A Shared Global Event Propagation System to Enable Next-Generation Distributed Services

Rob Knauerhase

Research Engineer
Intel Labs, Intel Corporation
knauer{at}jf.intel.com

With team: Mic Bowman, Paul Brett, Robert Adams, Jeff Sedayao, Aroon Nataraj, Michael Spindel

Infrastructure for Planetary-Scale Distributed Services

- How to accommodate dynamic / unreliable environment
  - Compute resources (load, availability)
  - Network resources (connections, bandwidth, latency, routes, load, address spaces)
- How to build infrastructure that supports new services
  - Can services “rendezvous” without ties to network address, allowing redundancy, migration, etc.? (location independence)
  - Do producers and consumers have to know specific details of each other’s existence? (service independence)
  - Can service interfaces be self-describing? (protocol independence)
  - Is asynchronous best-effort delivery sufficient and satisfactory for new types of distributed services? (timing independence)

“**Loose binding**” methodology == dynamic runtime discovery and composition of independent interoperating components.

PsEPR

- Planetary-Scale Event Propagation and Routing
  - Infrastructure supporting research into loosely-bound construction methods for distributed services
    - Event-based messaging abstraction (no guarantees)
    - Publish/subscribe semantics
    - Designed to support adaptability, scalability, and reusability of services and service components
- ... a loosely-bound service itself
  - Registry component maintains subscriptions and event delivery to local clients
  - Router component distributes messages across an overlay network of registries

PsEPR details

- PsEPR Events
  - Unique “chunks” of information, in XML
  - Published asynchronously by services for consumption by anyone
- Addressing
  - Channel – arbitrary namespace, with hierarchical grouping
    - Participants don’t need to know about subscribers – subscriptions and routing are transparent to the services and clients
  - Service – authenticated participant (producer or consumer of events)
  - Instance – one particular runtime instantiation of a service

PsEPR implementation

- PsEPR servers are built atop Jabber™ IM servers
  - Object-level addressing: map service/instance to IM name
    - Endpoints independent of, e.g., socket
  - Authentication for “free”
  - Existing techniques for firewall/NAT traversal
- Communication among IM servers in overlay network
  - Ability to drop messages or restrict delivery/change routes as consumers change
  - Also allows flexibility to explore alternate route algorithms
- Client libraries abstract away IM characteristics
  - Transparently find and connect to (or re-connect to) “best” PsEPR server

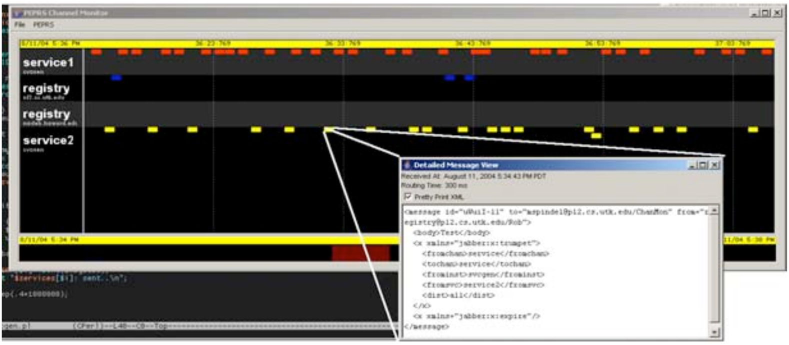
PsEPR-based Services

- Monitors and sensors
  - Output Trumpet data on channels
    - ~500 monitors generating ~20,000 events per day
- Tuple store (PLDB v2)
  - Listens for and stores data events on channels
  - Listens for query events and emits satisfying events
- Management supervisor
  - Monitors health and distribution of tuple stores
    - Starts and stops instances according to need and system load
    - Adds/removes redundancy as needed for quality of service

Each of these is a loosely-bound service using PsEPR events. PsEPR assists in adaptation, redundancy/reliability, scalability, and ease of development.

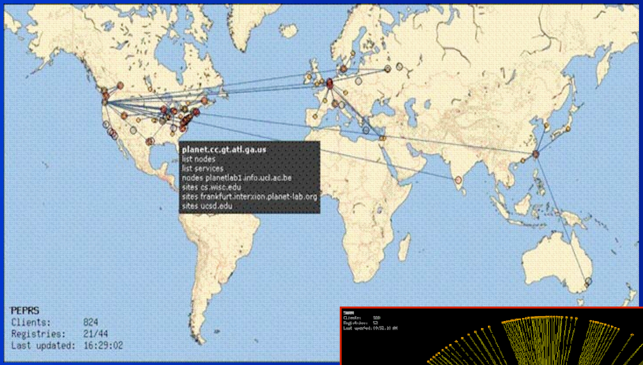
PsEPR Developer Services

- PsEPR debugger
  - Displays timescale view of events on a channel
- PsEPR visualization
  - Displays clients, servers, connections, and traffic

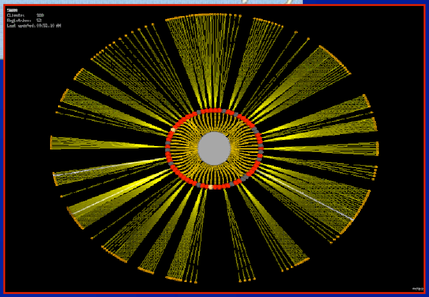


Received At: August 11, 2004 5:34:43 PM PDT
Routing Time: 300 ms
☑ Pretty Print XML

```
<message id="uWuiI-11" to="mspindel@pl2.cs.utk.edu/ChanMon" from="r
egistry@pl2.cs.utk.edu/Rob">
  <body>Test</body>
  <x xmlns="jabber:x:trumpet">
    <fromchan>service</fromchan>
    <tochan>service</tochan>
    <frominst>svcgen</frominst>
    <fromsvc>service2</fromsvc>
    <dist>all</dist>
  </x>
  <x xmlns="jabber:x:expire"/>
</message>
```


planet.cc.gt.atl.ga.us
list nodes
list services
nodes planetlab1.info.ucl.ac.be
sites cs.wisc.edu
sites frankfurt.interxion.planet-lab.org
sites ucsd.edu

```
PEPRS
Clients:        824
Registries:     21/44
Last updated:   16:29:02
```



These services work with any PsEPR service, without need of modification to the service or the infrastructure.

First-generation tools for PsEPR development...

Open research questions

- Tweaks on overlay routing
  - Combinations of overlay topologies (bus, star, tree, DHT) and dynamic reconfiguration between them
- Architecture and operation of loosely-bound systems
  - Toolkit for building and operating extremely large-scale decentralized services on PlanetLab
- Dynamic adaptation of services to environment
  - Within PsEPR (change overlay techniques with load)
  - PsEPR providing information to services
    - Hints for redundancy, migration, etc. based on environment
      - Number of subscribers, location of subscribers
      - Network considerations (load, latency, cost, ...)

Summary

- Infrastructure can help in building scalable, reliable systems in a dynamic distributed environment
  - Independence of location, service, timing, and protocol
  - Dynamic composition enables *adaptive* services
- PsEPR is a communication framework that facilitates construction of loosely-bound service components
  - Running on PlanetLab
  - Several examples of distributed, scalable, adaptable, transparently-interoperating services