Evaluating Named Data Networking for Large Scientific Data

Susmit Shannigrahi

Colorado State University & Berkeley Lab

Introduction

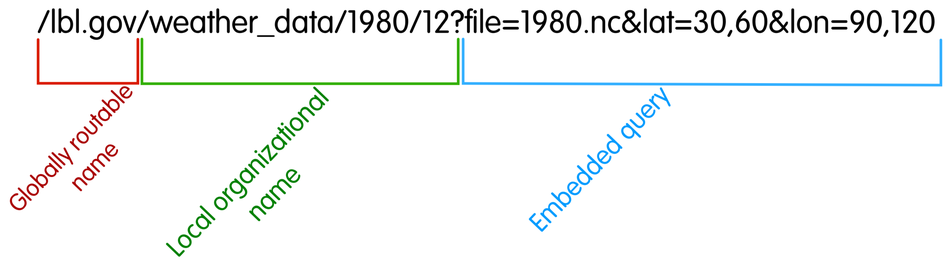
Large distributed scientific data requires:

- Catalogs for tracking location
- Complex application and middleware to access data

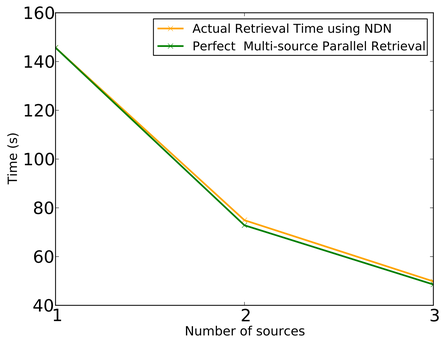
Named Data Networking (NDN)

- Addresses and locates data by name
- No need to track data locations
- Adapts retrieval strategies to content availability and network conditions

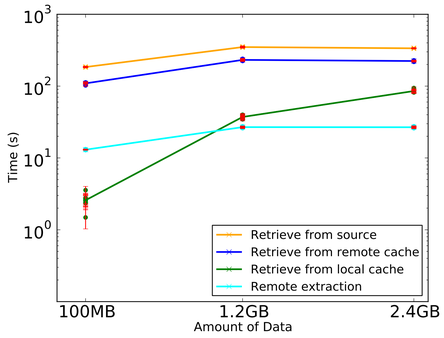
Naming of ndn interest for remote subsetting



Retrieval performance of ndn

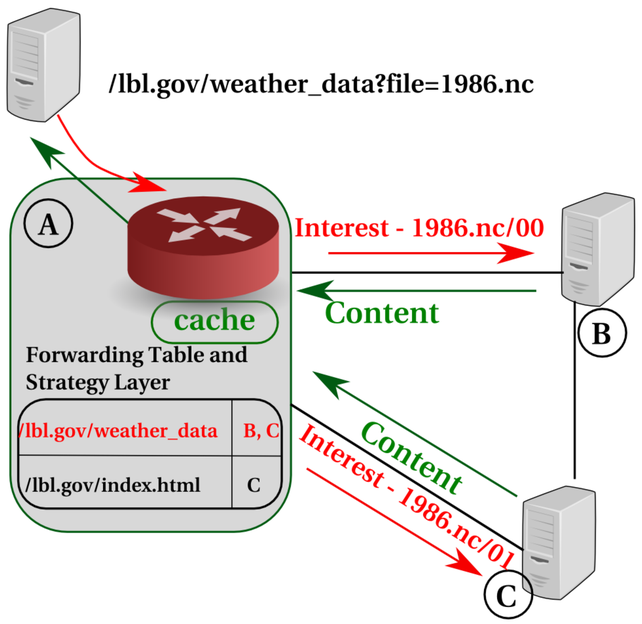


Parallel Retrieval Performance



Comparison of Retrieval Strategies

Benefits of ndn for large scientific data



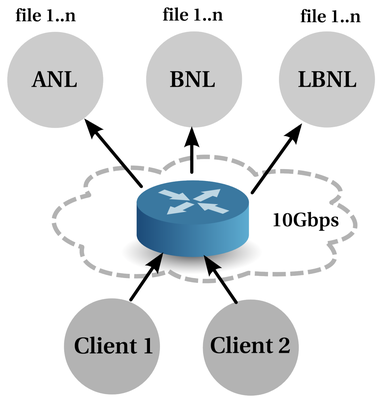
Content Retrieval in NDN - Parallelized Scenario

- Parallel retrieval from multiple sources without using pre-built catalogs
- Strategy is function of network, not application or middleware
- Returned data is **cached** at intermediate nodes
- Remote instructions can be encoded in interest name

Comparing ndn with other alternatives

	NDN	*CDX*	*DataCutter*	*EVPath*	*GridFTP*	*OpenDAP*
Transparent data access						
Application independent strategies						
Multi-source retrieval						
Remote subsetting on multiple sources						
Caching						
Name supported embedded query						
Where Implemented	NW	APP	MW	MW	APP	APP
Approximate LOC	595	2.2K	17.8K	37.2K	66.9K	70.6K

Experiment Methodology



Testbed Topology

- Dataset consisted of multiple files, 33GB in total. Each file was 105MB and held one month's data.
- Exp 1: Multi-source retrieval using NDN
- Exp 2: Comparison of four retrieval strategies
  - Whole dataset is retrieved from data source, intermediate cache and local cache, respectively. Data is extracted locally
  - Query is encoded in interest. Data is extracted at remote source and retrieved

Findings

- Parallelized retrieval using NDN achieves ideal load-balancing and speedup
- Even for large data, one intermediate cache can speed up retrieval by over 33%
- Name supported remote subsetting takes 90% less time than retrieving data from source
- Flexibilities in NDN easily support diverse retrieval scenarios - single source, multi-source

Conclusion

- NDN natively supports workflows involving large scientific datasets and reduces need for complex applications and middleware
- Remote operations reduces need for large transfers and is easily supported by NDN
- Flexible strategy layer can be adapted for intelligent data retrieval with little effort

References

- Networking Named Content - Conext '09 - Jaconbson et. al
- CMS computing: technical design report - CERN, 2005.

I thank Alex Sim, John K. Wu, Suren Byna, Inder Monga and Brian Tierney from LBNL and Christos Papadopoulos from CSU for their help and feedbacks.

http://cs.colostate.edu/~susmit susmit@cs.colostate.edu