Unidata.binaries.data:
Near Real Time Data Relay Using News Server Technology

Anne Wilson
Unidata Program Center
Boulder, Colorado

November 4, 2004

Usenet History

- Started in 1979, UUCP based
- NNTP (Network News Transport Protocol) became standard in 1986
- Streaming became part of protocol in 2000
- Current volume:
  - Tens of NSPs (News Service Providers)
  - Terabyte/day [Giganews]
  - 1 to 10 million servers [Pufrug]
  - 25 million users [Pufrug]
  - Over 100,000 newsgroups

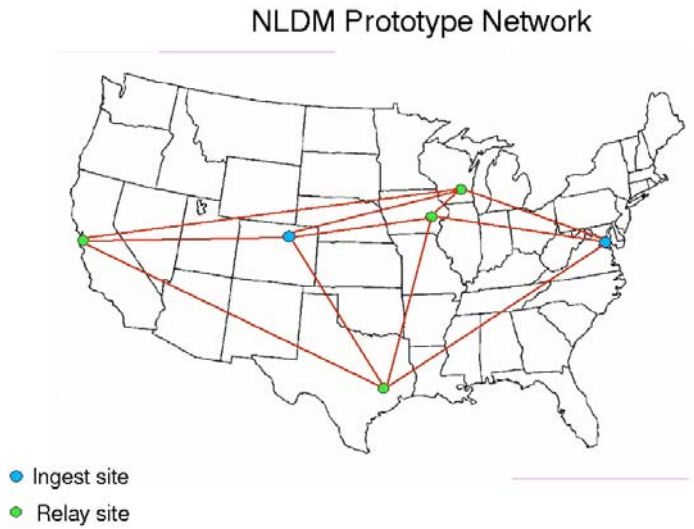
Usenet

- Decentralized, heterogeneous chaos of information, opinions, pictures, music
- And it still works!
- "*Come to think of it, there are already a million monkeys on a million typewriters, and the Usenet is NOTHING like Shakespeare*!" – Blair Houghton

NLDM Data Relay

- NNTP-based (Network News Transport Protocol) based data relay network
- Uses INN (Internet News)
  - Freely available, open source
- Feed types:
  - CONDUIT: forecast model output
  - CRAFT: level II radar
  - HDS: analysis and forecast fields
  - IDS, DDPLUS: large quantities of small text products
  - NEXRAD: level III radar products
  - UNIWISC: satellite imagery
  - NIMAGE: satellite imagery, up to 20MB

NLDM Sites



Hostname	Location	Function	OS
imogene.unidata.ucar.edu	Boulder, CO	Ingest	Linux
atm.geo.nsf.gov	Washington, D.C.	Ingest	Solaris
ldm.iihr.uiowa.edu	Iowa City, IO	Relay	Linux
tempest.aos.wisc.edu	Madison, WI	Relay	Linux
bigbird.tamu.edu	College Station, TX	Relay	Linux
methost24.met.sjsu.edu	San Jose, CA	Relay	Linux
joey.unidata.ucar.edu	Boulder, CO	Stats Processing	Linux
conan.unidata.ucar.edu	Boulder, CO	Stats Display	Solaris

NLDM Statistics

- Tracking
  - Latencies: maximum, average, cumulative
  - Products received
  - Bytes received
  - Number of inbound connections
  - Paths taken by articles
- http://my.unidata.ucar.edu/content/projects/nldm/relayStats/plotStats.php

News Relay and Data Relay Commonalities

- Fast, reliable transmission
- Logical grouping of domain into names
- Local management of data
  - File to disk
  - Pipe to a process
  - Invoke a program

News Relay and Data Relay Differences

News (INN)

- "articles"
- storage on order of days, weeks
- "too old" defined in terms of days
- designed to handle long term peer outages
- originally text based, requires encoding of binaries
- supports "readers"
- "peers"

Near Real Time Data Relay (LDM)

- "products"
- storage on order of minutes, hours
- "too old" defined in terms of seconds
- handles short term peer down times
- designed to handle binary data
- "upstream", and "downstream" sites

Push-based Article Propagation

LDM:

- Streaming transmission

INN:

- Streaming transmission
- Batched transmission
  - Via command line
  - Via file placement

Streaming Transmission

Relevant protocol messages:

LDM	NNTP	Function	Pipelined?
COMINGSOON	IHAVE	Ask first, wait for single response	No
	CHECK	Ask first, collect responses	Yes
HEREIS	TAKETHIS	Send without asking	Yes

LDM:

- "PRIMARY" designated request uses HEREIS
- "SECONDARY" designation uses COMINGSOON
- Configured by user, static
- Uses RPC layer

INN:

- CHECK allows construction of list of articles to be relayed based on collection of responses
- Dynamic switching between CHECK and TAKETHIS based on article rejection rate and configuration parameters
- Uses socket layer directly

Routing

INN – Flooding Algorithm:

- Automated routing via high interconnectivity, massive redundancy
  - Bandwidth usage mitigated by automatic CHECK/TAKETHIS switching
- Each site serves as a sender and a receiver
- “Pools” of articles
- Articles arrive at destination via fastest route possible
- Reliable under site failure if sufficiently interconnected

LDM:

- Multiple “PRIMARY” connections can serve like flooding
- In practice, more manual topology configuration, more frugal interconnectivity
  - Efficient bandwidth usage
  - More impact due to site failure

Product/Article Storage

LDM:

- Single memory mapped file (product queue)
  - Short term storage (minutes, hours)

INN:

- File-based
  - Longer term storage (days, weeks)
  - Supports “readers”, pull based retrieval
  - Requires expiration
- Memory mapped file-based
  - Short term storage (minutes, hours)
  - Physical buffers can be logically grouped into “meta” buffers
    - Physical buffer mgmt can be interleaved or round robin
- Overview file reflects current state of holdings
  - Useful for readers, cataloging systems
- Unified storage interface
  - Article “tokens” are handles to articles

Product/Article Headers

LDM:

- Fixed size header of eight fields:
  - feed type, product ID, origin, injection time, sequence number, signature, size

INN:

- Required NNTP headers
  - Subject, Newsgroups, From, Date, Message-ID
- Optional NNTP headers:
  - e.g., Content-Transfer-Encoding, Distribution,…
- Extra headers:
  - e.g., X-Product-ID, X-Signature, X-FeedType, X-SeqNum, …
  - Can be used as metadata
  - Useful for browsing, cataloging systems

Pull-based Transmission

INN:

- Protocol supports pull based retrieval
- Can retrieve:
  - Entire article
  - Article head
    - Useful for browsing metadata
  - Article body
- Designed for interaction

LDM:

- Does not support pull based retrieval

The Namespace

LDM:

- 31 feed types, bit map-based
- Finer matching uses regular expressions matched against prod ID
- Names could be expanded significantly in subsequent versions
- Not dynamic

INN:

- String-based, hierarchically structured namespace
  e.g., unidata.binaries.nexrad.<stationID>
- Cross-posting supported
  e.g., NLDM posts KABC N0R product to
  unidata.binaries.nexrad.KABC and
  unidata.binaries.nexrad.N0R
- Can by dynamically created and destroyed
- Subscription list matching is string based with wild match chars
  e.g., unidata.binaries.nexrad.N?R
- “Negative” subscriptions
  e.g., unidata.binaries.nexrad.N?R, !unidata.binaries.nexrad.N3R

Backlog Handling

INN:

- Maintains queue of tokens of undelivered articles for each peer
- Can relay article as long as article is in storage
- User configurable maximum size for queues
- Queue size trimmed from front so most recent articles sent
- Age of article not a factor in pushing
    - May be rejected by age upon reception

LDM:

- Sends product to downstream if:
    - Product in queue
    - Downstream connected
    - Product age within range specified by downstream

Connection Management

LDM

- One connection per REQUEST line
  - User configures number of connections
- Connection number is static

INN

- User configures maximum number of connections
  - global maximum
  - per peer maximums
- System spawns and destroys connections dynamically
  - maintains queue of article tokens to be delivered
  - two queue thresholds: low, high
  - adds connection if above high
  - drops connection if below low

Network Level Control

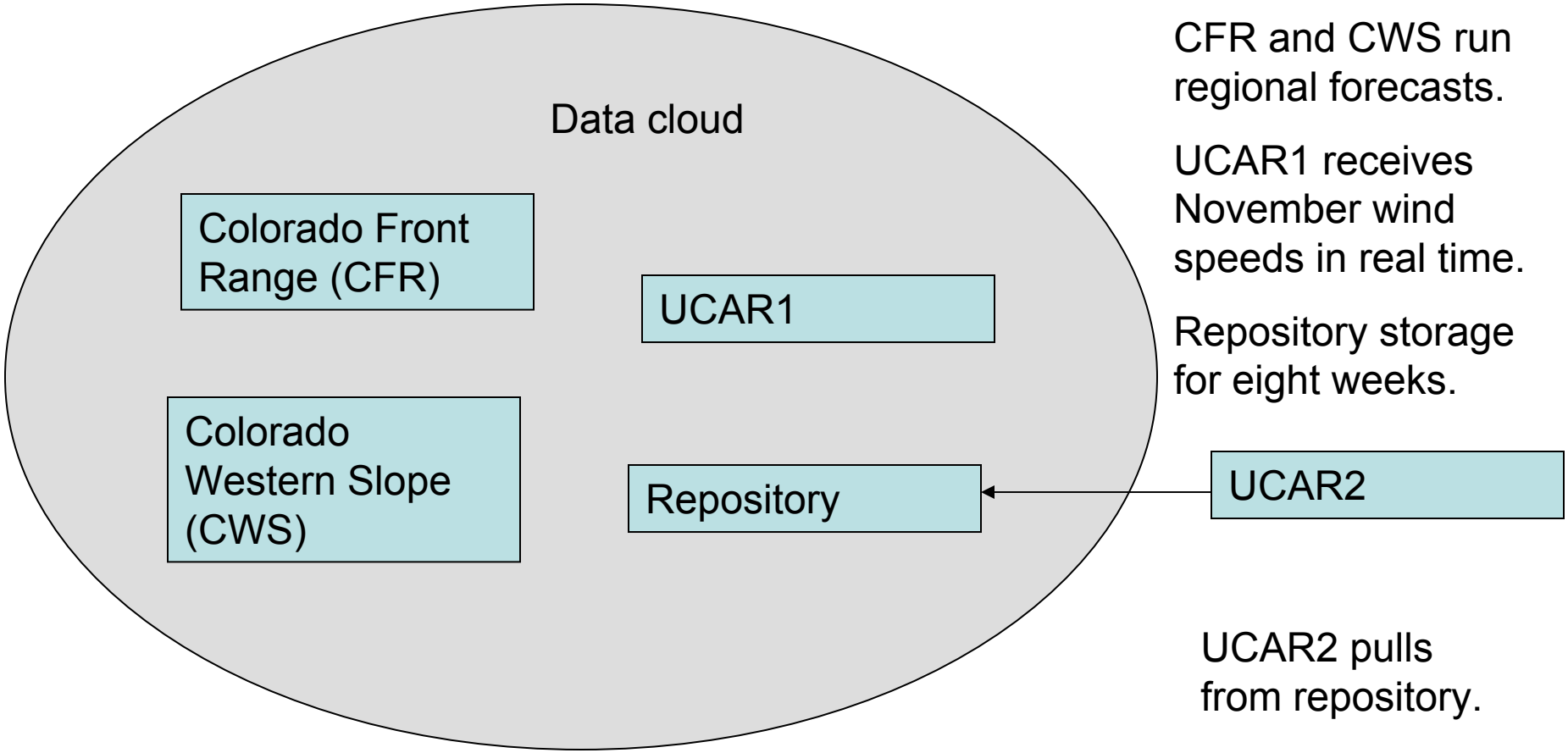
INN

- "Control" messages allow sites to automatically:
  - Add or remove a group name
  - Send a list of all locally known groups
  - Inform a site of having a particular product
  - Request a site to send a particular product
- Valuable because sites must know of new group names before they can accept articles posted to those groups

Either LDM or NLDM could be configured to respond to specially tagged messages via local product/article management.

Possible Scenario: CoForecastProject

- Multiple researchers collaborating at different geographical locations



CoForecastProject (cont.)

- CFR and CWS sites:

Colorado Front Range (CFR)

Colorado Western Slope (CWS)

1. Run model, e.g. WRF
2. Determine some metadata, e.g.
   X-InputModel: AVN
   X-InputModelTime: <yyyymmdd_hh>
   X-Windspeed-Min: <min>
   X-Windspeed-Max: <max>
   X-Windspeed-Units: meters per second
3. Create articles complete with metadata
4. To be continued…

CoForecastProject (cont.)

- Product group naming scheme:

<projectName>.<region>.<model>.output.<yymmddhh>.<level>.<parameter>

e.g.,

coForecastProject.cfr.wrf.output.2004110404.500.windSpeed

coForecastProject.cws.wrf.output.2004110404.500.windSpeed

Colorado Front Range (CFR)

Colorado Western Slope (CWS)

4. Notify network of new group names
5. Post new products to appropriate groups

CoForecastProject (cont.)

UCAR1

UCAR1 subscribes to:
coForecastProject.*.wrf.output.200411??04.*.windSpeed

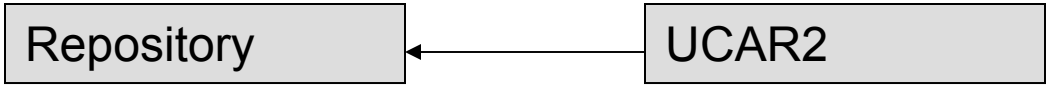
Receives data as soon as is available.

Repository

Repository subscribes to:
coForecastProject.*.wrf.output.*.*.*

Also receives data as soon as is available.

CoForecastProject (cont.)



UCAR2: wants to retrieve all WRF runs having wind speeds greater than some maximum for all pressure levels for November

1. Connects to Repository.
2. “Discovers” newsgroups coForecastProject.*.wrf.output.200411??.*.windSpeed.
3. One by one retrieves headers from articles in these groups.
4. Examines X-Windspeed-Min headers to find those less than maximum.
5. Pulls those articles.

Benefits

- Many features
  - Efficient streaming
  - Automated routing
  - Mix and match options for article storage, both short and longer term
  - Automated connection management
  - Browsing and pull-based retrieval support
  - Ability to attach metadata
  - Broad, dynamic name space
  - Intuitive subscription syntax, including negative subscriptions
  - Overview support for cataloging systems
  - Backlog handling
  - Network level control, with PGP verification of control messages
  - Password authentication for readers and posters
  - Resource tracking, notification of problems via email
  - Interactive command line interface to server
  - Free

- Lots of NNTP-based software available

Remaining Questions

- Detailed comparison of efficiency between LDM and NLDM
- Unexpected issues in wrapping of existing decoders

Costs

- Encoding
  - Visit of every byte could be combined with computation of signature
  - Decoding process required with NLDM but not LDM
  - Protocol could be modified
    - Would be incompatible with other NNTP based software
- Configuration complexity
- Working within an open source community

JNLDM

- Java and NNTP based receive-only client
- Intuitive, robust GUI
- Received CONDUIT data on laptop
- “Integrated” with Unidata Integrated Data Viewer (IDV) to display CRAFT data
  - Made a subclass of IDV
  - Notified IDV when products arrived from selected stations

What Next?

- Unidata Program Center (UPC) is evaluating our resource allocation
- LDM6 will serve us for the next 2 – 3 years