Geomancy: Automated data placement optimization

Oceane Bel
obel@ucsc.edu
Storage Systems Research Center
University of California, Santa Cruz

Abstract

- We developed a new tool, Geomancy, to automatically rearrange data in scientific systems to improve I/O throughput
- We leverage deep learning to learn about I/O trends in user workloads, and reposition high priority data to high performance storage mounts
  - It uses Keras as a machine learning framework
- We use our approach to optimize a physics simulation on a scientific system provided by Pacific Northwest National Laboratories (PNNL)
  - We gain a 49% increase in throughput compared to the standard data layout with all files on one mount **(Home mount)**
  - Matched the highest performing heuristic with **less** data movement

Background

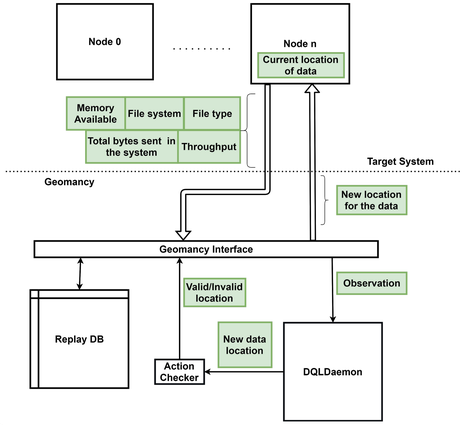
- Many systems use heuristics to balance data layout in a system, but heuristics may fall short due to:
  - Overspecialization to one system
  - Constantly shifting user demands
  - Hardware changes and failures
- Expensive for people to manually exhaustively test and implement all strategies for effectiveness:
  - In our experiment, we place 24 files over 4 mounts, making our search space $4^{24} = 2 \times 10^{14}$
- Systems prioritize heavily accessed (**Manual prioritization**) or large files for fast nodes:
  - Can overengineer layout or put heavily contended files on slow mounts

PNNL System

Geomancy optimizes a scientific simulation that runs on a node of the replicated BELLEII system containing 4 mounts:

- Home: Isilon NFS
- Tmp: RAID 1 Volume: 223 GB (2 x 240GB SSD, 2.5", SATA, 6Gbps, 1 DWPD, 3D)
- RAID 5 Volume: 9313 GB (6 x 2TB, 2.5", 7200RPM, SATA)
- usbtmp type: externally mounted Dell st2000dl HDD

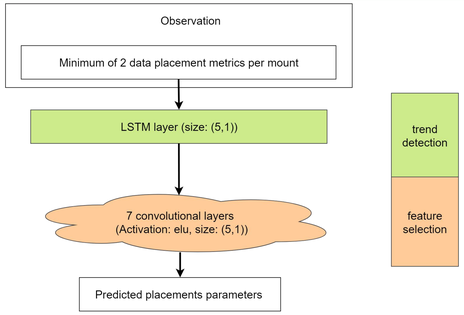
The System



- System overview:
  - Geomancy interface, ReplayDB, DQLDaemon
  - Data collection and access points to the system
- Access metrics are continuously gathered by the system
- Geomancy's interface directs the data flows between the **DQLDaemon**, **ReplayDB** and the **target system**
- Performance data stored into ReplayDB (sqliteDB)
- Categorical parameters (e.g. file paths) hashed into integer values
- **Action Checker**: checks if a **valid** location is chosen by DQL Daemon
  - Looks at configuration file of available node
  - Each node has a list of restrictions, such as:
    - IP address that can write to the node
    - Valid actions (read only nodes, read write nodes, etc.)

Geomancy's neural network

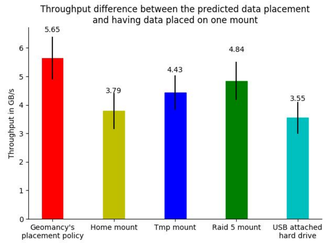
- The training data contains **location ID**, **data type**, **latency of action** and **number of bytes accessed**.
- Minimum of 2 accesses per file on each mount needed for training data
- **30%** of the training data is used to validate the model. The validation data is never seen during training
- Each index in the output array corresponds to a node available in the system
  - Links predicted throughput to nodes in the system
- Our neural network will predict the trends of the throughput of the available nodes.
  - This allows Geomancy to predict how the throughput will change for each node during the next accesses.
  - It also allows Geomancy to understand the access number for which to change the mount the data will be accessed on.

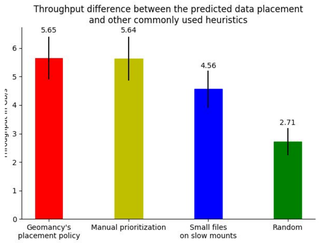


LSTM layer relates temporal dependencies and the convolutional layers build a sparse representation of the data

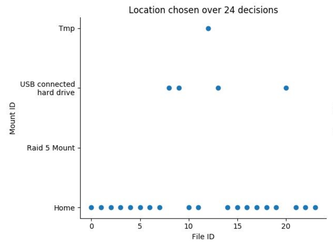
- This allows the neural network to disentangle the information present and build a clearer representation of which action caused a certain state

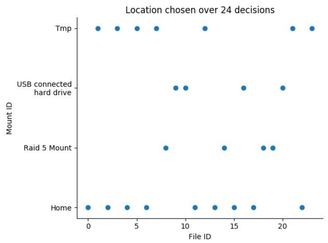
Preliminary results





We compared Geomancy's placement policy against three other heuristics, and placing all data on the same mount. Although manual prioritization (**below** on the **right**) of the files did as well as Geomancy's placement (**below** on the **left**), Geomancy moves **less** files between nodes to achieve the same speedup.





Future Work

- Bottleneck detection and mitigation:
  - See if Geomancy can detect bottlenecks in systems
  - Modify Geomancy to detect and remedy bottlenecks on the fly without human intervention
- Pre-emptive data movement:
  - Pre-empting workload changes by changing locations of data depending on the time of day
- Extension to data block location optimization

Acknowledgements: This research was conducted in collaboration with CERN and PNNL, who provided the data used for the experiments. I would like to thank professor Darrell Long and professor Ethan Miller for their guidance and help on this poster. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. We would like to acknowledge the Department of Energy and the National Science Foundation for sponsoring the research described here.