

An End-to-End Approach to Globally Scalable Programmable Networking

Micah Beck, Terry Moore, James S. Plank
Logistical Computing and Internetworking Lab
Computer Science Department
University of Tennessee

Reviewed by Qian Wan for CS6814 in Fall 2003.

Key concepts and names

- End-to-end principles for the Internet
- Logistical Networking(LoN)
- Internet Backplane Protocol(IBP)
- **Logistical Network Computing(LoNC)**
- **Network Functional Unit(NFU)**
- Logistical Computing and Internetworking (LoCI) Laboratory at <http://loci.cs.utk.edu/>
- Internet2 at <http://www.internet2.edu/>

Background Knowledge Preview

- Reference [2] on “end-to-end arguments”
- Latest Publications link at <http://loci.cs.utk.edu/>

Logistical Networking
Sharing More than the Wires

The Internet Backplane Protocol:
A study in Resource Sharing

Background Details

- End-to-end argument
 - against low-level function implementation
 - different arguments for different applications
- Logistical Networking
 - logistics is the root of the name
 - expose and use intermediate resources, such as storage and computation
- Internet Backplane Protocol
 - middleware

Introduction

- Internet is still mainly for data communication
- Can we expand the power of Internet
- Scalability issue
- Various services requirements issue
- Robust and fast models



IBP deployment diagram copied from <http://loci.cs.utk.edu/>

Building on Logistical Networking

- Logistical Networking as a solid infrastructure support
- IBP serves as the “best effort” storage service
- exNode implemented to aggregate byte arrays in IBP depots to form pseudo-file

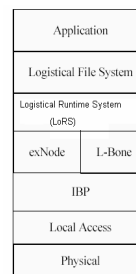


Figure 1: The Network Storage Stack

Computation Scalability

- Design following the end-to-end principles
- Semantics of the service need to be simple and weak
- Best effort nature of service, which is similar as IP and IBP
- Fragmentation of different type of services
 - end-to-end
 - end-point
- Intermediate nodes service computation states
 - Stateless if no endpoint check
 - Make the state explicit if allow endpoint check

More Computation Scalability Issue

- Correctness through endpoint checks, this also proves the effectiveness of the end-to-end arguments
- Security is still a challenge
 - The approach should follow end-to-end rules

Detailed LoNC components

- Network Function Unit(NFU)

- Execution layer resource
- Generic services matching IP and IBP: resource aggregation, fault detection and global addressing
- Execute NFU_op operations on the depot specified in parameters

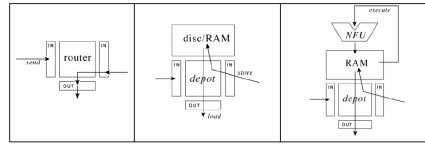


Figure 3: Intermediate nodes to manage bandwidth (IP router), storage (IBP Depot), and computation (NFU-enabled Depot)

NFU_op(depot, port, operation, soft, cap_1, ... cap_n)

Parameters in NFU_op are cryptographically secure names

Data Structure of Aggregation of Network Computation

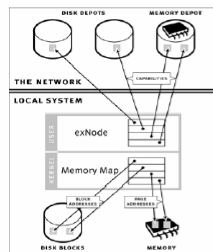


Figure 4: The exNode provides a uniform view of data and process state in LoNC

Since the operation commands are NFU_op and they operate on specified IBP depots, the aggregation of processes is similar as the aggregation of storage.

Speed up NFU Performance

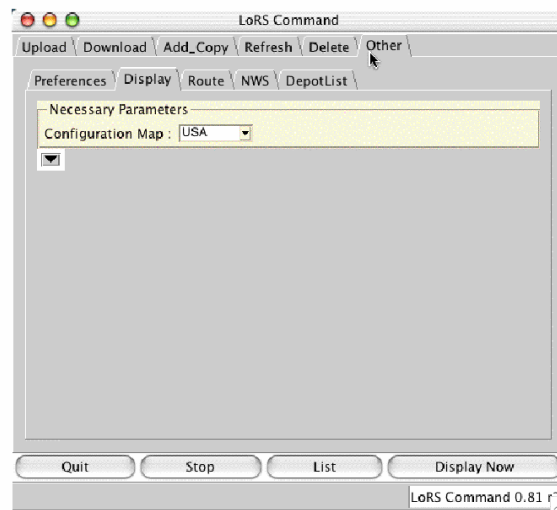
- Pipeline
- Stored program model with caching?

Conclusion

- The authors believe in the key idea of pushing functionality from lower layer to higher ones to get better results when people can identify or model efficiently based on the end-to-end arguments[2].

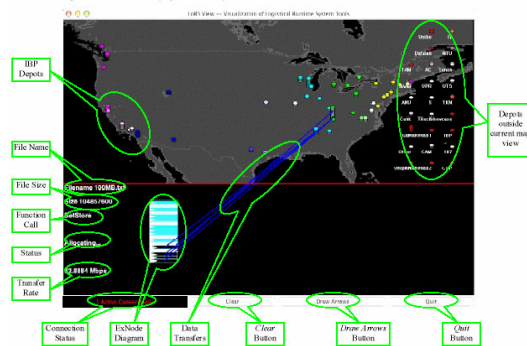
Extra from the authors' group web pages

Figure 2: LoRS Command Window



Storage Upload

Figure 12: LoRS View display features - Upload



Storage Download

