

# Publish-Subscribe for High-Performance Computing

Zhang Xianwen

Dresden University of Technology

04.01.2006

[s3274573@inf.tu-dresden.de](mailto:s3274573@inf.tu-dresden.de)

# Outline

- Part I: Introduction
- Part II: Basic Echo Functionality
- Part III: Event Filtering and Transformation
- Part IV: Summary
- Part V: References

# Part I

## Introduction

# Motivation

- Event-based communication mechanisms are widely used in distributed applications and services.

# Motivation

- Traditional publish-subscribe systems can provides:
  - +Scalability
  - +Adaptability
  - +Fault tolerance
  
- But for the requirements of high-performance computing in grid environments:
  - -Substantial overhead
  - -Delivering latency
  - -Significant loss of bandwidth

# Overview of ECho

- An event delivery middleware system for high-performance computing
- An anonymous group communication mechanism
- Network traffic for multiple channels is multiplexed over shared communications links

## Part II

# Basic Echo Functionality

# Efficient Event Notification

- Decentralized ,lightweight channels
- Network traffic for individual channels is multiplexed over shared communications links
- Provides inter- and intra- process communication

# Efficient Event Notification

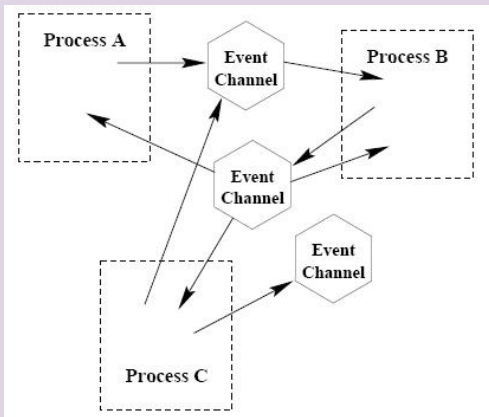


Figure 1: Processes using Event Channels for Communication.[1]

# Characteristics of ECho

- Support for efficient transmission and handling of fully typed events
- Support the robust evolution of sets of programs communicating with events

# Base Type Handling and Optimization

- Echo event types describe C-style structure made up of atomic data types and extend it to support null-terminated strings and dynamically sized arrays.
  - Achieve several performance optimizations(optimize buffer usage,minimize copying...)
- use a wire format that is equivalent to the native data representation(NDR)of the sender.
  - Wire format:the representation of the data in transmission
- converts the wire format(NDR) to the receiver's native format at receiving side
  - reduce the latency in heterogeneous message exchange.

# Type extension

- allowing variation in data types associated with a single channel.
  - source may submit an event whose type is a superset of the event type of its channel
  - sink may have an event whose type is a subset of the event type of its channel

# Basic Data Exchange Performance

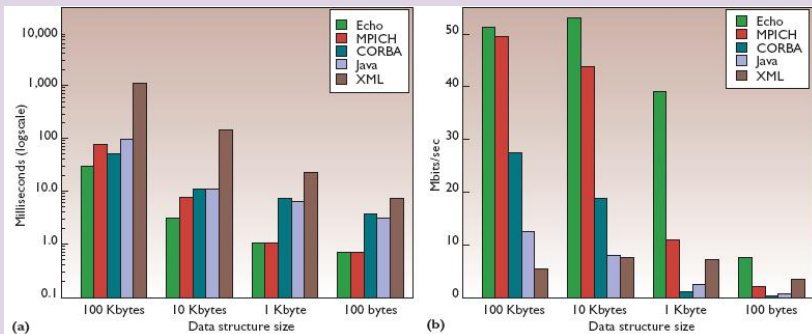
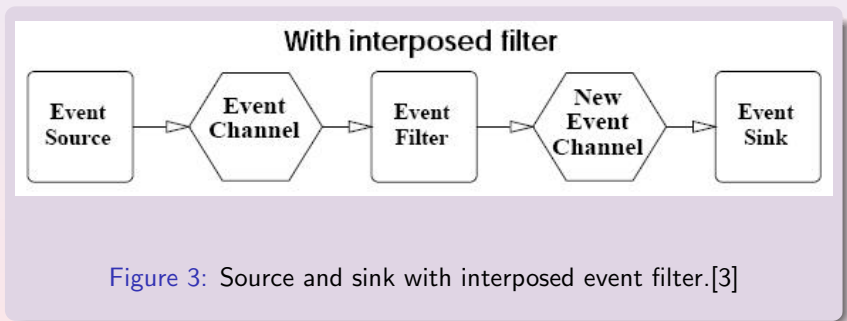


Figure 2: Echo's performance with respect to (a) round-trip latency (b) delivered bandwidth. [1]

## Part III

# Event Filtering and Transformation

# General Model



# General Model

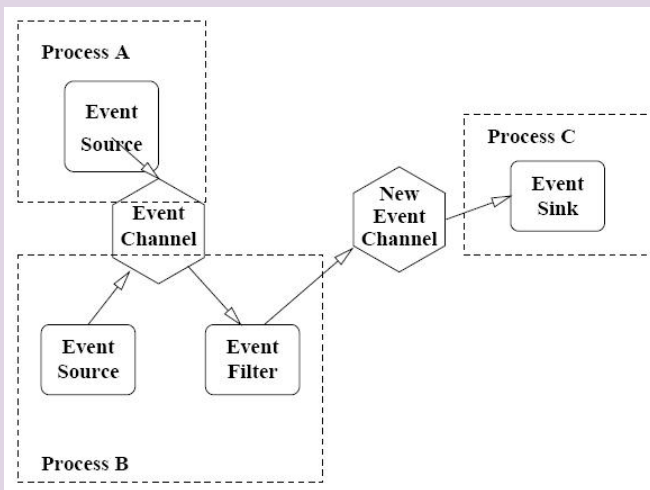


Figure 4: Filter with more than one source.[3]

# General Model-Derived Event Channels

- Creating new derived event channel through derivation function  $F$
- $F$  is in the original channel, specified by the sink

# Derived Event Channels

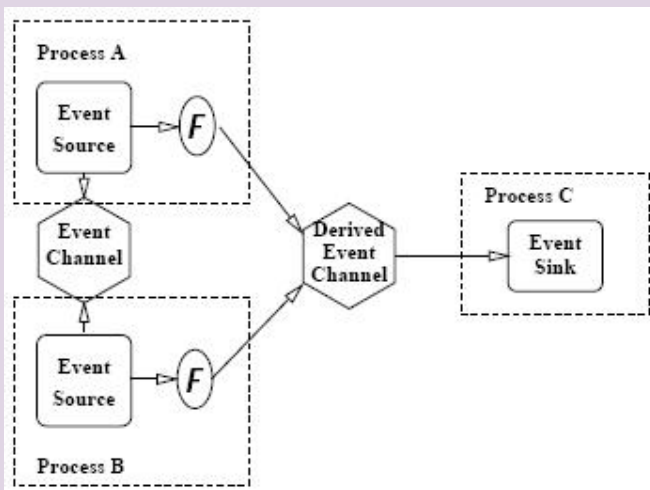


Figure 5: A derived channel and function F moved to event sources.[3]

# Mobile Functions and the E-Code Language

- The function  $F$  is expressed in E-Code and creating a native version of  $F$  is via dynamic code generation on the source host.
- Echo currently supports two kinds of derived event channels.

# Overlays:EVPath

- EVPath:a middleware for facilitating the construction of event notification overlay networks.
- using stones to implement the path abstraction dynamically

## Part IV

# Summary

# Summary

- Echo meets the needs of the high-performance and distributed applications.
- Echo takes the advantages of PBIO for minimizing the data copying.
- Echo provides dynamic type extension for lack of a priori knowledge.
- Echo's Derived Event Channels allow sink-specified event filtering and event data reduction to be applied on the source and of event transmission.

## Part V

# References

# References

- [1]Greg Eisenhauer,Karsten Schwan,Fabian E Bustamante:  
Publish-Subscribe for High-Performance Computing.
- [2]Greg Eisenhauer Fabian E. Bustamante Karsten Schwan:  
Event Services for High Performance Computing. *Proceedings  
of High Performance Distributed Computing (HPDC-2000)*.
- [3]Greg Eisenhauer:  
The ECho Event Delivery System. *A users guide for ECho,  
updated July 2002.*

## Part VI

It is time to ask questions now.