LDM 6.4: Combining Power and Flexibility

Steven Emmerson

New Features
Future Directions

New Features in 6.4

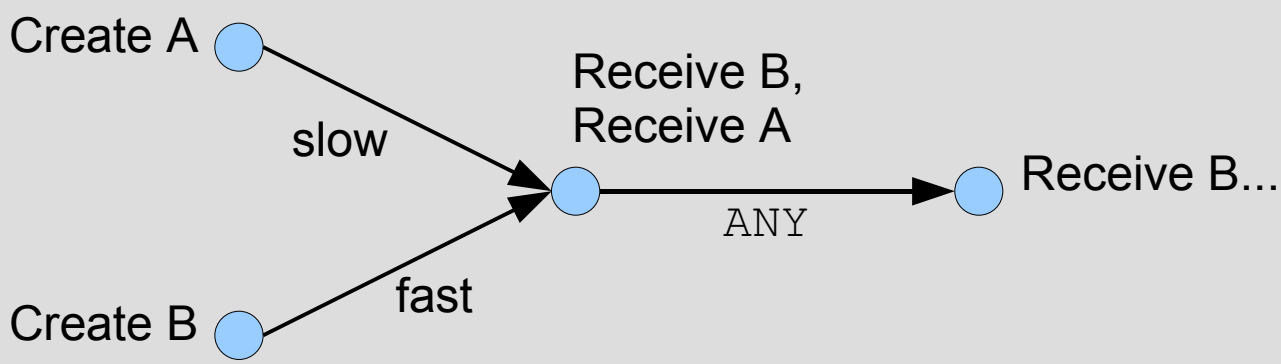
- Port independence
- Requests for data by signature
- Upstream filtering
- Receive-mode auto-shifting

Port Independence

- Before 6.4
  - Must use 388
  - Requires superuser privileges to install
  - Requires education of network administrators
- 6.4
  - May set default port at build-time
  - May override port at run-time
  - May specify upstream port

```
REQUEST ANY .* idd.domain:388
```

Requests for Data by Signature



- Problem: Reconnection often causes data-loss in aggregated feeds because product creation-time determines request “from”-time
- Solution: When reconnecting, use signature of last successfully-received data-product

```
LDM-6 Desired product-class: 20050623154153.405 TS_ENDT
{{ANY, ".*"},{NONE, SIG="11719082928ccf26c433a556c0c8d8c8"}}
```

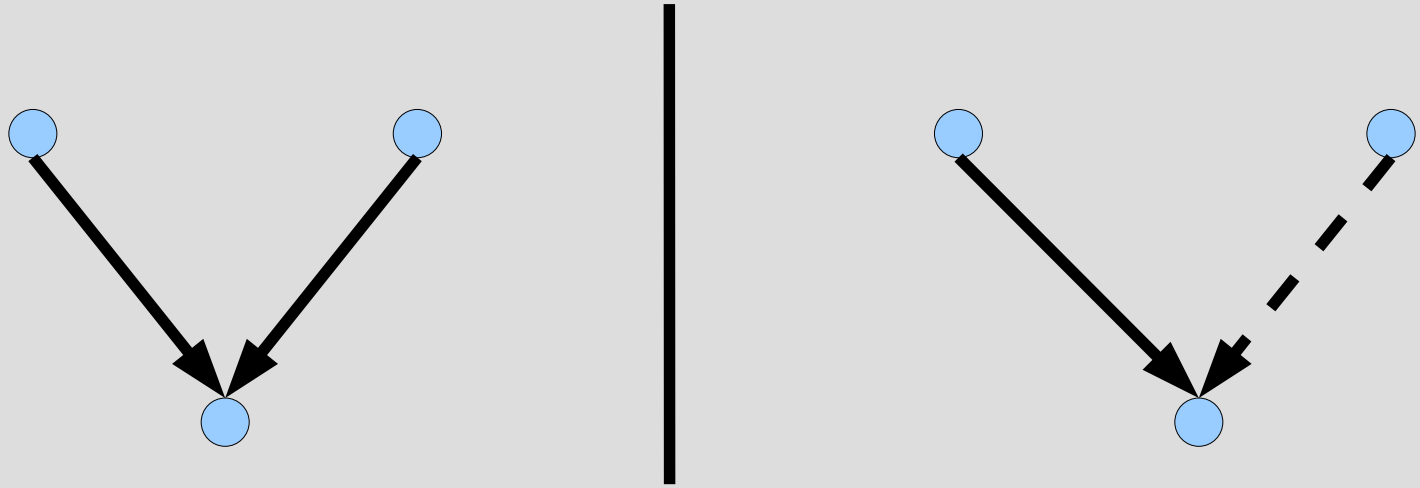
Upstream Filtering

- Pre-6.4
  - No way to prevent downstream LDM from obtaining all data-products in a feedtype
  - Lack of user-definable feedtypes makes it difficult to restrict data flows
- 6.4
  - ALLOW-entry enhanced with OK-pattern and NOT-pattern:

```
ALLOW IDS|DDPLUS \.com$ ^SAUS ^SAUS4
```

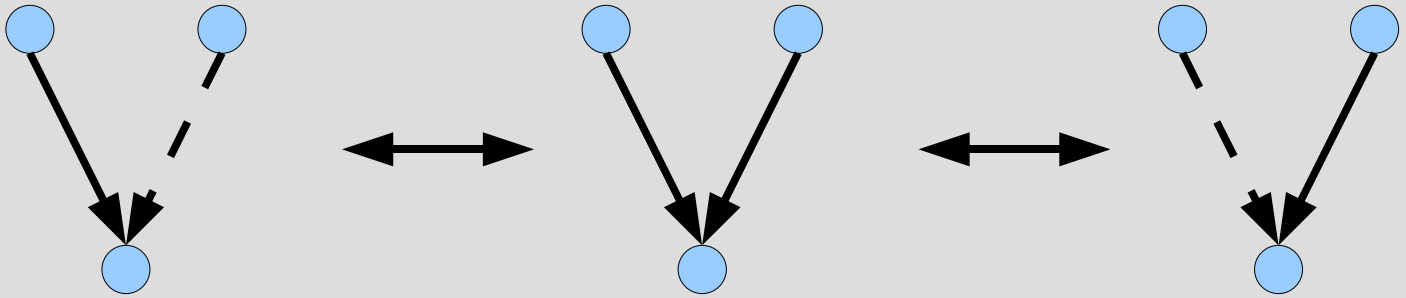
  - Allows user-definable data-flows to be managed by product-identifier

Pre-6.4 Receive-Modes



- Static PRIMARY and ALTERNATE feeds means trading reliability for bandwidth
  - 2 PRIMARY feeds is reliable but doubles the bandwidth usage
  - PRIMARY + ALTERNATE feeds reduces bandwidth but is unreliable

Receive-Mode Auto-Shifting in 6.4



- First feedtype/pattern REQUEST-entry is initially PRIMARY feed
- All other REQUEST-s with same feedtype and pattern are initially ALTERNATE feeds
- Downstream LDM automatically shifts between PRIMARY and ALTERNATE modes depending on insertion success-rate

Future Directions

- 6.4 is about as far as the LDM can go without changing the protocol (i.e., LDM 7)
- Given a protocol change, how far should one go?
  - Eliminate RPC in favor of true peer-to-peer communication?
  - Use overlay networks?
  - Rewrite in C++?
- For the short-term:
  - Deploy
  - Fix bugs
  - Work on UDUNITS package


© Grenada Explorer