Asynchronous Notification in Condor
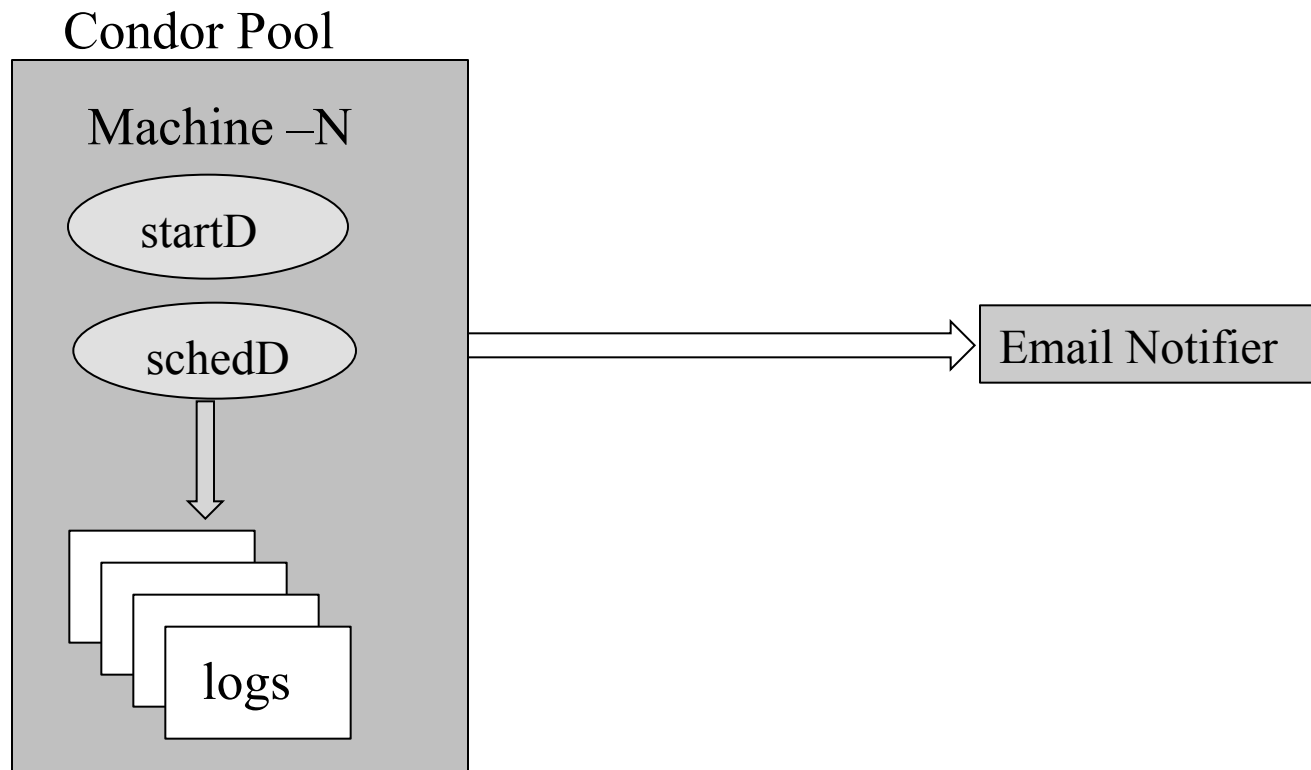# By
Vidhya Murali
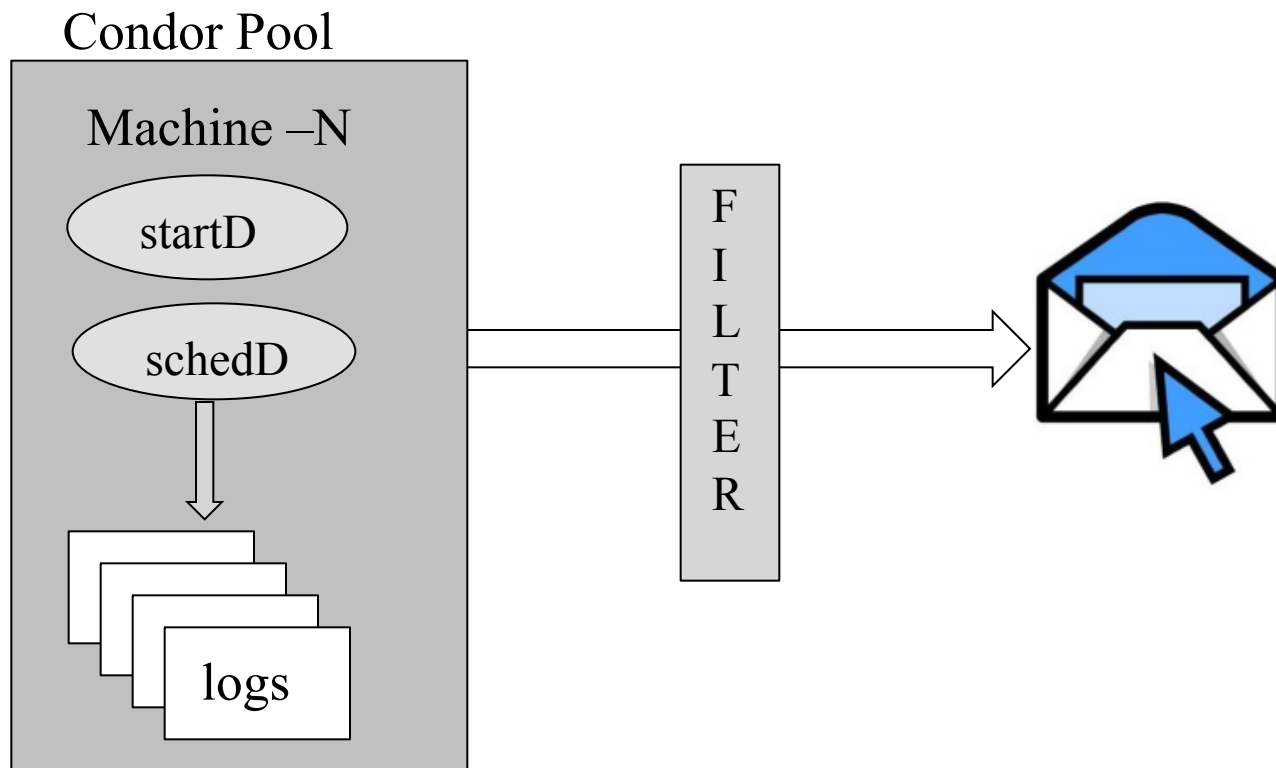
Condor Project
Computer Sciences Department
University of Wisconsin-Madison

CONDOR
high throughput computing

THE UNIVERSITY of WISCONSIN
MADISON

Existing Condor Notification System

Condor Pool

Machine –N

startD

schedD

logs

Email Notifier

Goal -1

Condor Pool

Machine –N

startD

schedD

logs

F
I
L
T
E
R

CONDOR
high throughput computing

THE UNIVERSITY of WISCONSIN
MADISON

Goal -2

Condor Pool

Machine –N

startD

schedD

logs

INTERMEDIATE

CONDOR
high throughput computing

THE UNIVERSITY of WISCONSIN MADISON
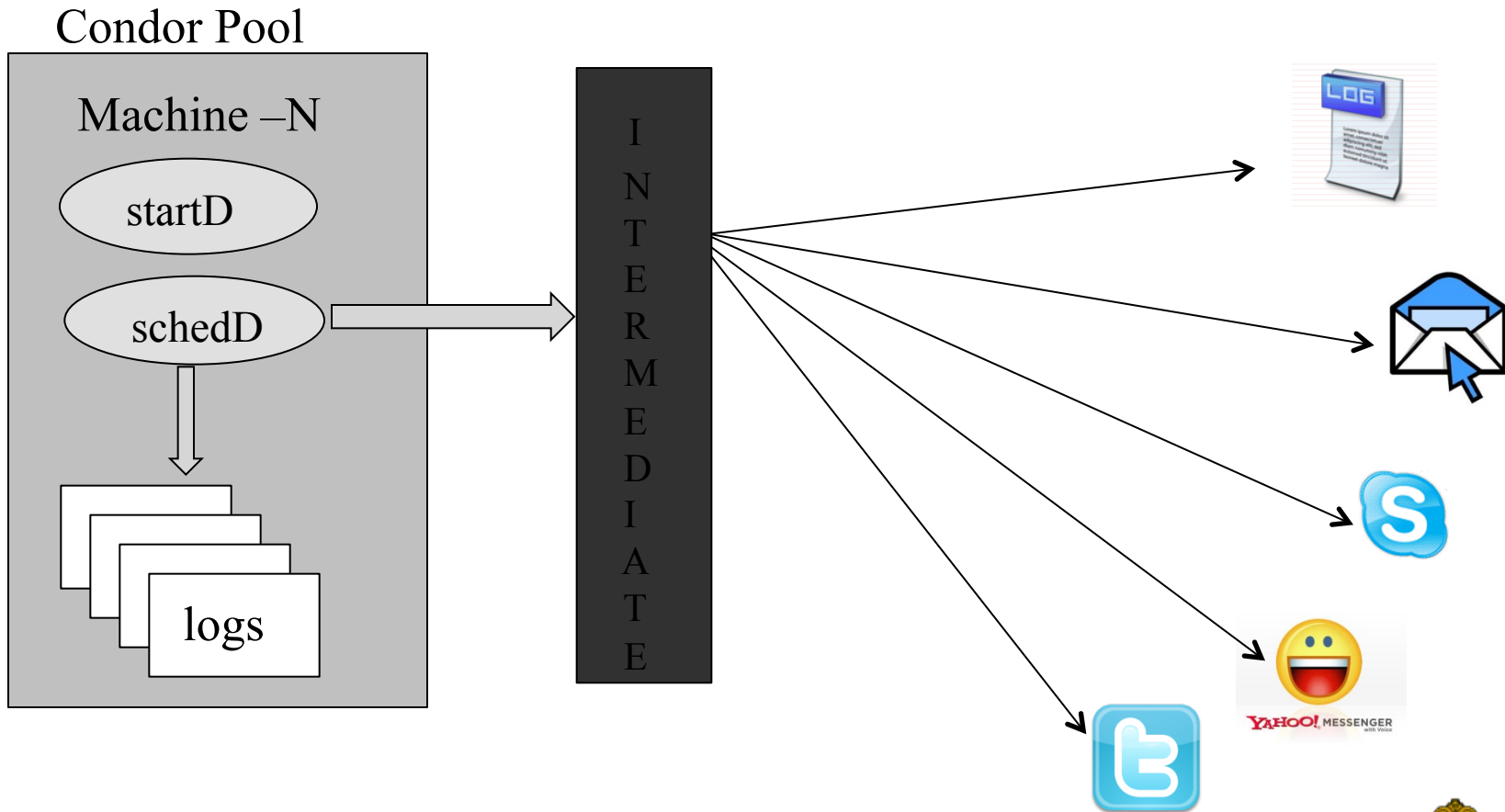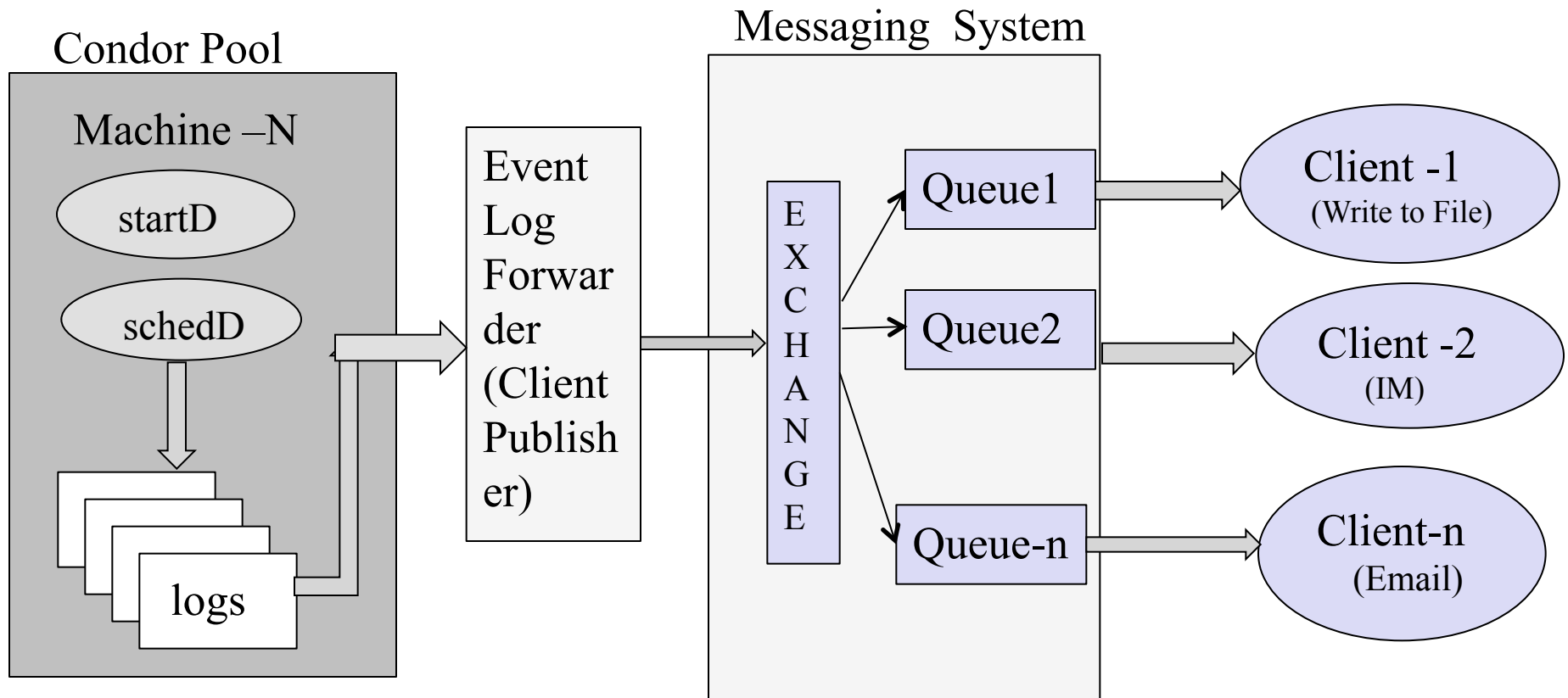
Goals

- Asynchronously transfer condor's event logs(messages) to external listener client(s)
- Event Log Format Transformation
- Event Log Filtering

> Without affecting current schedD setup!

Condor-Messaging System Architecture

Condor Pool

Messaging System

Machine –N

startD

schedD

logs

Event Log Forwarder (Client Publisher)

E X C H A N G E

Queue1

Queue2

Queue-n

Client -1
(Write to File)

Client -2
(IM)

Client-n
(Email)

CONDOR
high throughput computing

www.cs.wisc.edu/Condor

THE UNIVERSITY of
WISCONSIN
MADISON

Outline

> MOM

> Introduction to AMQP

> Integration of Qpid with Condor

> Applications – Skype Integration

> Conclusion

Message-oriented middleware (MOM)

> Infrastructure focused on sending and receiving messages that allows application modules to be distributed over heterogeneous platforms

> Features:
  - Asynchronous Delivery
  - Routing messages (broadcast, multicast possible)
  - Message transformation /mapping

> Specifications eg: XMPP, AMQP

Advanced Message Queuing Protocol (AMQP)

> Open standard application layer protocol for MOM

> Key Features:
  - Message orientation
  - Queuing
  - Routing
 - Reliability
  - Security

AMQP Model
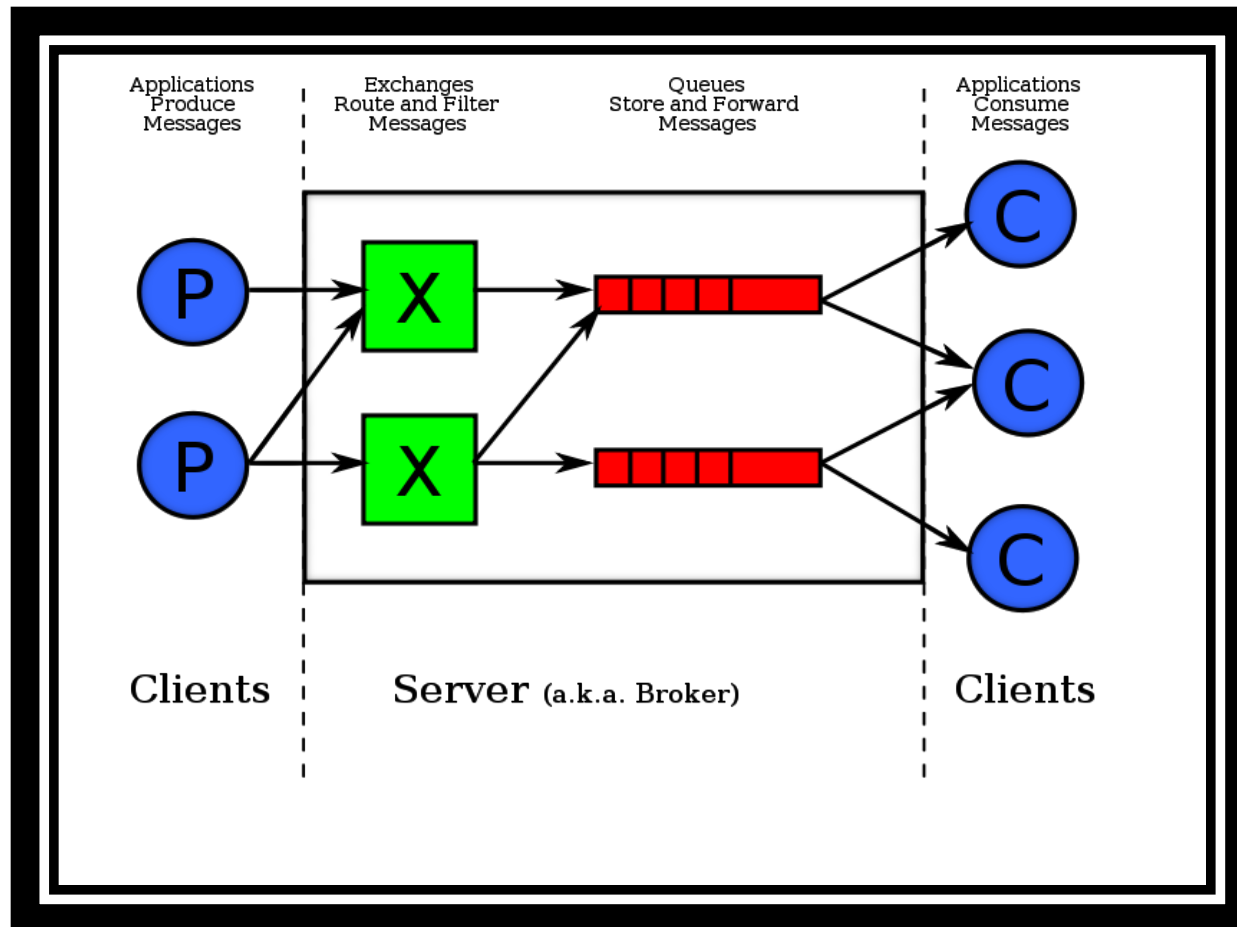
AMQP Entities

> **Exchanges** : entities to which messages are sent.
- Belongs to component Broker.
- Config : **DURABLE**

> **Queues** : entities which receive messages
- Config : **DURABLE,EXCLUSIVE**

> **Binding** : ties one queue to an exchange
- Config : **Conditional on routing key**

AMQP contd...

> Typical messaging patterns

- *Request-Response* : messages are sent to or from specific recipients

- *Publish-Subscribe :* information is distributed to a set of recipients according to various subscription criteria

- *Round-Robin* : tasks are distributed fairly among a set of recipients.

Durable (persistent) Pub/Sub Delivery
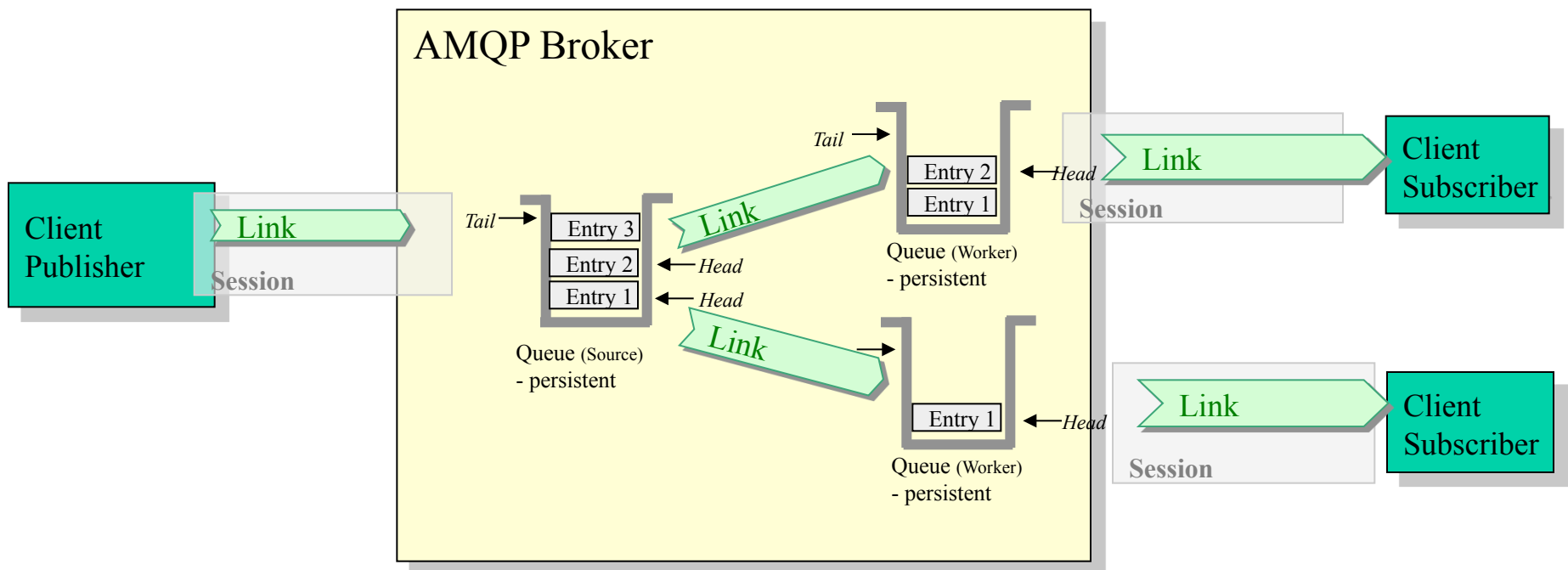
Integration of Qpid with Condor

Apache Qpid is an open source implementation of AMQP.
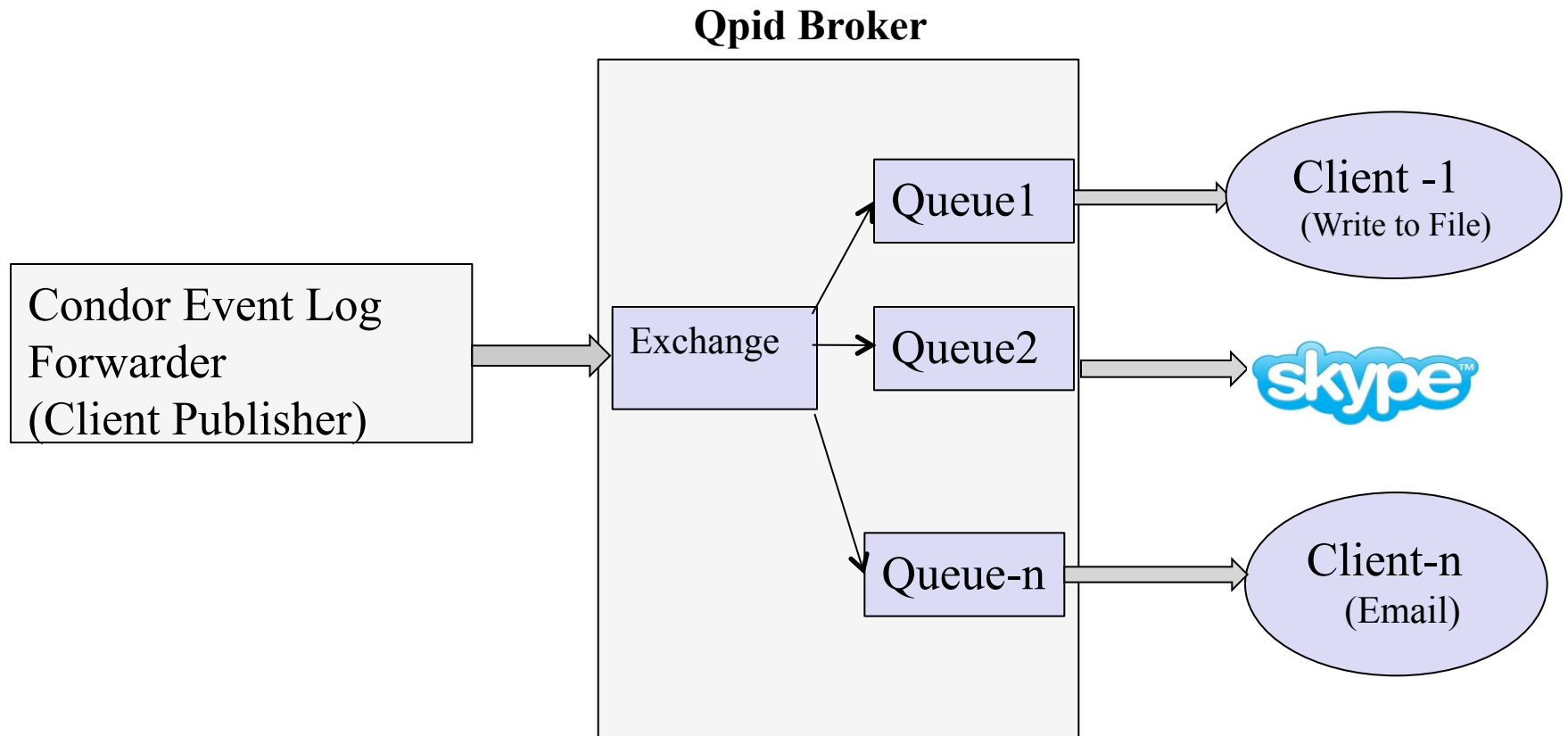
- Using C++ library

Condor_Qpid Daemon: runs the Qpid broker, initializes the queues ,publishes classAd

Condor_jobs:

CLIENT PUBLISHER: Condor_EventLog_Forwarder

CLIENT SUBSCRIBER : Any listener that reads messages writes to file,send emails, IMs

Condor-Qpid Integration Architecture

Qpid Broker

Condor Event Log
Forwarder
(Client Publisher)

Exchange

Queue1 → Client -1 (Write to File)

Queue2 → skype

Queue-n → Client-n (Email)

CONDOR
high throughput computing

www.cs.wisc.edu/Condor

THE UNIVERSITY of WISCONSIN MADISON

Skype for Qpid Client

> Configurations for sending messages:
> - 1 to 1
> - 1 to selected ids in the contact list
> - 1 to all in the contact list

Conclusion

> Part of condor add-ons

> Event log message filtering at client listener level – Feedback mode

> Extension to integrate with other clients like Twitter, Facebook

CONDOR
high throughput computing

www.cs.wisc.edu/Condor

THE UNIVERSITY
of
WISCONSIN
MADISON

Questions?

Thank You!