Introduction

- Many data processing algorithms used now in neuroscience are open source in nature. (Figure 1)
- However, requisite technical expertise/ expensive hardware reduce usability for broader neuroscience community. Open source is not really open source. (Figure 1)
- Maintaining software, dealing with installation issues, etc, a significant drag on developers' time (where developers are typically grad students or PDs) - good recent ref here: https://www.nature.com/articles/d41586-019-02046-0
- We introduce the Neuroscience Cloud Analysis Platform (NCAP) to address this problem.
- NCAP follows Analysis as a Service model: We host algorithms on the cloud, worry about dependencies, optimizing hardware, code maintenance instead of the end user. Greatly simplifies user experience and also maintenance burden for developer.
- Unlike most "as a service" models, end user does not pay for access. Extending the usability of open source, not restricting it.
- NCAP is not about any single algorithm; it is a software-agnostic platform that takes care of the technical trailblazing of setting up a scalable data analysis pipeline.

Old way: bad for users: have to buy+maintain hardware. Install software, deal with dependencies, keep track of code branches etc. Keeping environments the same over multiple machines is hard, hurting reproducibility. The burden is borne by junior rsearchers. Slow / inelastic. Headaches with large data, slowing down progress and reducing the number of scientific questions we ask. Bad for developers: maintaining software over multiple platforms, helping people with install, manual or no log tracking, etc.

New way: cheap, fast, scalable, easy to use, easy to maintain, enables new scientific questions to be asked

These benefits have been clear for years. Cloud computing is the dominant approach in industry and many scientific fields (cites). It has also been used in neuroscience (cites - i bet the fmri people are ahead of us on this). But under-utilized in many areas of neuroscience. Why? Barriers to entry (hard to set up) and several misconceptions about cost, upload time bottlenecks, interactive mode is hard, etc. In this work we **remove these barriers to entry** and present evidence to dispel these misconceptions.

Results (Figure 2,3,4). Goal here: dispel the fallacies listed above. Establish that the cost per job is low, the upload time isn't a dominant cost per dataset, that working in interactive mode is possible. Also show the expected results: we can process really big datasets quickly by exploiting parallelism.

- NCAP on one dataset is faster/more powerful than local machine, easier to use than cluster. (Figure 2)

- NCAP achieves state-of-the-art performance on established workhorse algorithms (CNMF/-E,DLC). (Figure 2)
- NCAP cost and time scales better than other solutions in terms of dataset size and #.
- NCAP easily extends to multi-step processing for data processed in big batches. (PMD+LOCANMF) (Figure 2)
- NCAP modularizes hardware with software for minimal-hassle, reusable data processing. (Figure 3)
- NCAP can be tailored for many relevant use cases: parameter search, algorithm benchmarking, [online experiments?] (Figure 3)
- NCAP Example. Handles training, evaluation, postprocessing and custom summary statistics for DLC, Froemke data, under fixed resource constraints exponentially better than existing methods (Figure 4)

Methods

- NCAP is implemented in AWS with S3, EC2, IAM, SSM and Lambda functions. Infrastructure designed using boto3, aws cli.
- Monitoring, diagnostics for figure 2 implemented using custom cloudwatch logs.
- Specifics of setting up optimal hardware for each of the algorithms considered
- What about data storage / sharing? Set up an option to keep the data around and share it. Use case we should handle: the algorithm doesn't work well on a dataset I just uploaded; report this to the developer with a link to the dataset so they can fix the issue / improve the code. (What we don't want to do is make this yet another public data sharing website - let other people handle this problem.)
- What diagnostics do you compute + deliver to the user?

Discussion

- For broad community, there a significant barrier of entry to setting up open source software at research-grade quality.
- On computational side, maintaining open source across diversity of platforms diverts time from other projects.
- Consolidating hardware and software on the cloud through NCAP saves time and money for everyone.
- Tightens feedback loop between experimental/computational collaborators for workflows in development
- Establishes potential for new experimental design at scale
- This is worth citing somewhere here: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5860644/
- Set up entry point for getting set up with NCAP (contact us somehow), exit strategy

Figure 1:

Figure 1:

Software Name	Hardware Dependencies	Software Dependencies	Supported Platforms	Expected Cost
~~	GPU ($$)	FFMPEG, OpenCV, Python	Linux	$
~~	Baseline Memory ($$)	Matlab ($$)	Mac, Windows	

Figure 2:

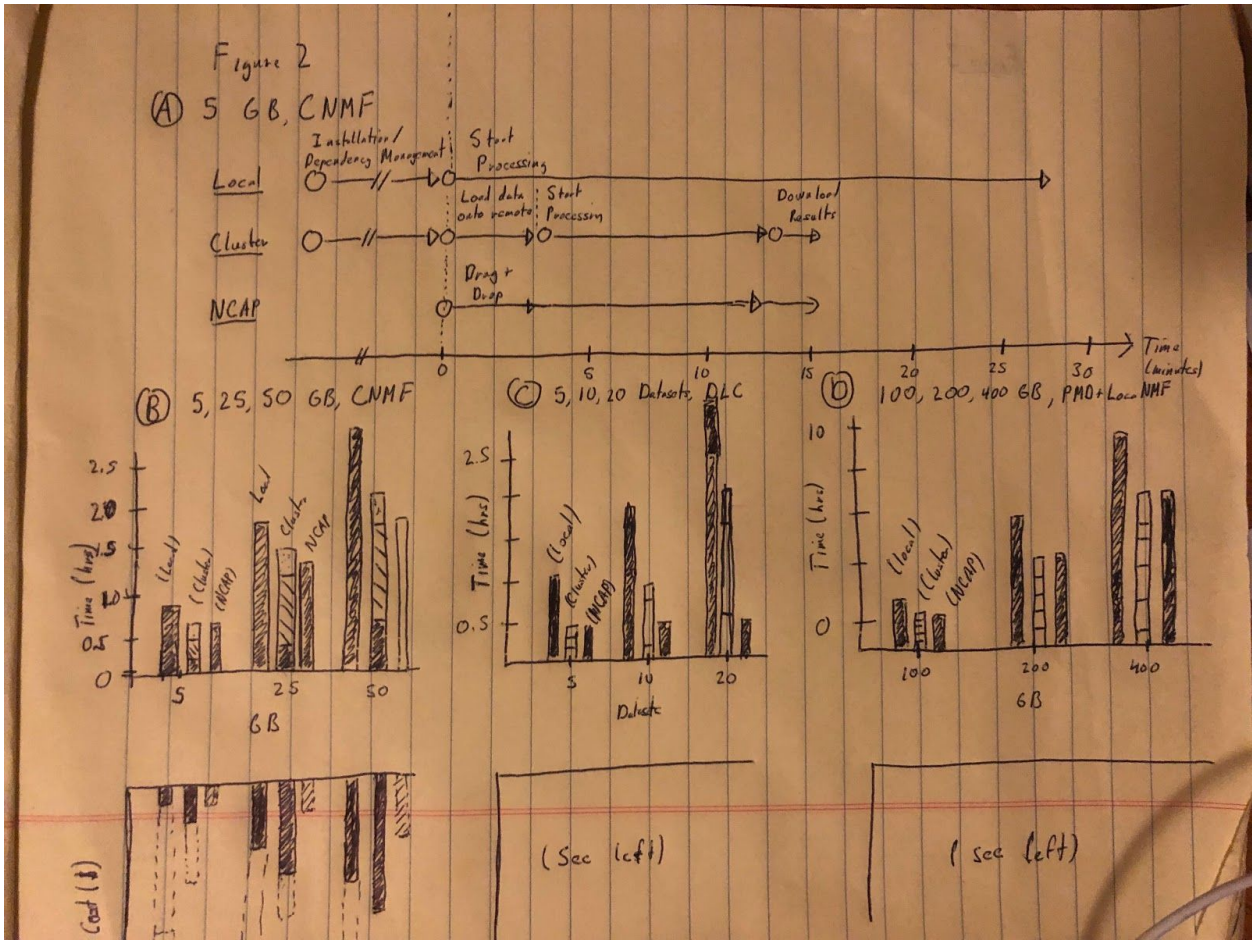


Figure 2
A 5 GB, CNMF
Installation / Dependency Management
Start Processing
Local
Load data remote
Start Processing
Download Results
Cluster
Drag + Drop
NCAP
Time (minutes)
B 5, 25, 50 GB, CNMF
Time (hrs)
Local
Cluster
NCAP
GB
Cost ($)
C 5, 10, 20 Datasets, DLC
Time (hrs)
Datasets
($ec left)
D 100, 200, 400 GB, PMD+Local NMF
Time (hrs)
GB
($ec left)

Figure 3:

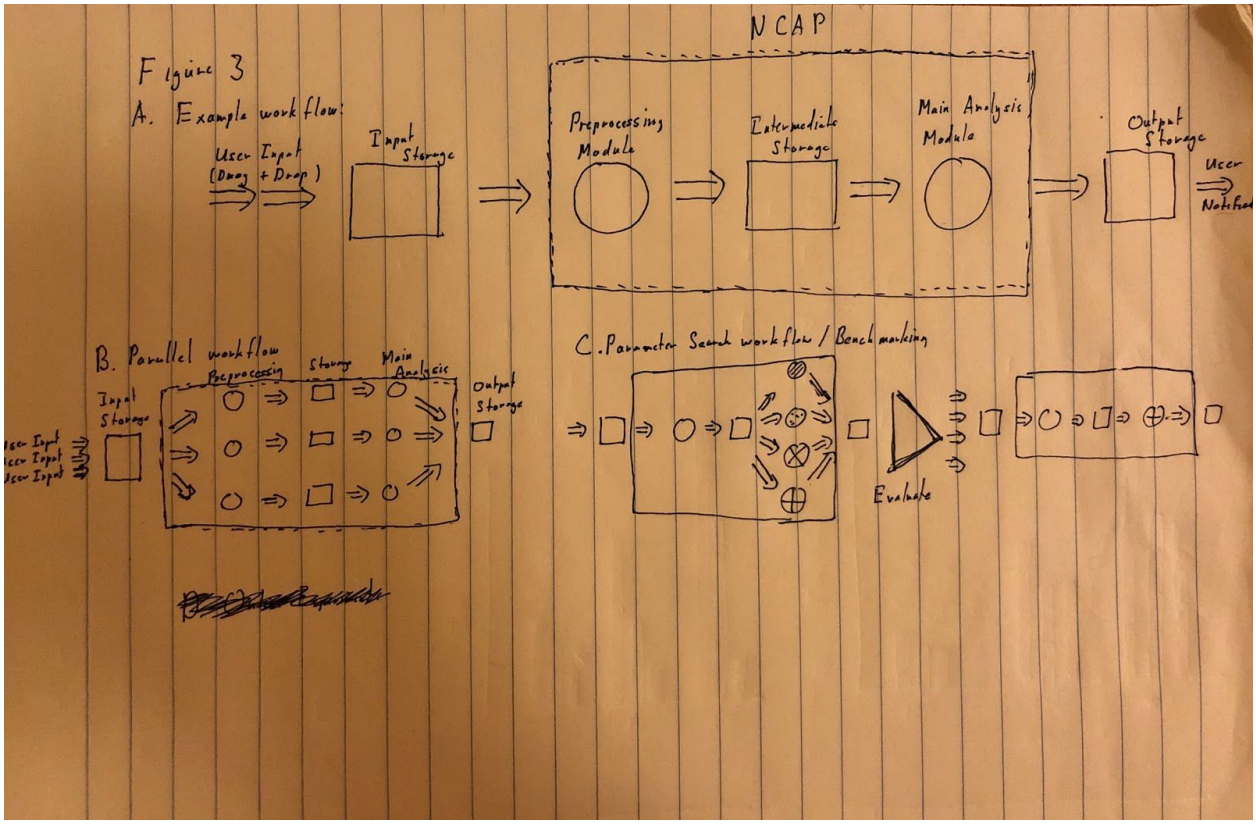
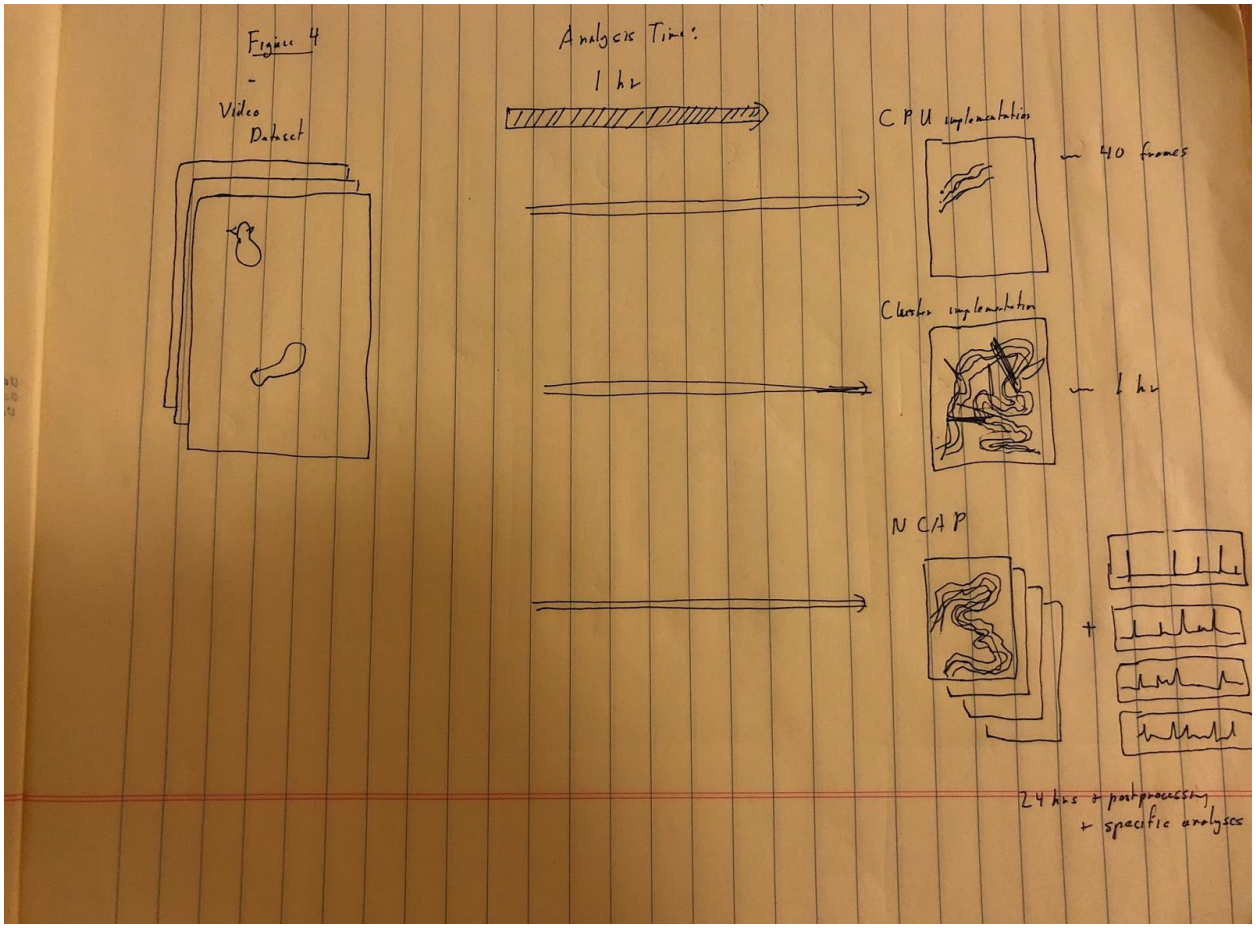


Figure 3
A. Example workflow:
User Input (Drag + Drop)
Input Storage
NCAP
Preprocessing Module
Intermediate Storage
Main Analysis Module
Output Storage
User Notified
B. Parallel workflow
Input Storage
Preprocessing
Storage
Main Analysis
Output Storage
C. Parameter Search workflow / Benchmarking
Evaluate

Figure 4:



Meeting notes 7/18

Include panels/figure for what the power user is doing.

Get more cost information from John about the crossover time.
The approximate crossover time is 50% efficiency. Think about elasticity.

How do we get the credibility on the setup?

Example

PIs might not know the struggle!

Figure 1:
Old way/new way difference? A schematic? A table?

“This burden is often carried by junior scientists ”
How interactive can you make things? Can you include this in something like figure 4?

Something about persistence of storage. Persistence w/ meta analysis. Persistence without meta analysis. Ephemeral. Fork the repo. We are not a data storage site.

Logs, User Uploads