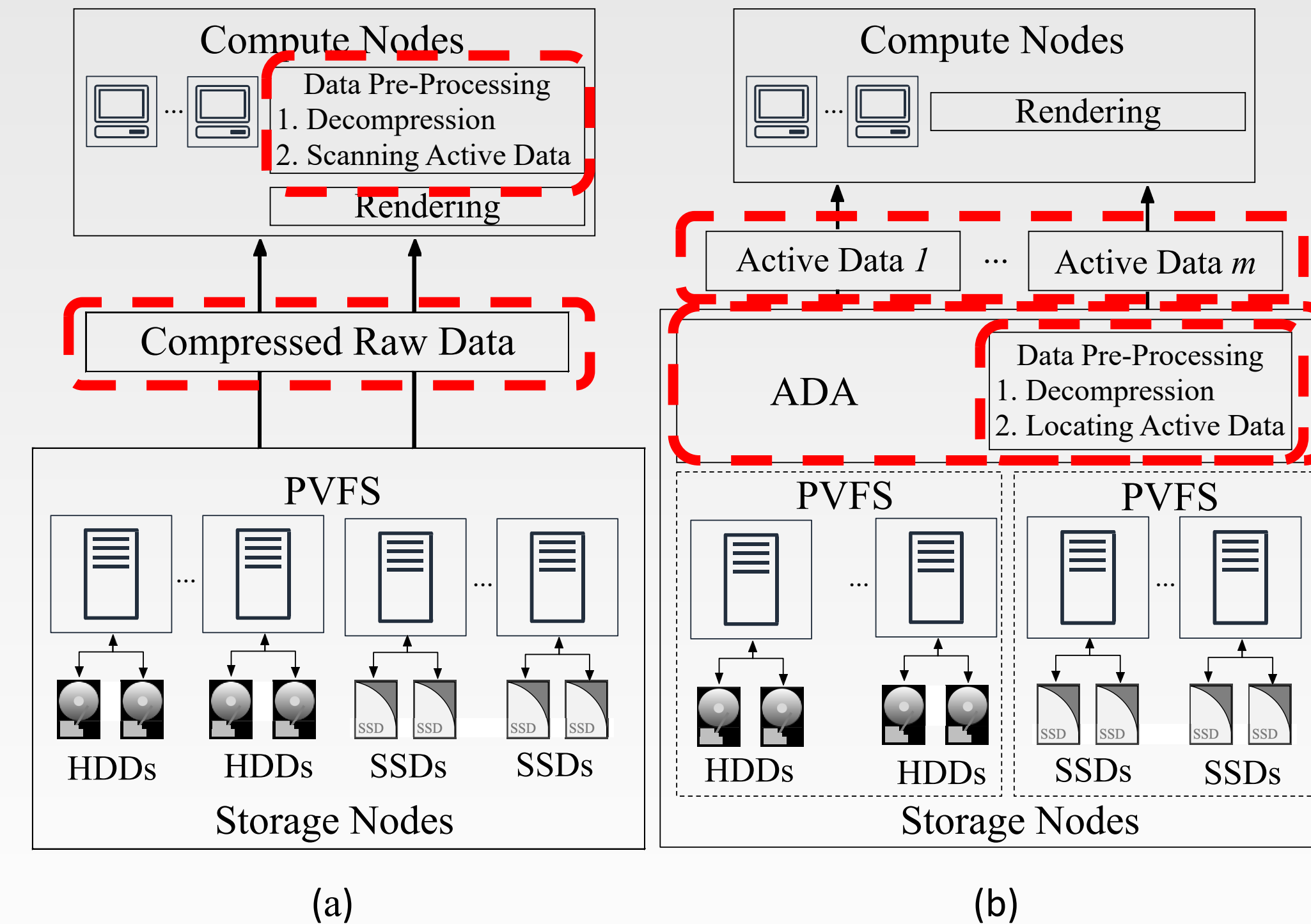
Design: Overview

- Design goal
- ADA Architecture
 - Data Pre-Processor
 - I/O Determinator
- ADA workflow
 - Data Writing
 - Data Retrieving

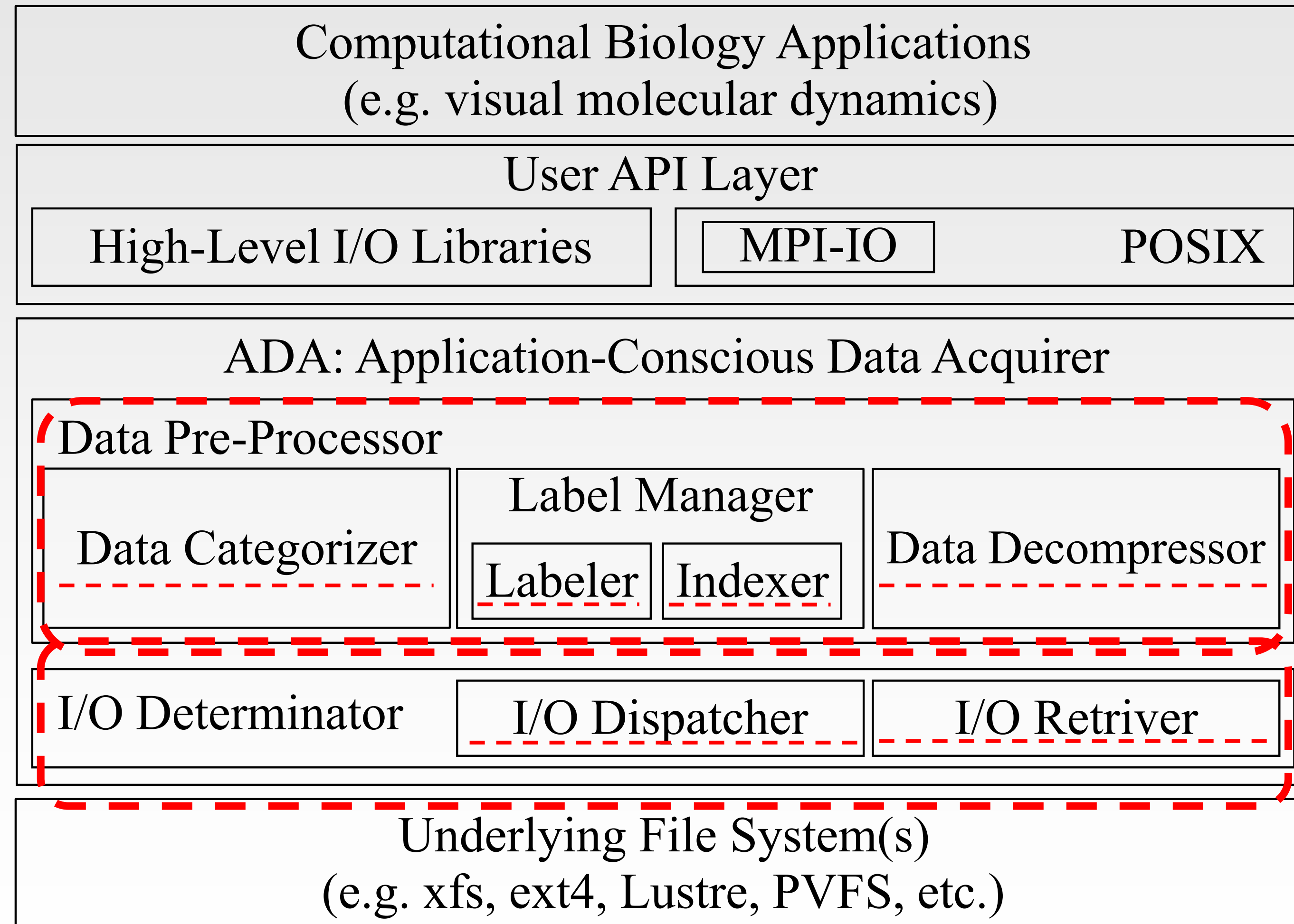


Design Goal

- ADA is a **middleware** between compute node and storage node
- ADA should perform Data pre-processing before data retrieving
 - **avoid redundant work**
- ADA only provide the **needed data**
 - domain scientists focus on



ADA Architecture

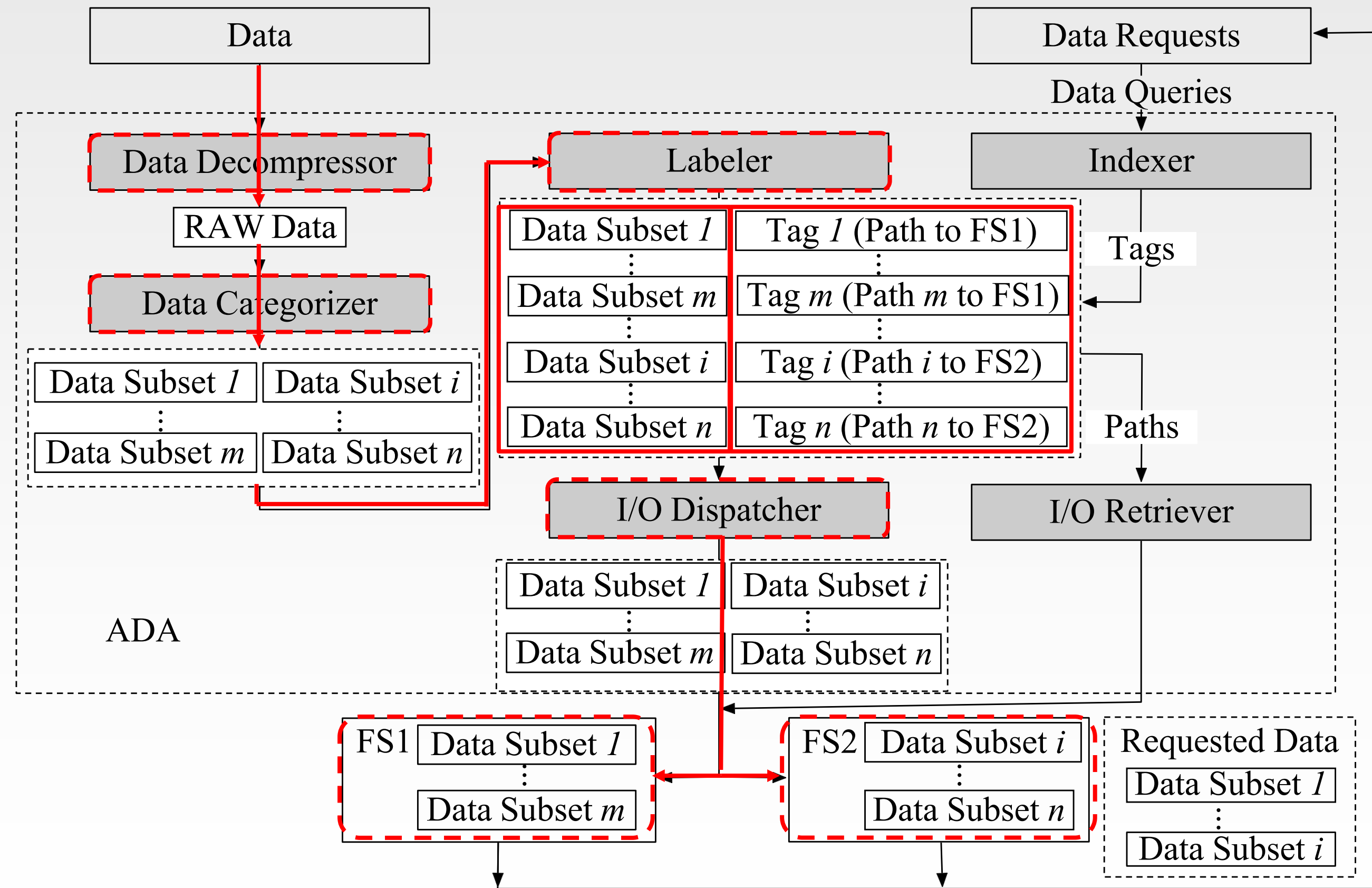


ADA Module Function

- Data Decompressor: **Decompress** data into raw data.
- Data Categorizer: **Categorize** data into subsets.
- Labeler: **Assign** a tag to a data subsets.
- Indexer: **Map** the tag with data subsets, get path from it.
- I/O Dispatcher: **Dispatch** data subset to underlying File System via tag.
- I/O Retriever: **Retrieve** data subset via tag.

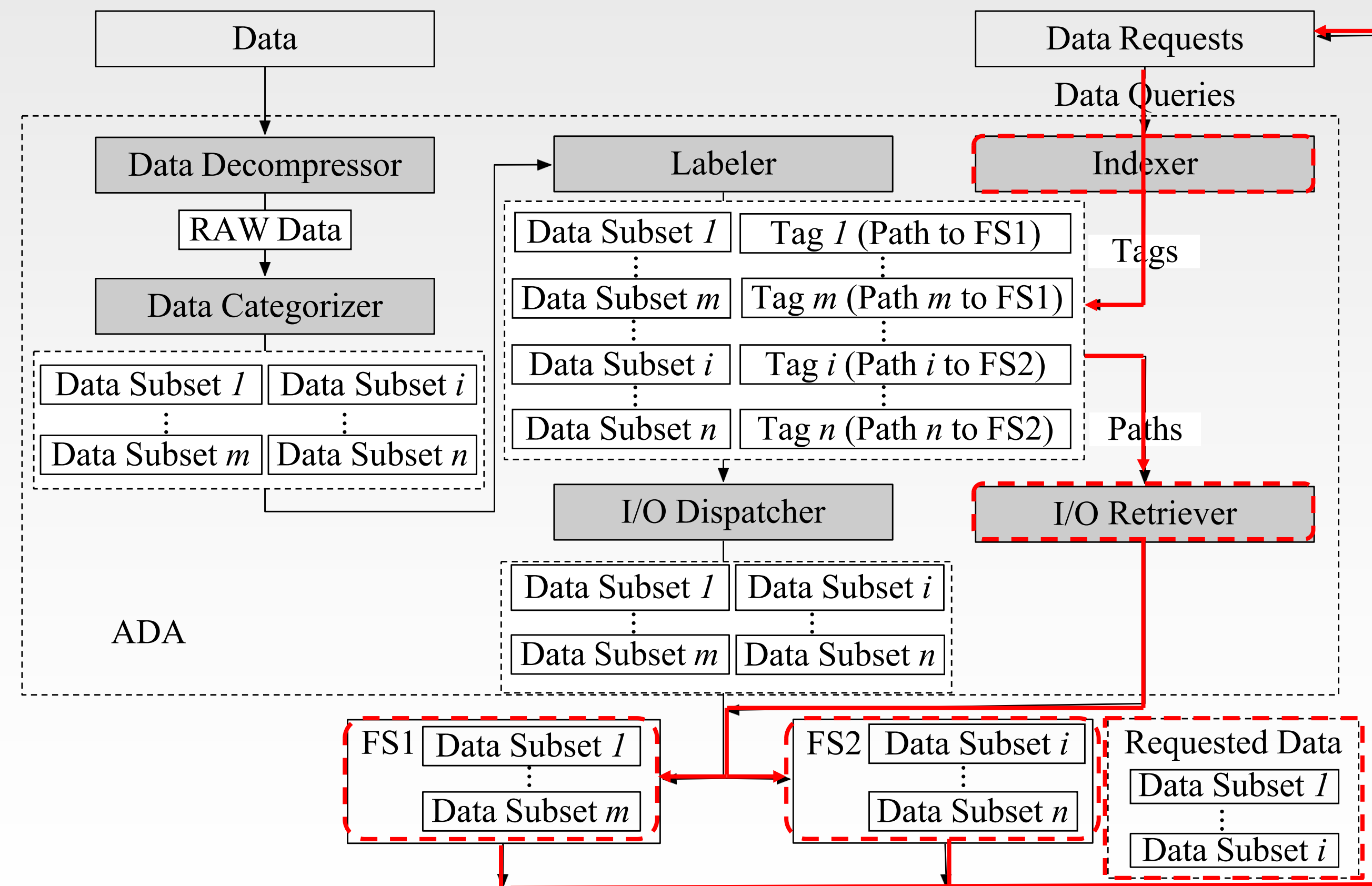


ADA Dataflow - Writing



ADA Dataflow - Retrieving

- Example commandline: `mol addfile /mnt/bar.xtc tag 1, ..., i`



Outline

- Background
- Motivation
- Design
- **Evaluation**
- Conclusion



Evaluation: Experimental Setup

- Hardware
 - Single-node server:
 - Intel Xeon CPU E5-2603 v4 @1.70GHz, 16GB DRAM
 - Two 256GB NVMe SSD
 - CentOS release 6.10 (Final) with ext4
 - 9-node hybrid PVFS cluster:
 - 3*compute node: 6-core Intel Xeon CPU E5-2603 v4 @1.70GHz, 16GB DRAM
 - 3*1TB HDD+3*256GB NVMe SSD
 - CentOS release 6.10 (Final) with PVFS2(OrangeFS 2.8.2)
 - Fat Node:
 - Intel Xeon CPU E7-4802 v3 @1.90GHz, 1007GB DRAM
 - 10*1TB WD HDDs(RAID 0)
 - The CentOS release 7.3 with 3.10 kernel with XFS



Evaluation: Experimental Setup

- Data set for single-node server and PVFS cluster
 - Real world GPCR data, collected from ShanghaiTech iHuman
 - Up to 5006 frames. Compressed file up to 800MB, which become 2612MB after decompress, 1108MB of them are decompressed non-water data.

Number of Frames	Loaded Data (MB)		
	PVFS (compressed)	ADA (all)	ADA (non-water)
626	100	327	139
1251	200	653	277
1877	300	980	416
2503	400	1306	555
3129	500	1632	693
3754	600	1959	832
4380	700	2285	970
5006	800	2612	1108



Evaluation: Experimental Setup

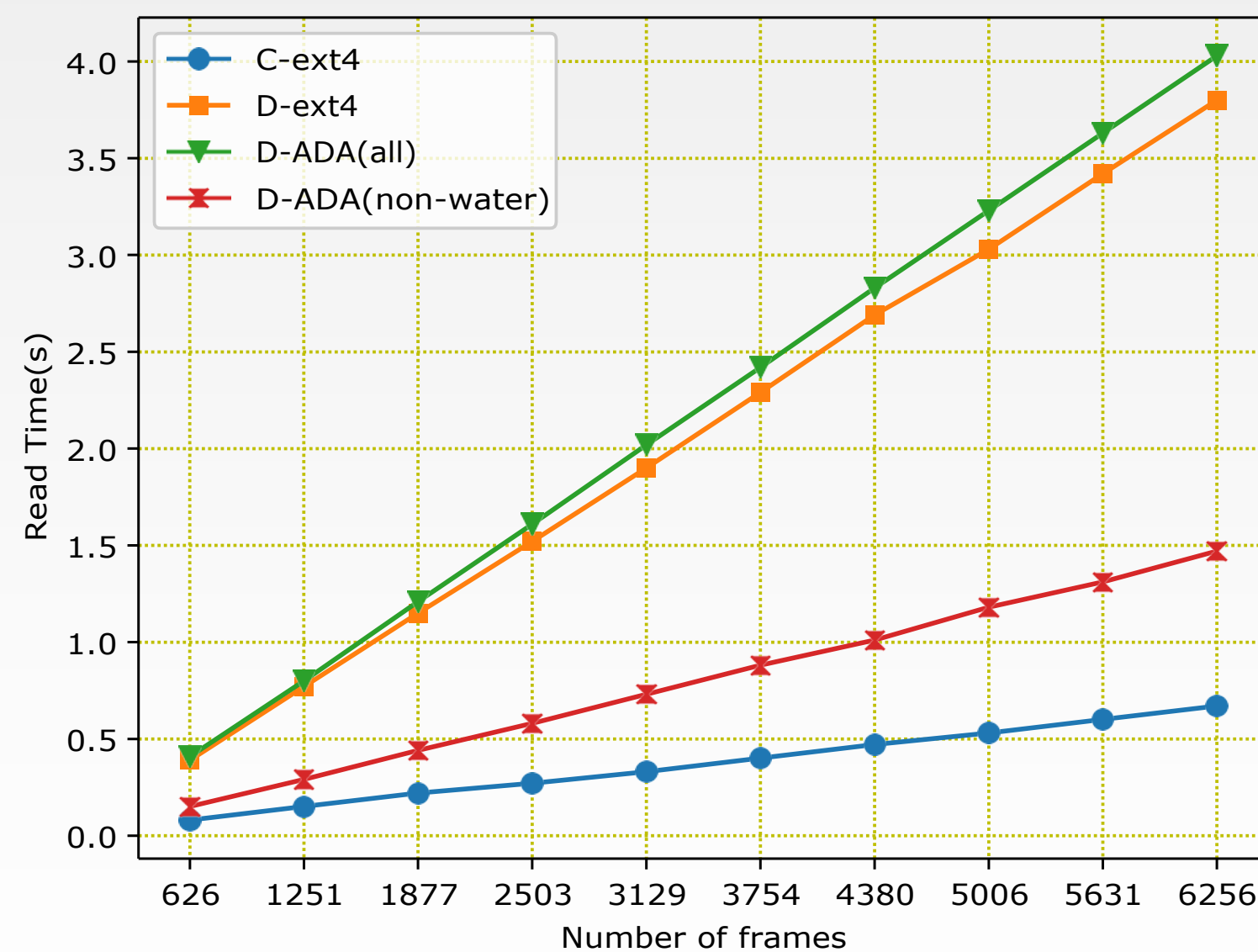
- Data set for Fat Node
 - Real world GPCR data, collected from ShanghaiTech iHuman
 - Much larger Data set
 - More frames, larger data size(10x than previous),
 - Up to 5,004,800 frames. Compressed file up to 800GB, which become 2612.8GB after decompress, 1108.8GB of them are decompressed non-water data.

Number of Frames	Loaded Data (GB)		
	XFS (compressed)	ADA (all)	ADA (non-water)
62,560	10	32.7	13.9
187,680	30	98	41.6
312,800	50	163.3	69.3
437,920	70	228.6	97
625,600	100	326.6	138.6
938,400	150	489.9	207.9
1,251,200	200	653.2	277.2
1,564,000	250	816.5	346.5
1,876,800	300	979.8	415.8
2,502,400	400	1306.4	554.4
3,440,800	550	1796.3	762.3
4,379,200	700	2286.2	970.2
5,004,800	800	2612.8	1108.8

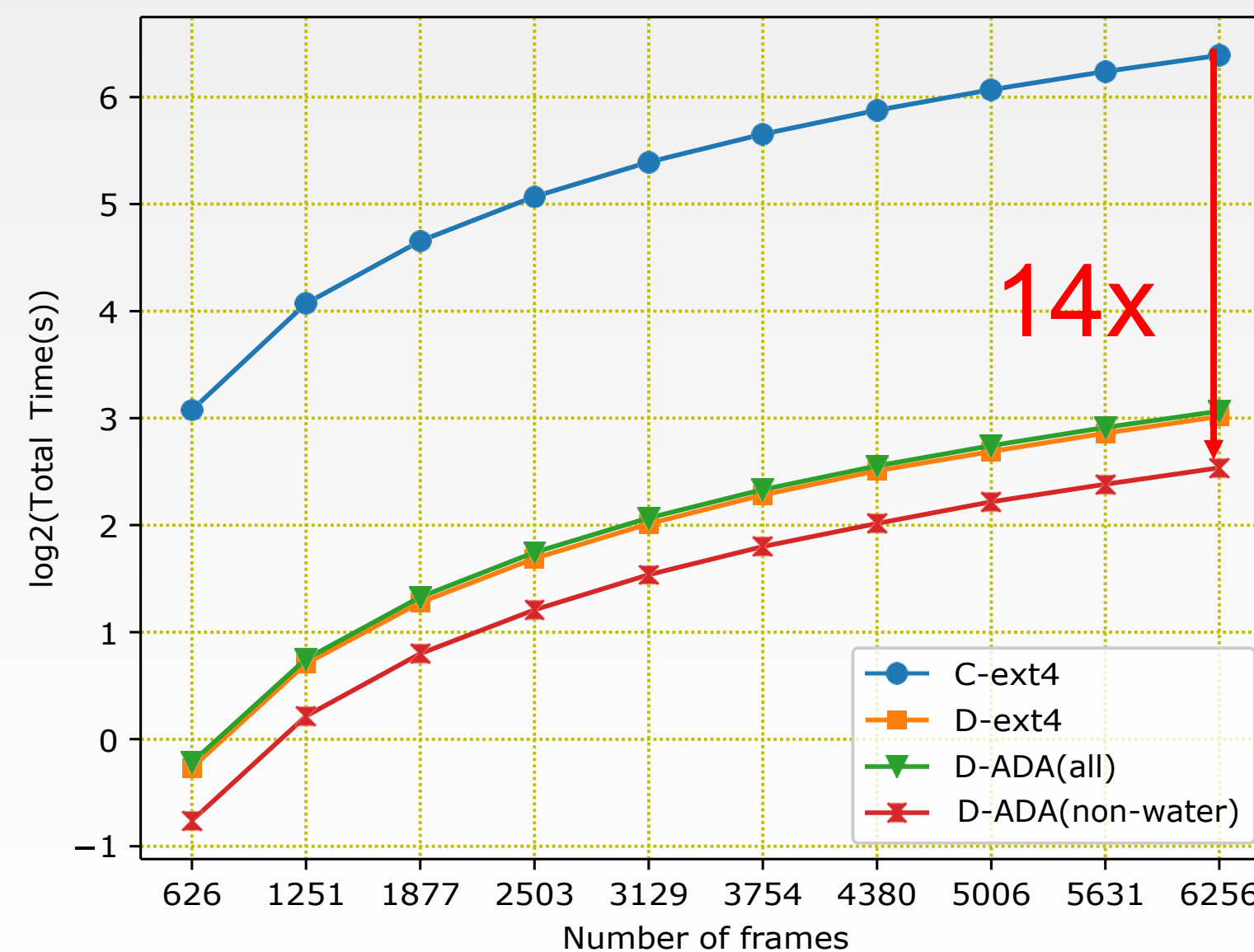


Evaluation: a Single-Node Server

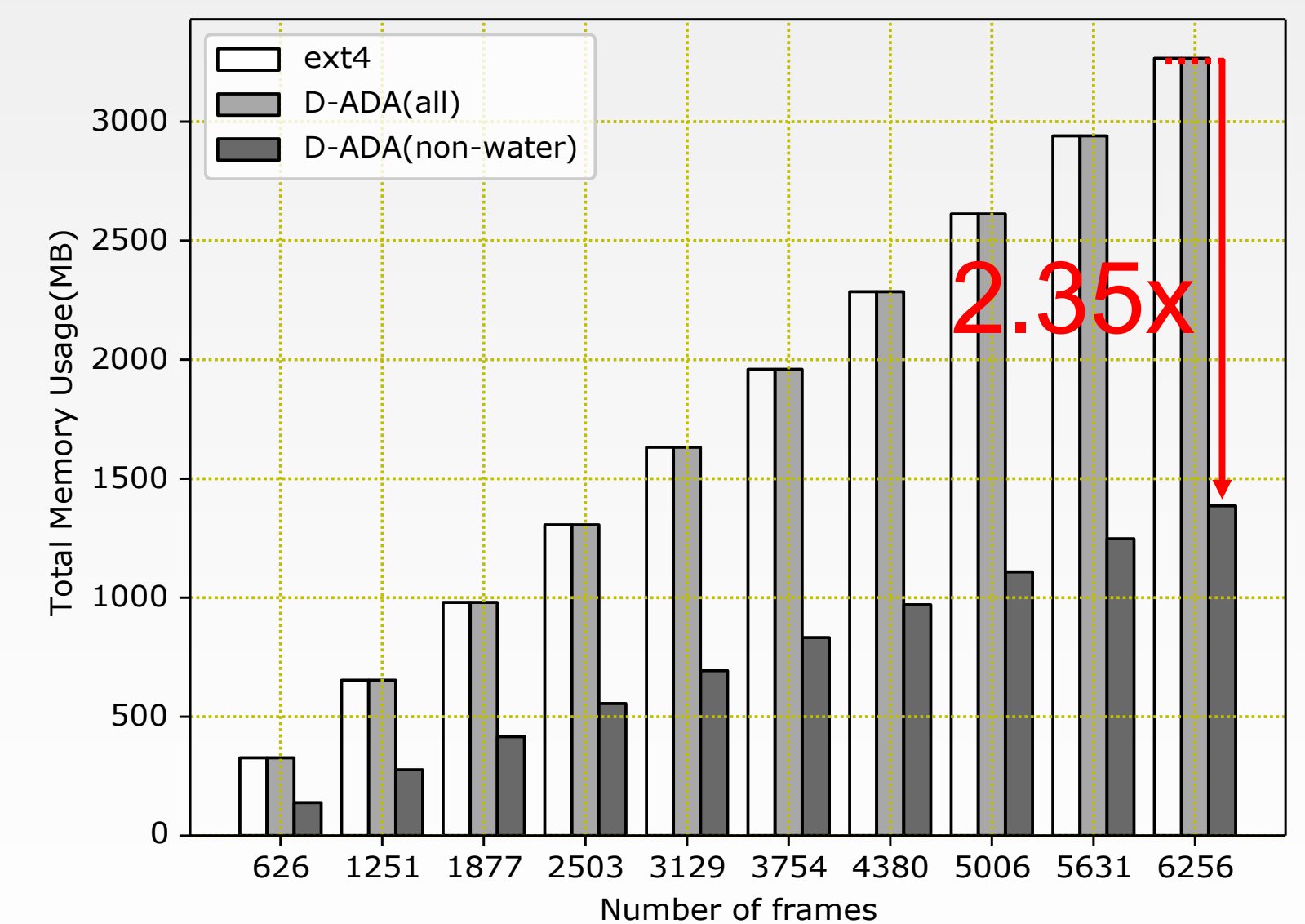
- Evaluation on a single-node SSD Server
 - 14x faster, save 2.35x memory



Read time



Total time(log2)

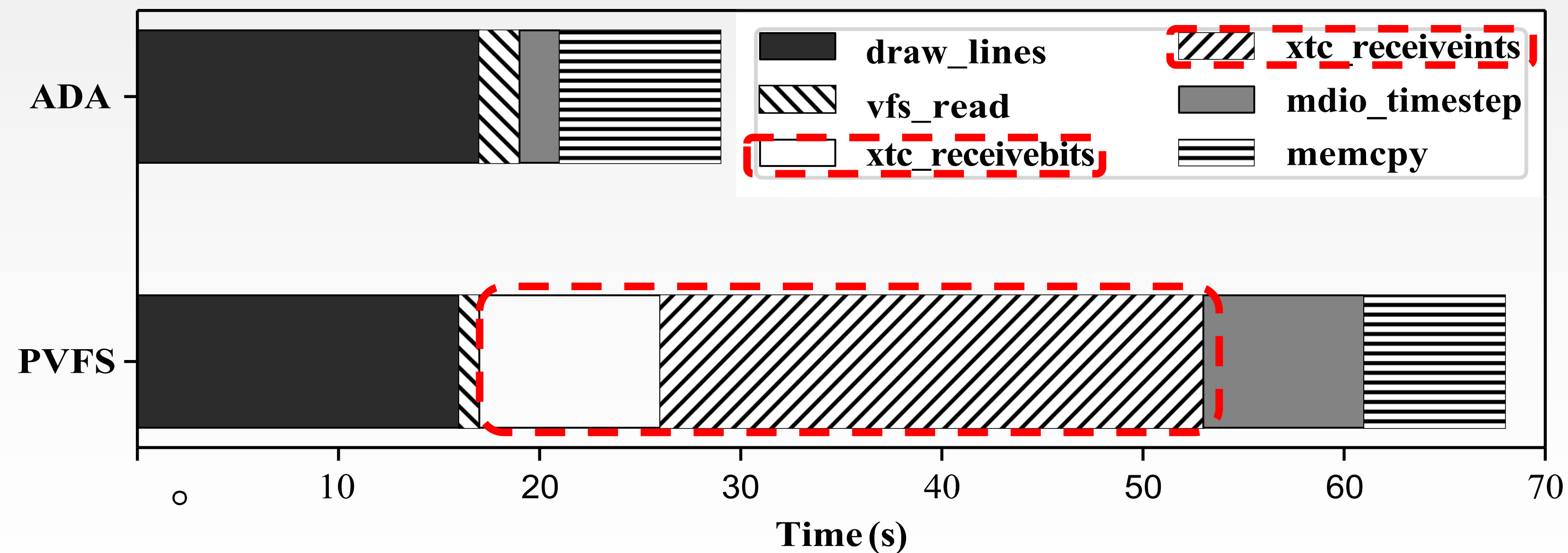


Total Memory Usage



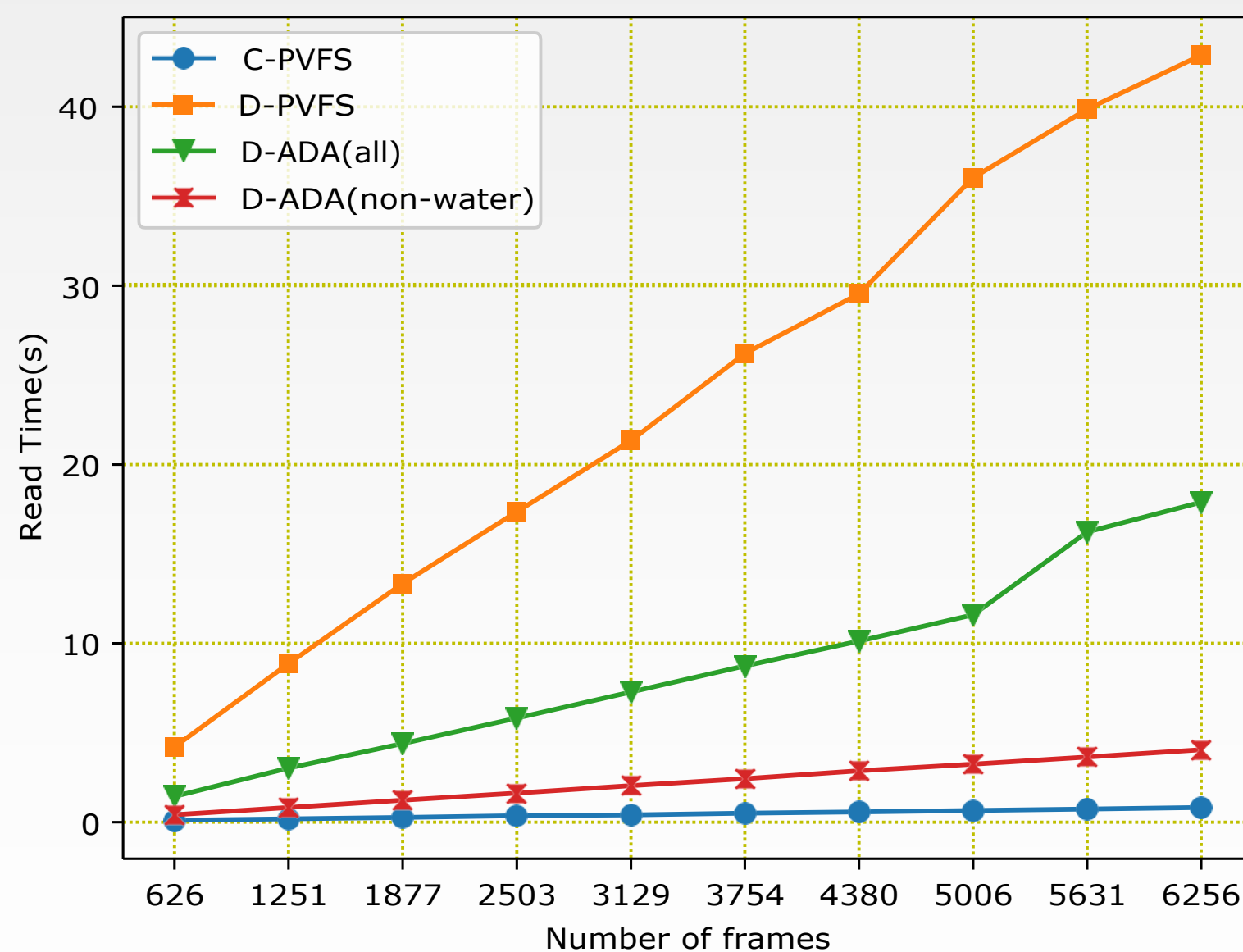
Evaluation: CPU Burst Time

- CPU burst time comparison.
- Decompression weight more than 50%, which can be eliminated in ADA.

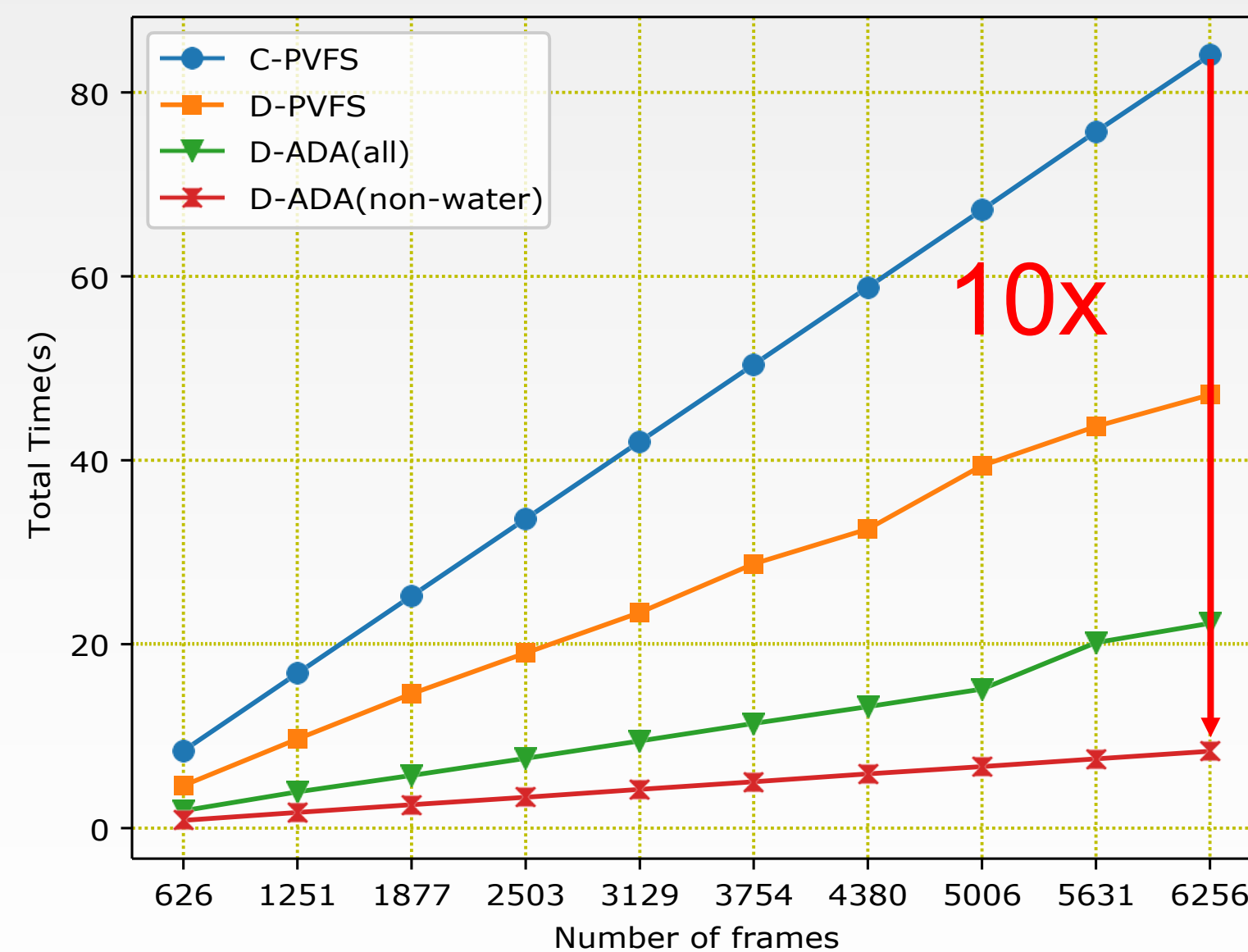


Evaluation: a 9-Node Hybrid PVFS Cluster

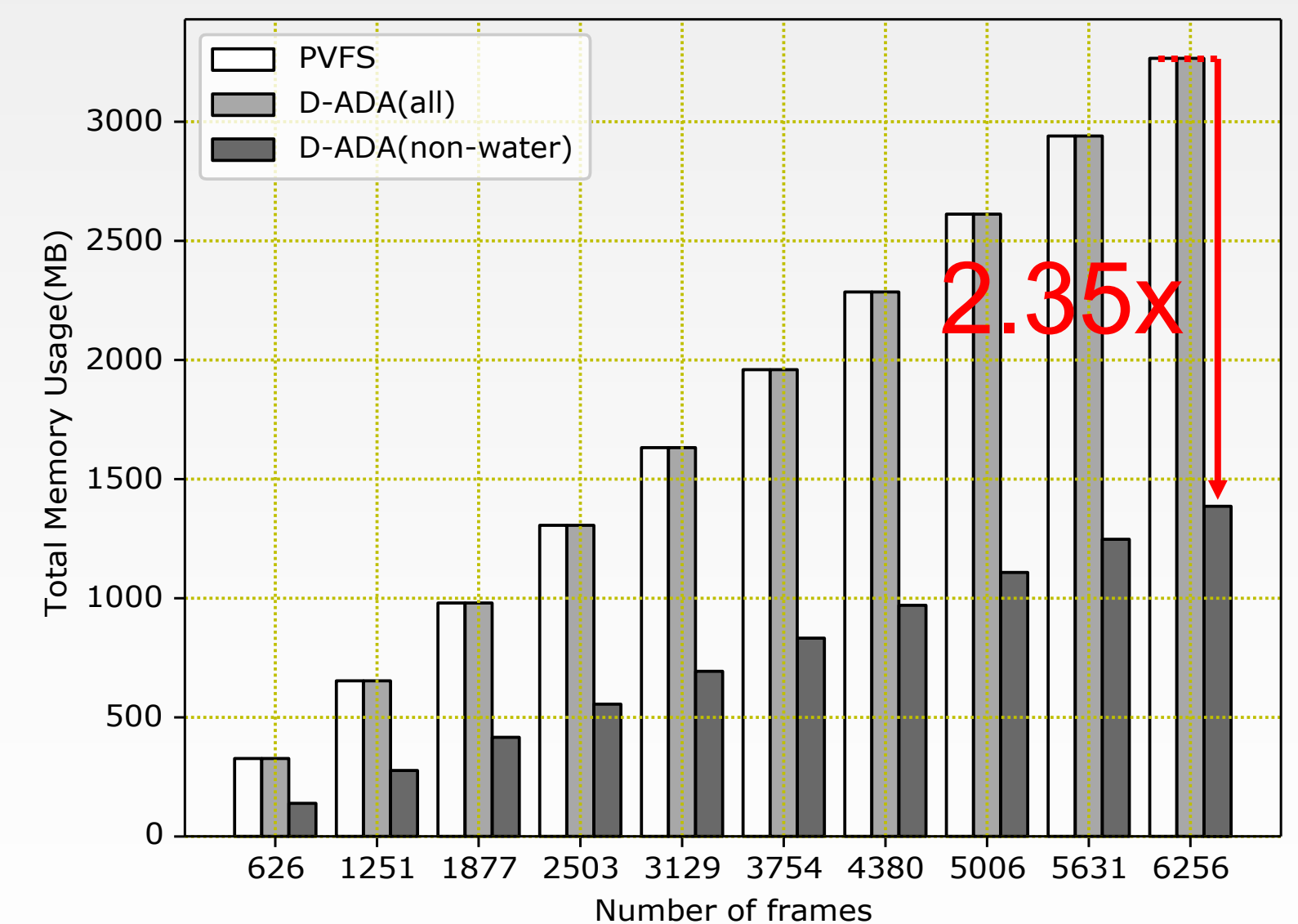
- Using the same data set as single-node server used.
- Evaluation on a 9-node hybrid PVFS cluster
 - 10x faster, save 2.35x memory



Read time



Total time

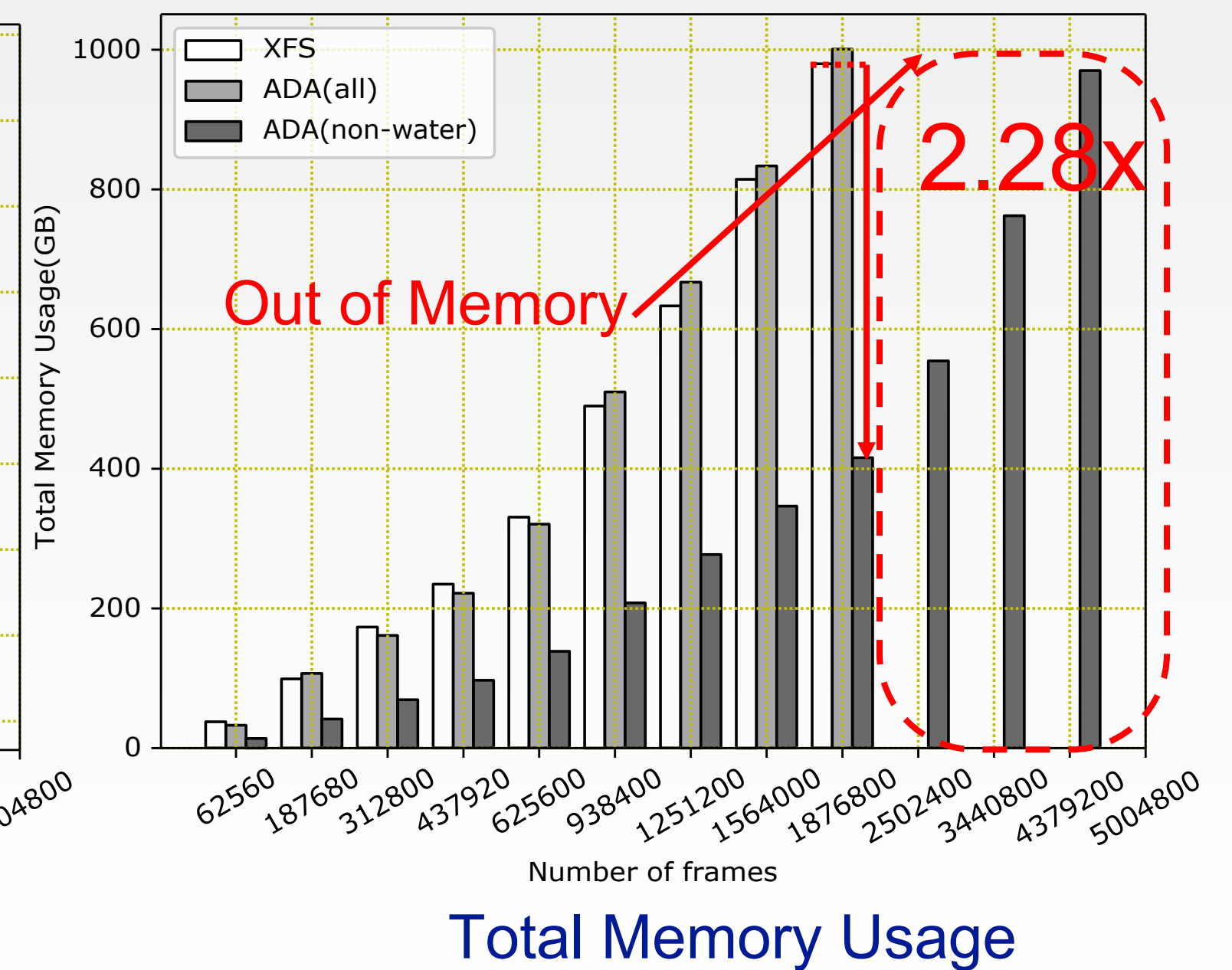
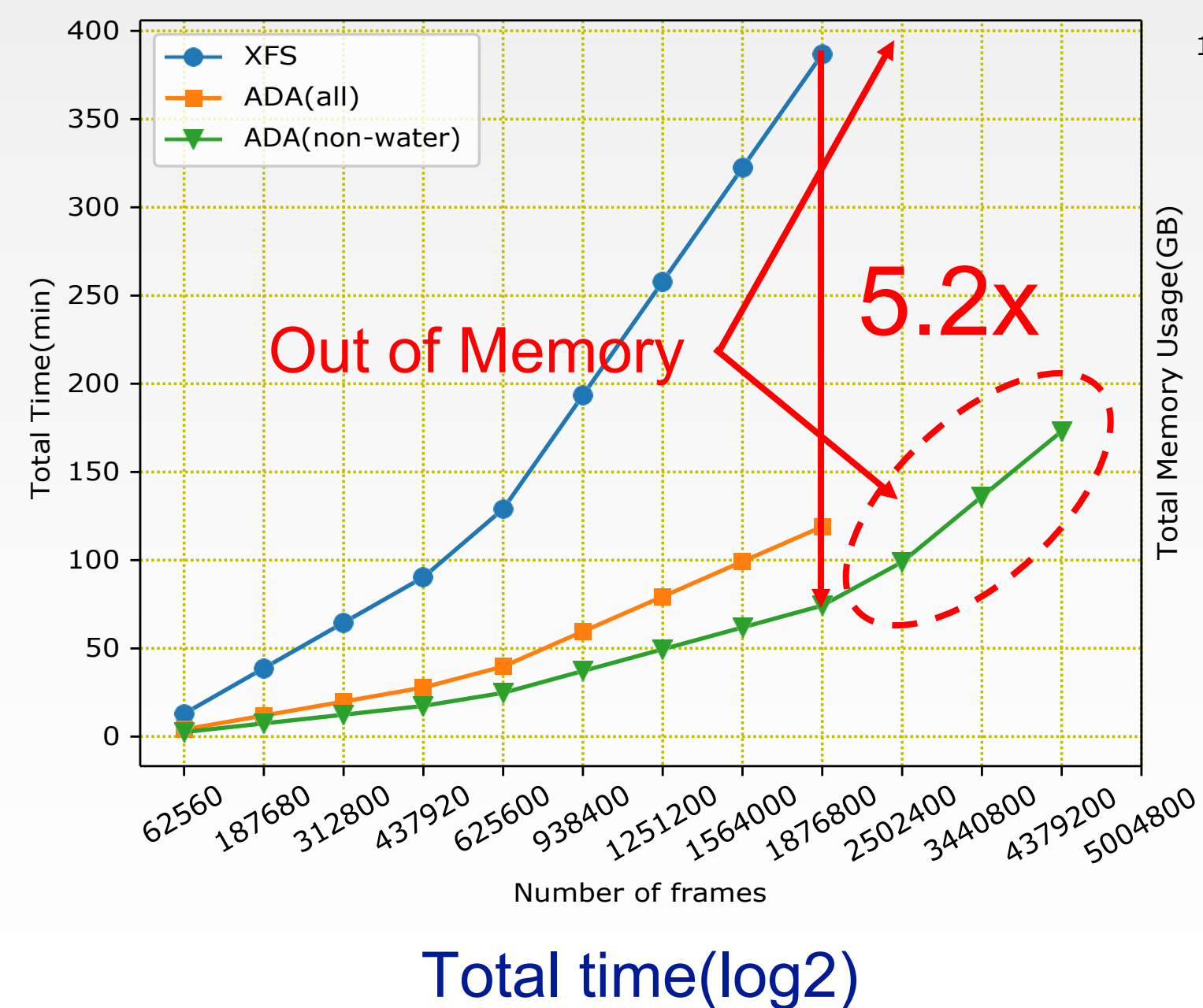
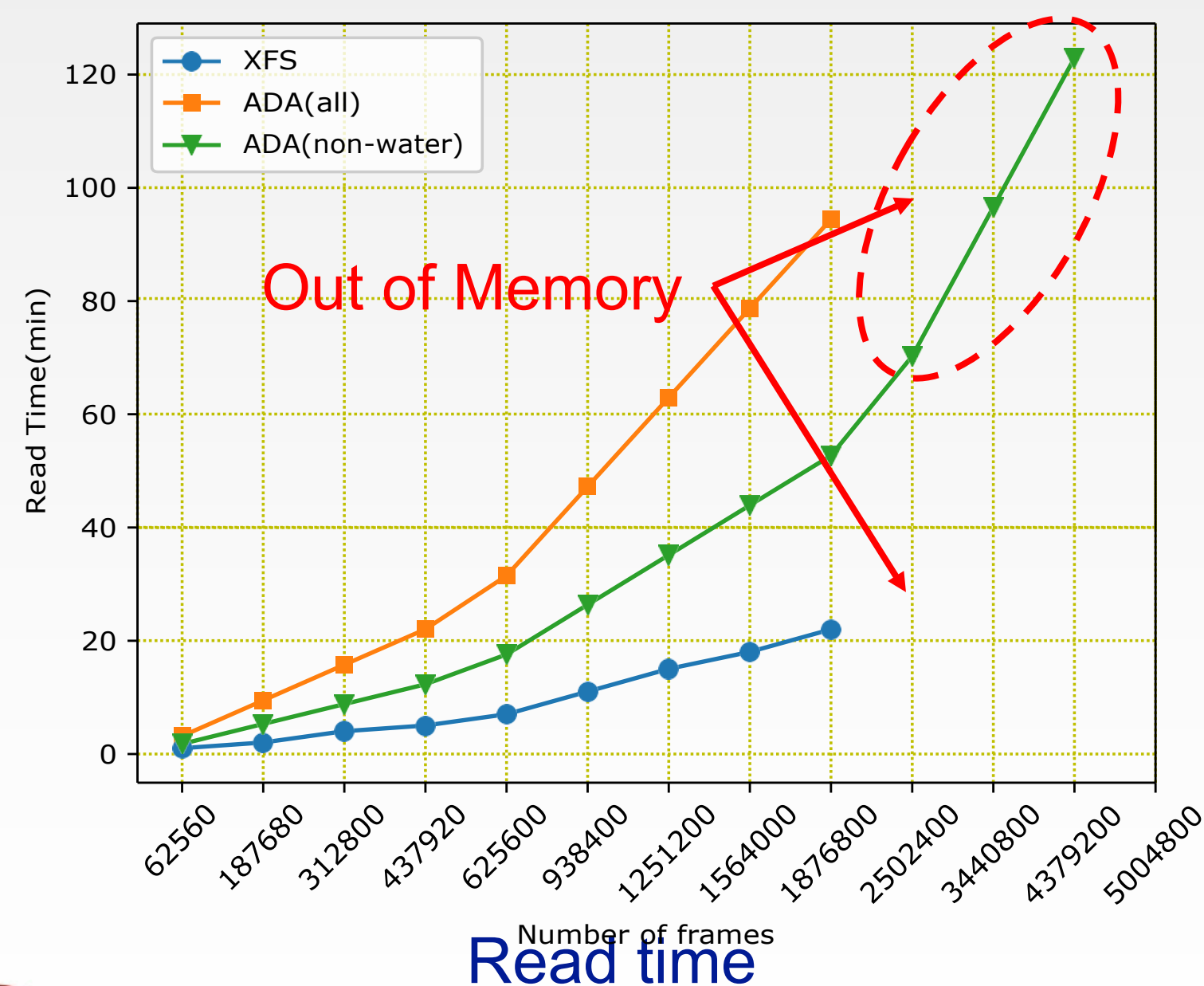


Total Memory Usage



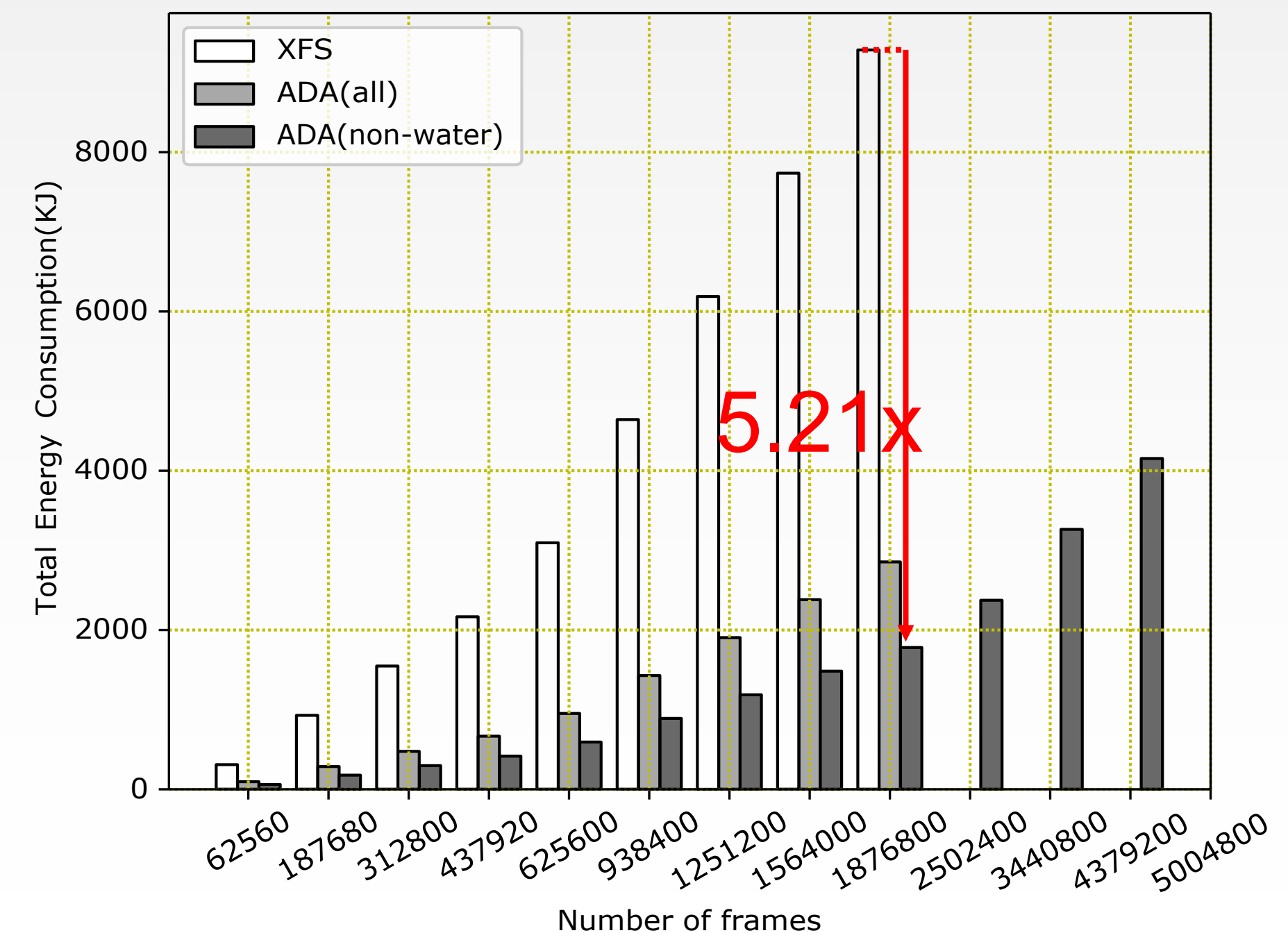
Evaluation: Fat Node

- Much larger data set to reach the memory bottleneck.
- Evaluation on Fat Node
 - 5.2x faster, save 2.28x memory, render 2.3x frames



Evaluation: Energy consumption

- Real world calculated by A PDU, test in Fat Node.
- ADA consume only 1780 KJ energy while ordinary VMD consume 9283KJ, saved up to **5.21x** energy(**80.8%**).
- Device limited energy consumption is 12378 KJ(XFS) and 4252 KJ(ADA), ADA saved **2.91x** energy.



Results Summary

- Evaluation on a Single-Node Server
 - 1.4x faster, save 2.35x memory
- Evaluation on a Small Cluster
 - 10x faster, save 2.35x memory
- Evaluation on a Fat Node
 - 5.2x faster, save 2.28x memory
 - Save 2.91x-5.21x energy
 - Render 2.3x more frames



Outline

- Background
- Motivation
- Design
- Evaluation
- **Conclusion**



Conclusion

- Identify a data processing challenge faced by many molecular dynamics applications for visualization and analysis.
- A lightweight file system called ADA.
 - A Dedicated middleware for VMD.
 - ADA saves up to 14x time, up to 2.35x memory, 2.91x to 5.21x energy.
 - ADA can render much larger(2.3x) trajectory data.





QTreeIndex?

