IBM T. J. Watson Research Center

# Challenges and Experience in Prototyping a Multi-Modal Stream Analytic and Monitoring Application on System S

Kun-Lung Wu, Philip S. Yu, Bugra Gedik,
Kirsten W. Hildrum, Charu C. Aggarwal,
Eric Bouillet, Wei Fan, David A. George,
Xiaohui Gu, Gang Luo, and Haixun Wang

VLDB 2007, Vienna, Austria

09/26/2007

# Outline

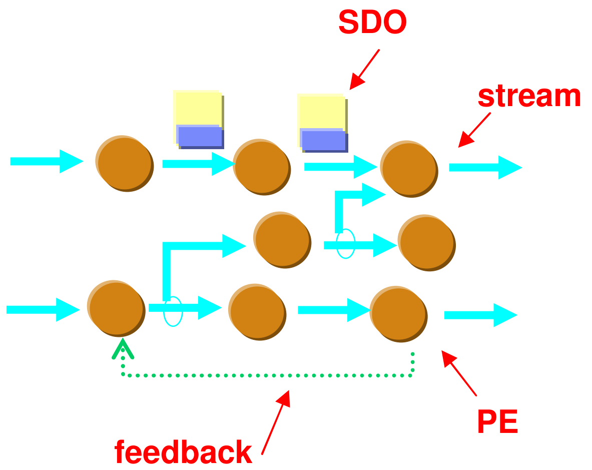
- **Introduction**
  - System S
  - DAC: A Disaster Assistance Claim Monitoring Application
- **Challenges & Experience in Prototyping DAC**
  - Workload generation
  - Design and Implementation
  - Deployment
- **A Demo video**
- **Lessons learned**

# Overview of System S

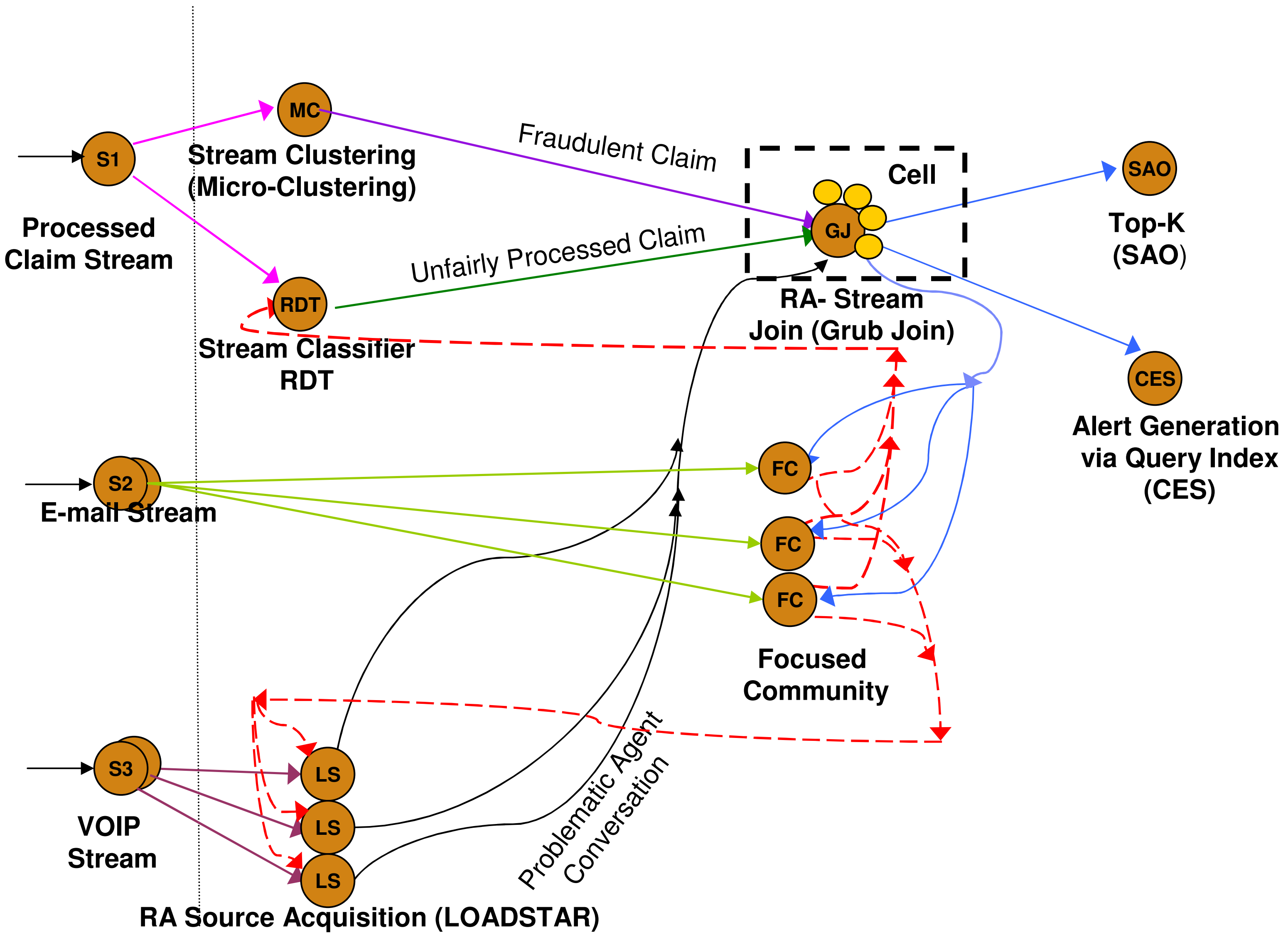
- **System S is a high-performance, distributed, computing platform designed to host stream-oriented applications**
  - High ingestion rates with continuously adaptive processing
  - Multiple programming models
    - Inquiry service, declarative (StreamSQL-like), PE APIs
  - Advanced and evolving feature sets
  - Scales from very small to large to very large hardware configurations
    - From single machines to a cluster of 200 or so blade nodes, each with 2-4 cores

# Overview of System S (continued)

- **Stream programming model**
  - "Branching pipeline" computational model
  - Stream, PE, Stream Data Object (SDO) and Processing Graph
    - Job configuration, flow specifications and PE templates
- **Key components of System S**
  - Dataflow graph manager (DGM)
  - Data fabric (DF)
  - Resource manager (RM)/Scheduler (SODA)
  - PE execution container (PEC)

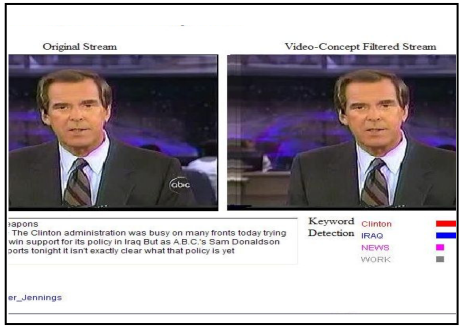


# An example PE processing graph



# Example Stream Applications on System S

### Dense Info Grinding



Original Stream

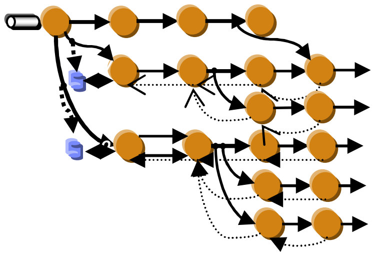
Video-Concept Filtered Stream

apons
The Clinton administration was busy on many fronts today trying win support for its policy in Iraq But as A.B.C.'s Sam Donaldson orts tonight it isn't exactly clear what that policy is yet

Keyword Detection: Clinton, IRAQ, NEWS, WORK

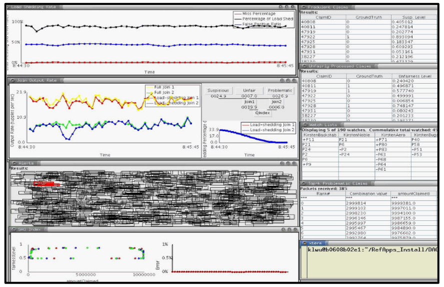
er_Jennings

-Broadcast News Analysis
-Multi-modal (audio, video, txt)
-Just-good-enough analytics

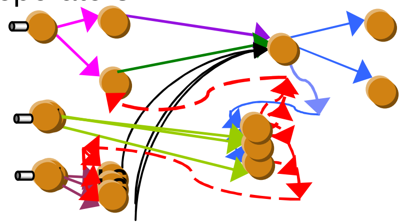


-20 PEs (10 unique)
-4 processing nodes
-20 1-Mbps MPEG transport
-128 hours audio/video
-3 jobs

### Disaster Assistance Claims

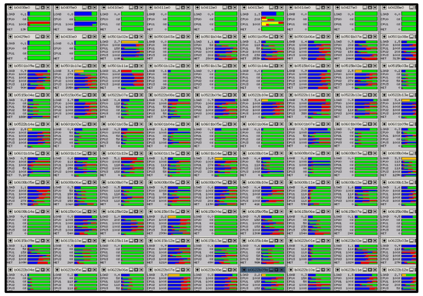


-Real-time processed disaster assistance claim review
-Claim documents, email, VoIP
-Stream mining and relational operators

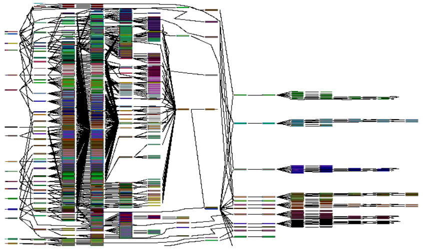


-51 PEs (27 unique)
-35 processing nodes
-3 data sources
-60Mbps raw data input

### Large Scale

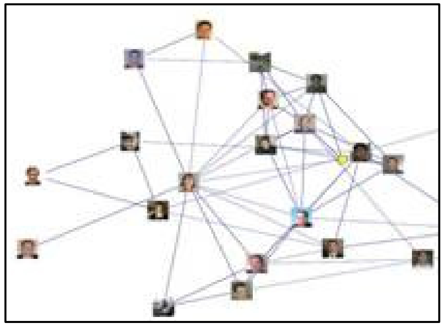


-Multi-application Workload
-Network packet analysis
-Stress runtime environment

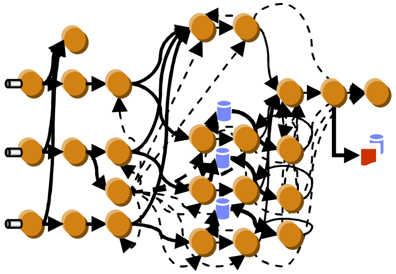


-40-784 PEs
-1-100 processing nodes
-1-21 data sources
-110 jobs

### Who's Talking to Whom



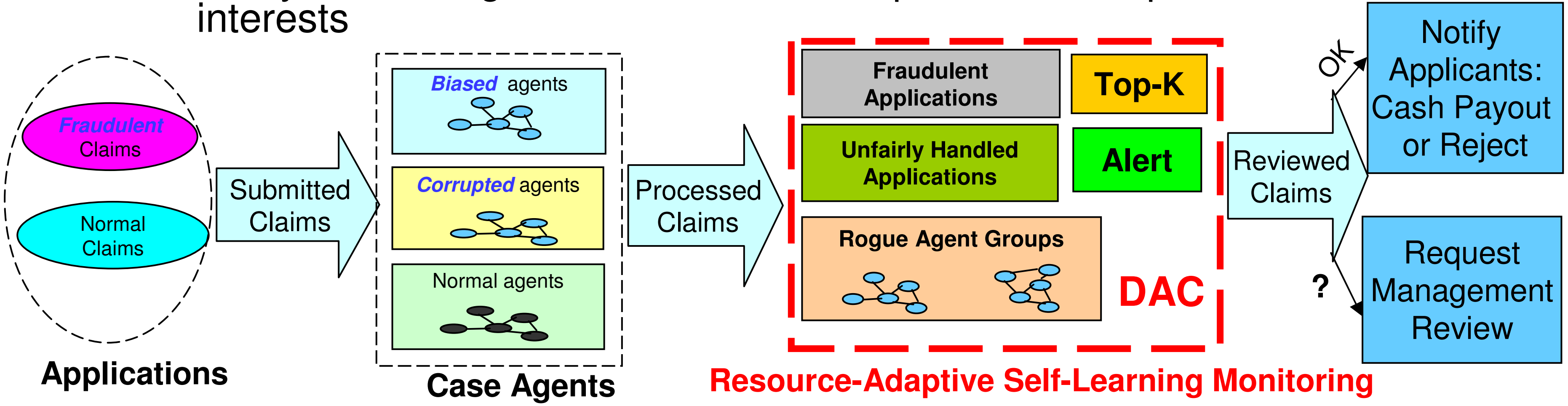
-VoIP Packet Analysis
-Noisy, lossy, correlated data
-Distributed, adaptive analytics



-11-38 PEs (20 unique)
-1-14 processing nodes
-1-3 data sources
-3 jobs
-200 concurrent streams

# Disaster Assistance Claim (DAC) Monitoring

- **In a disaster claim processing center**
  - Claims may be fraudulent or unfairly processed
  - Claim processing agents may be engaged in criminal activities to cheat the agency
- **Goals of DAC monitoring:** ***real-time reviewing processed claims***
  - Identify for management review in real-time the fraudulent or unfairly processed cases before the decisions are conveyed to the applicants
    - Once the money is paid out, it is hard to get it back
  - Identify the problematic processing agents and the potential crime groups
  - Notify the management based on alert profiles and top-k cases of interests



Resource-Adaptive Self-Learning Monitoring

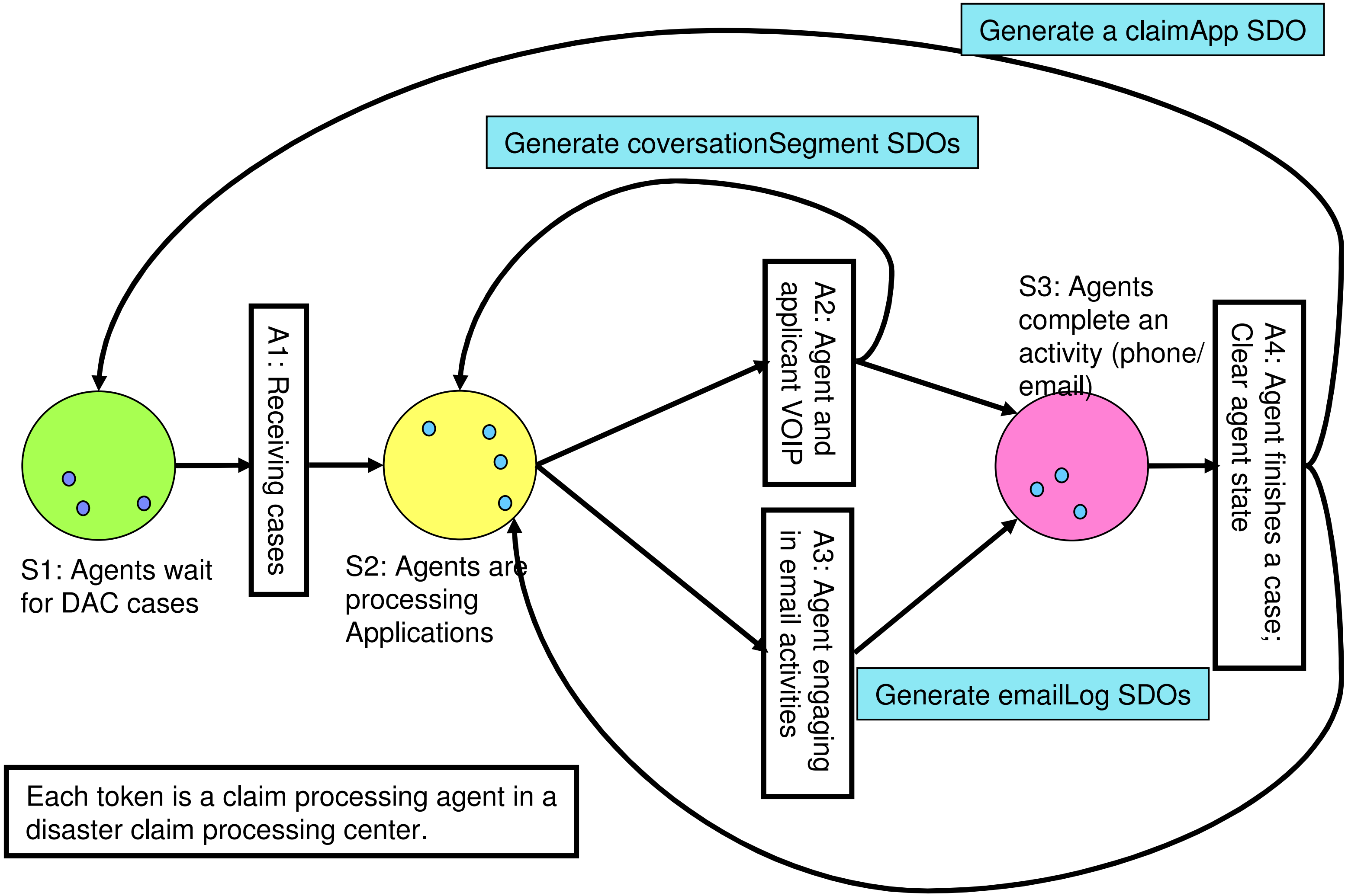
# Challenges

- **Workload generation**
  - How do we model a claim processing center, specify data distributions for individual streams and implement correlated distributions across streams?
- **Design and Implementation**
  - How do we develop individual PEs with stream analytic algorithms to identify fraudulent claims, unfairly processed claims, problematic agents and criminal communities?
  - How do we integrate so many different PEs into a complex processing graph?
- **Deployment**
  - How do deploy PEs so that we could correlate streams with vastly different data rates?

# Assumptions on source streams

- **Multiple modalities, generated through a workload generator**
  - Processed claims: *fraudulent or unfairly treated ?*
  - VOIP packets on the conversations between applicants and processing agents: *agent behavior abnormal?*
  - Agent E-mails: *agent community/crime group?*
  - Video Surveillance data: *agent community/crime group?*
  - News event data: *disaster information (correlating with fraudulent claims)*

# Workload generation



Generate a claimApp SDO

Generate coversationSegment SDOs

S1: Agents wait for DAC cases

A1: Receiving cases

S2: Agents are processing Applications

A2: Agent and applicant VOIP

A3: Agent engaging in email activities

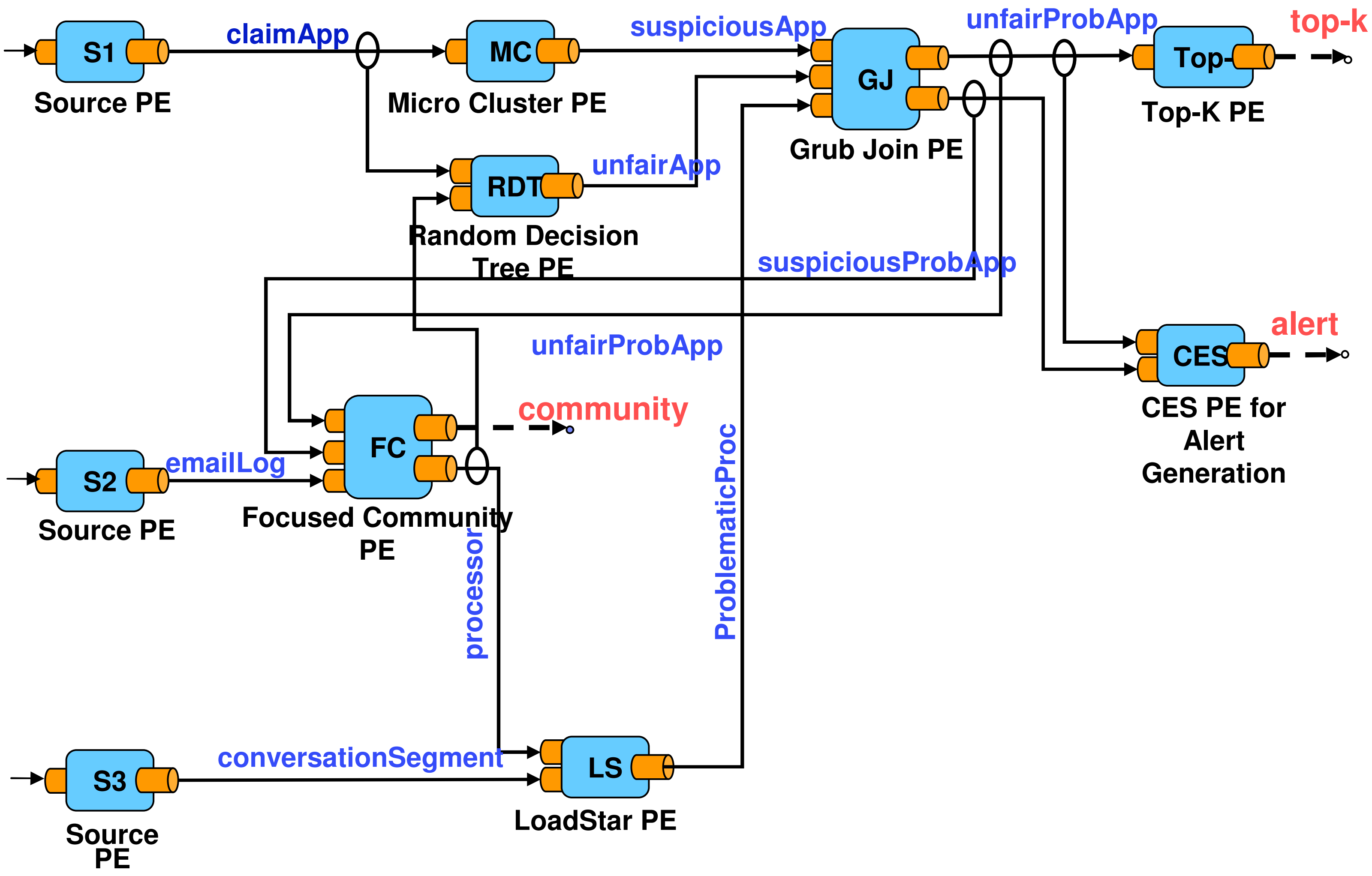
S3: Agents complete an activity (phone/ email)

A4: Agent finishes a case; Clear agent state

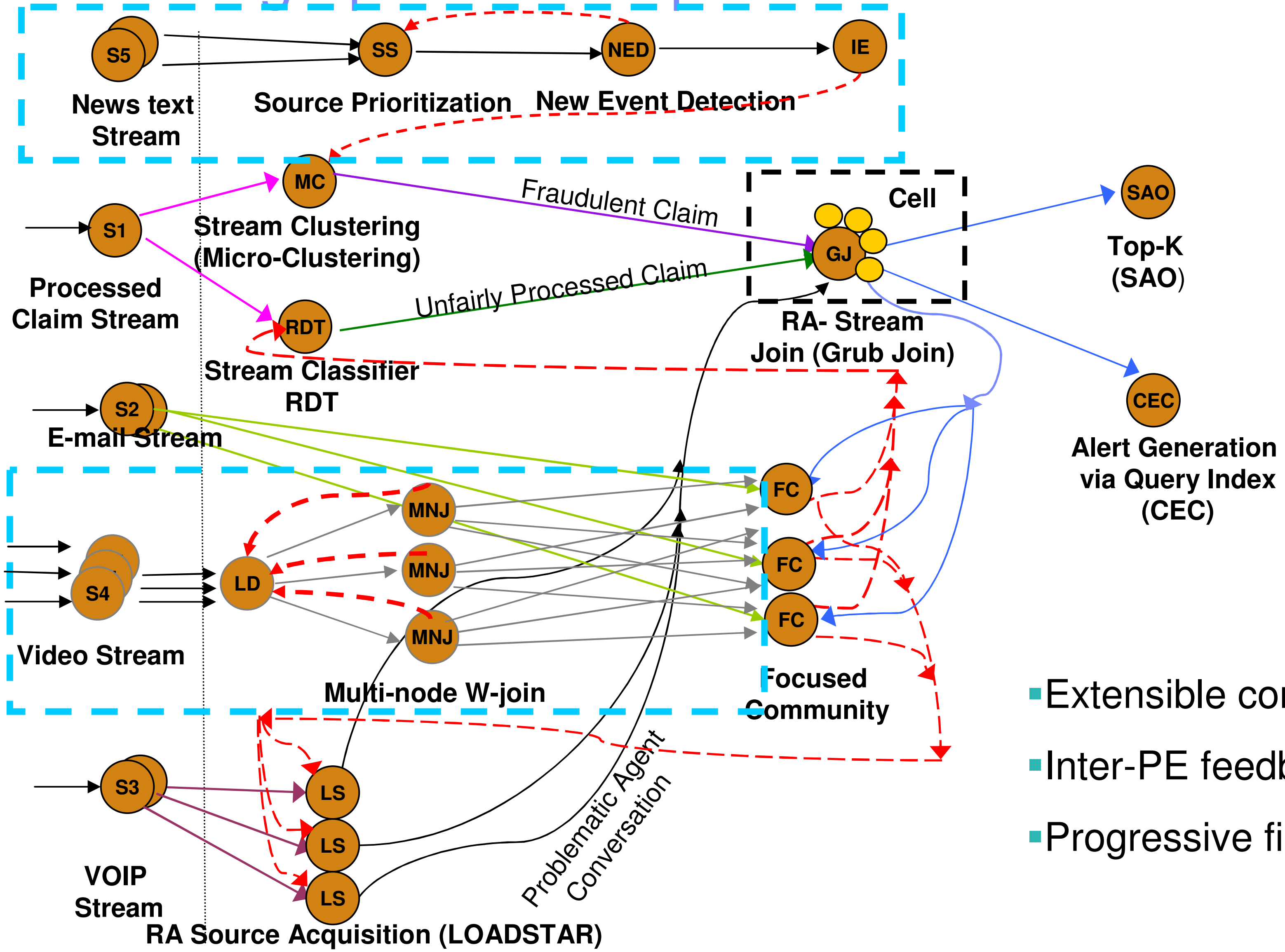
Generate emailLog SDOs

Each token is a claim processing agent in a disaster claim processing center.

# PE flow graph of a simplified DAC



# PE flow graph of a complete DAC



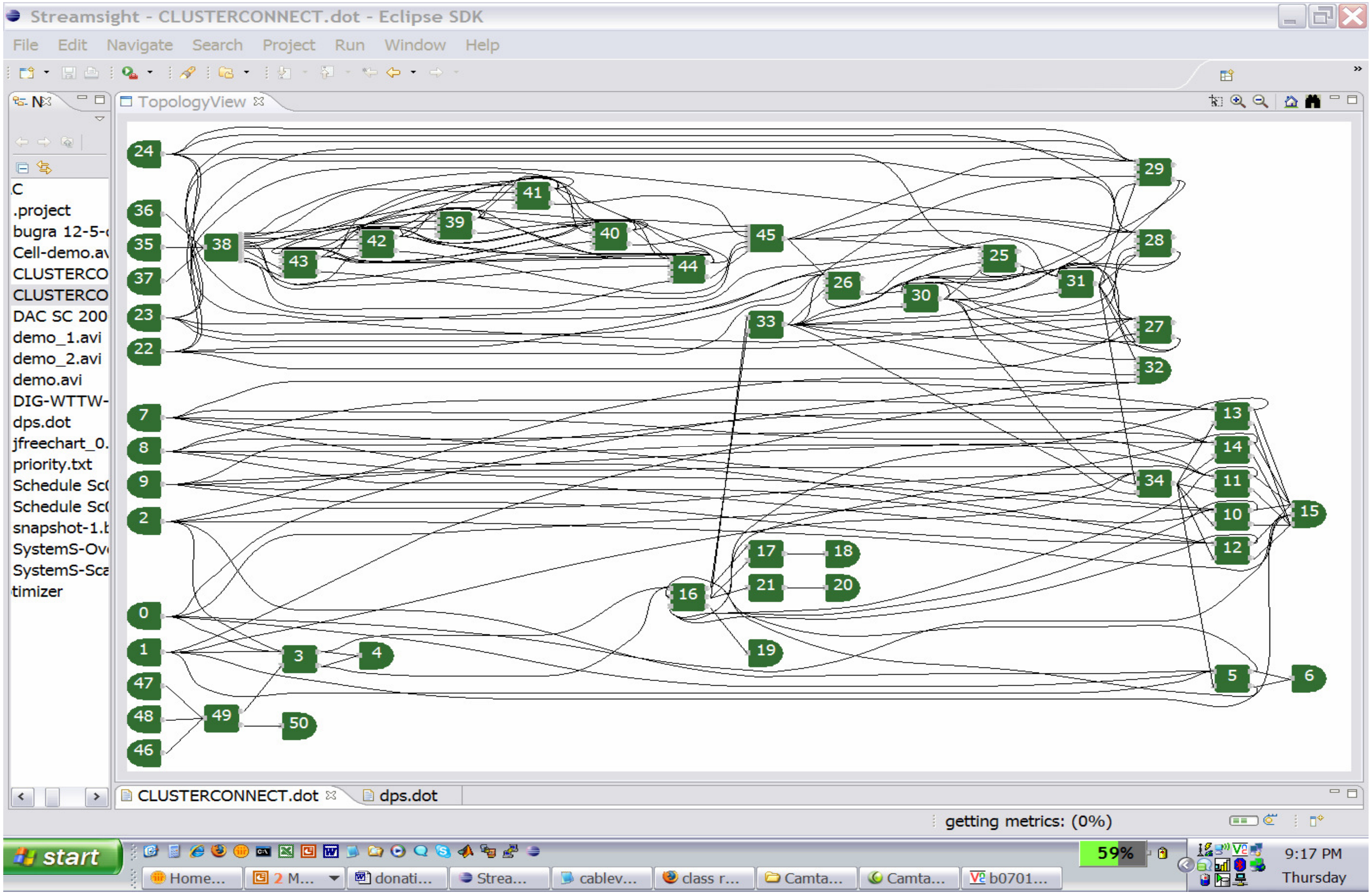
- Extensible composition
- Inter-PE feedback
- Progressive filtering

# Deployment of DAC

- **Mismatch among the stream rates causes mismatch in PE processing load**
  - Processed claim (1) : e-mails (50) : conversation segments (100)
  - MC:FC:LoadStar ~ 1:50:100
- **Parallelism**
  - Multiple FC PEs and LoadStar PEs
  - Flow specifications make it easy to split streams among PEs
- **Resource-adaptive computation**
  - LoadStar PE and GrubJoin PE employ intelligent load shedding

# A DAC Demo

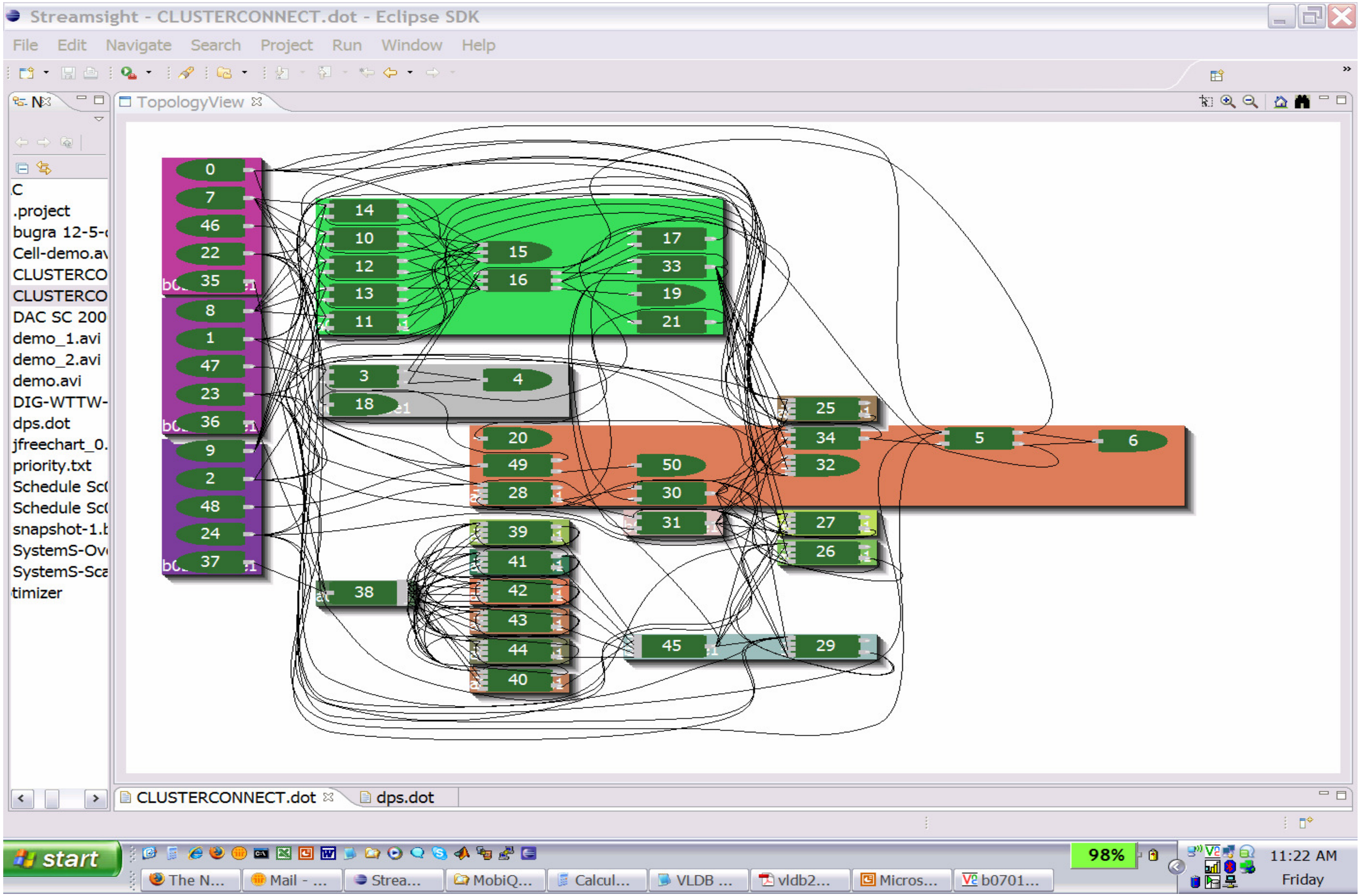
- **Individual stream analytics: each PE showcases different stream technologies**
  - Self-learning stream mining, parallelizable stream algorithm, resource adaptive computation,, etc.
- **Integration of various stream analytics into a comprehensive application**
  - Stream speed & processing load mismatch among PEs
    - ***Multiple stream modalities***: Claim application stream, VOIP stream, e-mail stream, video stream, news stream
  - Synergism & cooperation between PEs
    - ***Inter-PE feedback***
    - ***Progressive filtering***
- **Load spreading & Parallel Processing**
  - Using flowspec, both static as well as dynamic, to spread load among different PEs



Streamsight - CLUSTERCONNECT.dot - Eclipse SDK

C
.project
bugra 12-5-
Cell-demo.a
CLUSTERCO
CLUSTERCO
DAC SC 200
demo_1.avi
demo_2.avi
demo.avi
DIG-WTTW-
dps.dot
jfreechart_0.
priority.txt
Schedule Sc
Schedule Sc
snapshot-1.
SystemS-Ov
SystemS-Sc
timizer

getting metrics: (0%)


Streamsight - CLUSTERCONNECT.dot - Eclipse SDK
File Edit Navigate Search Project Run Window Help
TopologyView
C
.project
bugra 12-5-
Cell-demo.av
CLUSTERCO
CLUSTERCO
DAC SC 200
demo_1.avi
demo_2.avi
demo.avi
DIG-WTTW-
dps.dot
jfreechart_0.
priority.txt
Schedule Sc
Schedule Sc
snapshot-1.
SystemS-Ov
SystemS-Sca
timizer
0
7
46
22
35
8
1
47
23
36
9
2
48
24
37
14
10
12
13
11
15
16
17
33
19
21
3
4
18
25
20
49
28
34
32
5
6
50
30
31
27
26
39
41
42
43
44
40
38
45
29
CLUSTERCONNECT.dot
dps.dot
start
The N...
Mail - ...
Strea...
MobiQ...
Calcul...
VLDB ...
vldb2...
Micros...
b0701...
98%
11:22 AM
Friday

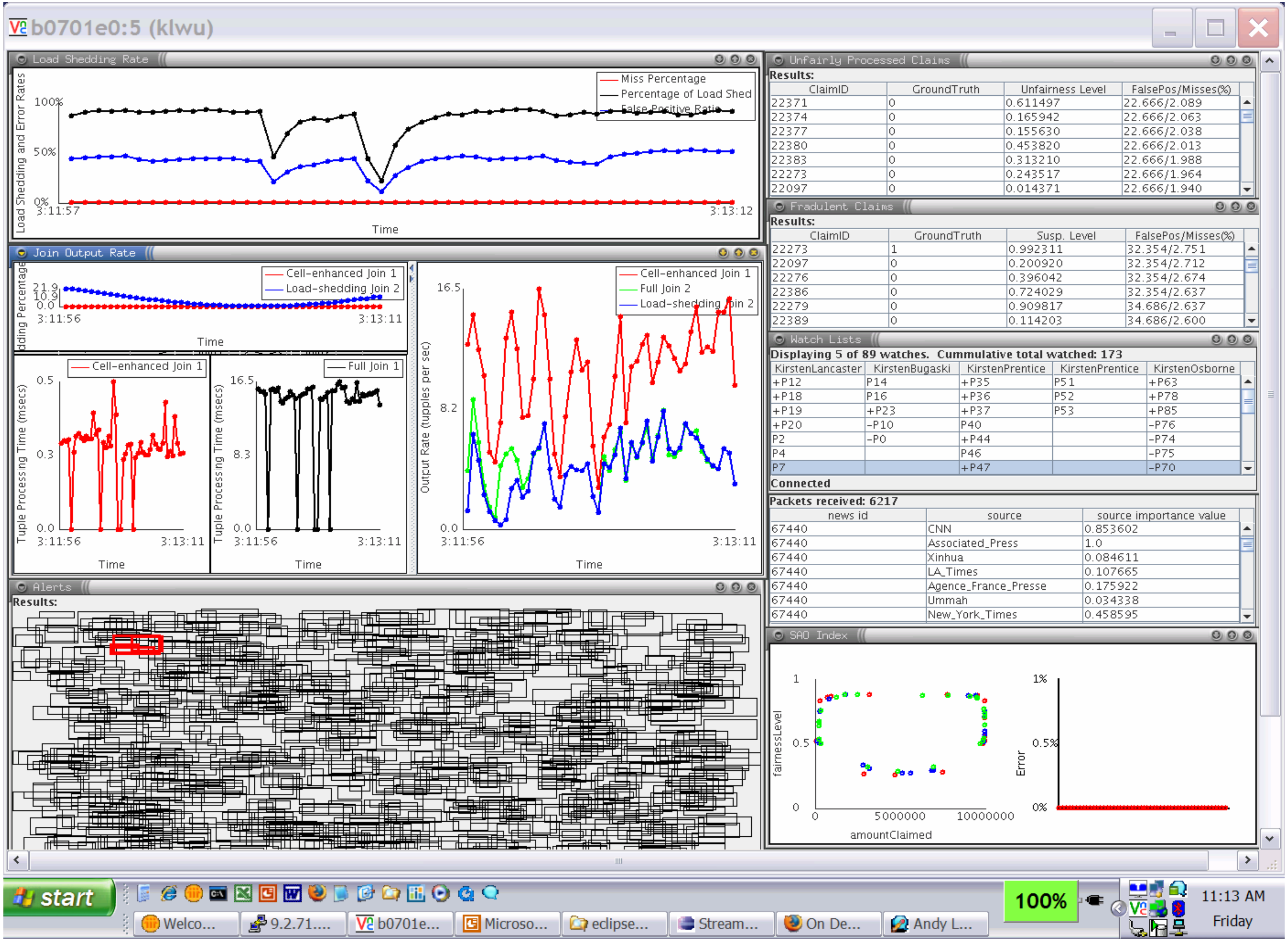
# A DAC demo video

# Lessons Learned

- **Successfully morphed traditional analytic algorithms into stream-based ones**
  - Micro-cluster PE and Random Decision Tree PE
- **Created resource-adaptive stream algorithms**
  - GrubJoin PE and LoadStar PE, New Event Detection PE
- **Demonstrated the extensible nature of the system, allowing incremental application design, from simple to truly complicated stream applications**
- **Effectively handled and correlated five streams of different modalities with vastly different rates and processing requirement**
- **Programming PEs was made easier without knowing too much about the SPC implementation details**

# Thank you!

- **Questions and comments?**



Unfairly Processed Claims

Results:

| ClaimID | GroundTruth | Unfairness Level | FalsePos/Misses(%) |
|---|---|---|---|
| 22371 | 0 | 0.611497 | 22.666/2.089 |
| 22374 | 0 | 0.165942 | 22.666/2.063 |
| 22377 | 0 | 0.155630 | 22.666/2.038 |
| 22380 | 0 | 0.453820 | 22.666/2.013 |
| 22383 | 0 | 0.313210 | 22.666/1.988 |
| 22273 | 0 | 0.243517 | 22.666/1.964 |
| 22097 | 0 | 0.014371 | 22.666/1.940 |

Fradulent Claims

Results:

| ClaimID | GroundTruth | Susp. Level | FalsePos/Misses(%) |
|---|---|---|---|
| 22273 | 1 | 0.992311 | 32.354/2.751 |
| 22097 | 0 | 0.200920 | 32.354/2.712 |
| 22276 | 0 | 0.396042 | 32.354/2.674 |
| 22386 | 0 | 0.724029 | 32.354/2.637 |
| 22279 | 0 | 0.909817 | 34.686/2.637 |
| 22389 | 0 | 0.114203 | 34.686/2.600 |

Watch Lists

**Displaying 5 of 89 watches. Cummulative total watched: 173**

| KirstenLancaster | KirstenBugaski | KirstenPrentice | KirstenPrentice | KirstenOsborne |
|---|---|---|---|---|
| +P12 | P14 | +P35 | P51 | +P63 |
| +P18 | P16 | +P36 | P52 | +P78 |
| +P19 | +P23 | +P37 | P53 | +P85 |
| +P20 | -P10 | P40 |  | -P76 |
| P2 | -P0 | +P44 |  | -P74 |
| P4 |  | P46 |  | -P75 |
| P7 |  | +P47 |  | -P70 |

**Connected**

**Packets received: 6217**

| news id | source | source importance value |
|---|---|---|
| 67440 | CNN | 0.853602 |
| 67440 | Associated_Press | 1.0 |
| 67440 | Xinhua | 0.084611 |
| 67440 | LA_Times | 0.107665 |
| 67440 | Agence_France_Presse | 0.175922 |
| 67440 | Ummah | 0.034338 |
| 67440 | New_York_Times | 0.458595 |