



What do we want out of a Network?

Dr. Eric A. Brewer
Professor, UC Berkeley
Co-Founder & Chief Scientist, Inktomi

HotInterconnects Keynote, August 2002



“Distributed Systems” don’t work...

- ◆ **There exist working DS:**
 - Simple protocols: DNS, WWW, Napster
 - Inktomi search, Content Delivery Networks
- ◆ **But these are not classic DS:**
 - Not distributed objects, no modularity
 - No RPC
 - Complex ones are single owner (except phones)
- ◆ **System goals & network not well aligned**

HotInterconnects Keynote, August 2002



Our Perspective

- ◆ **Parallel Computing**
- ◆ **Cluster Computing**
- ◆ **Inktomi builds two distributed systems:**
 - Global Search Engines
 - Distributed Web Caches
- ◆ **Not a hardware guy**



HotInterconnects Keynote, August 2002



Basic Claim

We are generally unclear about what properties we need out of a network.

- ◆ **Consequence: Some desired higher-level properties are not really obtainable**
- ◆ **Many open problems in the design of whole stacks**

HotInterconnects Keynote, August 2002

Agenda



- ◆ Simple Properties
- ◆ Complex Properties
- ◆ Overlay Networks

HotInterconnects Keynote, August 2002

Example: Active Messages 2



- ◆ Interface: reliable delivery or it returns the message to you (as an exception)
- ◆ Advantages:
 - Sender need not keep a copy for retransmission
 - Can exploit reliable network when it exists (layer keeps a copy if not)

HotInterconnects Keynote, August 2002

Network Basics



Reliable
Ordered
Flow Control

| Reliable | Ordered | Flow Control | Protocol | Use Case |
|----------|---------|--------------|----------|-----------------------------|
| X | X | X | TCP | Reliable WAN |
| X | X | | | Parallel computers, busses |
| X | | X | | Large data transfer |
| X | | | RDP | CM-5 |
| X | X | | | Stock quotes, cache updates |
| X | | | | LAN updates, source routing |
| | | X | | Streaming, Digital Fountain |
| | | | UDP | Do it yourself |

HotInterconnects Keynote, August 2002

Example: Streaming



- ◆ Needs flow control, but not reliability or ordering
- ◆ Option 1: TCP
 - Provides flow control
 - Delays delivery to provide order => misses deadlines
- ◆ Option 2: UDP
 - No flow control
 - Can't really build it on top: user-level feedback loop is too slow
 - Can exploit data as it arrives, less interruptions
- ◆ Right answer:
 - Flow-control only session
 - Plus, joint source-channel coding for effective bitrate!
 - Control which bits are not received
 - Requires Application Level Framing (ADUs)

HotInterconnects Keynote, August 2002

Property: Multiplexing clients/flows?



- ◆ **Does the server have to:**
 - Deal with high concurrency?
 - Say “no” sometimes (graceful degradation)
 - Treat clients equally (fairness)
 - Bill for resources (and have audit trail)
 - Isolate clients performance, data,
- ◆ **Large servers have > 10,000 flows**

HotInterconnects Keynote, August 2002

Property: Partial Failure



- ◆ **Can the two sides fail independently?**
- ◆ **Can't be transparent (like RPC) !!**
 - New exceptions must be handled (other side gone)
- ◆ **What does an “ack” mean?**
 - Idempotent calls?
 - Use Transaction Ids (to solve replay problem)
 - Is an ack persistent across node failure?
- ◆ **Reclaim local resources**
 - e.g. kernels leak sockets over time => reboot
- ◆ **Use Level 4/7 switches to hide failures?**
 - Only if no session state... (unlikely)

HotInterconnects Keynote, August 2002

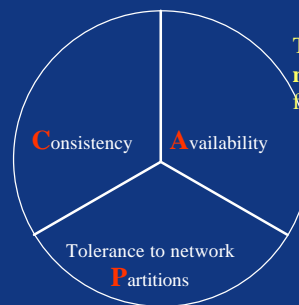
Agenda



- ◆ **Simple Properties**
- ◆ **Complex Properties**
- ◆ **Overlay Networks**

HotInterconnects Keynote, August 2002

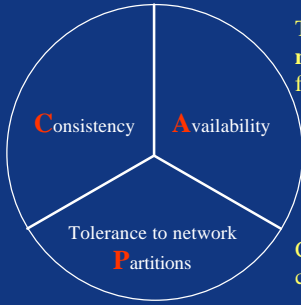
The CAP Theorem



Theorem: You can have **at most two** of these invariants for any shared-data system

HotInterconnects Keynote, August 2002

The CAP Theorem

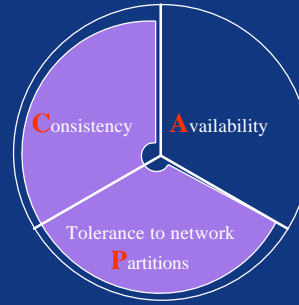


Theorem: You can have at **most two** of these invariants for any shared-data system

Corollary: Partitions => must choose A or C

HotInterconnects Keynote, August 2002

Forfeit Availability



Examples

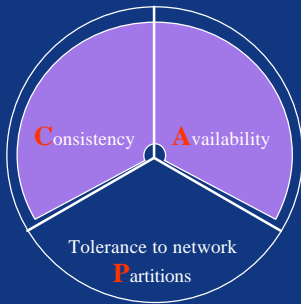
- ◆ Distributed databases
- ◆ Distributed locking
- ◆ Majority protocols

Traits

- ◆ Pessimistic locking
- ◆ Make minority partitions unavailable

HotInterconnects Keynote, August 2002

Forfeit Partitions



Examples

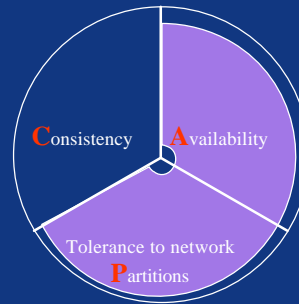
- ◆ Single-site databases
- ◆ Cluster databases
- ◆ LDAP
- ◆ Fiefdoms

Traits

- ◆ 2-phase commit
- ◆ cache validation protocols
- ◆ "SAN style"

HotInterconnects Keynote, August 2002

Forfeit Consistency



Examples

- ◆ Coda
- ◆ Web caching
- ◆ DNS

Traits

- ◆ expirations/leases
- ◆ conflict resolution
- ◆ Optimistic
- ◆ "Internet style"

HotInterconnects Keynote, August 2002

Property: Trust the other side?



inktom |

- ◆ **What if we don't trust the other side?**
 - Or partial trust (legal contract), or malicious?
 - Have to check args, no pointer passing
- ◆ **Limited release of information (leaks)**
- ◆ **Kernels get this right:**
 - copy/check args
 - use opaque references (e.g. File Descriptors)
- ◆ **Most systems do not get this right...**
 - TCP, Napster, web browsers
- ◆ **Security boundaries tend to be explicit**
 - Holes come from software!

HotInterconnects Keynote, August 2002

Example: protocols vs. APIs



inktom |

- ◆ **Protocols have been more successful than APIs**
- ◆ **Some reasons:**
 - protocols are pass by value
 - protocols designed for partial failure
 - not trying to look like local procedure calls
 - explicit state machine, rather than call/return (this exposes exceptions well)
- ◆ **Protocols still not good at trust, billing, evolution**

HotInterconnects Keynote, August 2002

Property: Boundary evolution?



inktom |

- ◆ **Can the two sides be updated **independently**?**
(NO)
- ◆ **The DLL problem... (updates break other apps)**
 - Boundaries need versions
 - Multiple versions need to co-exist on same machine
 - Names should include version number
- ◆ **Negotiation protocol for upgrade?**
 - Do both sides need to authorize upgrade?
- ◆ **Promises of backward compatibility?**

HotInterconnects Keynote, August 2002

Property: Energy efficiency?



inktom |

- ◆ **Open problem: what is the right network layering for energy?**
- ◆ **Claim: probably not reliable delivery**
- ◆ **Need to exploit two things:**
 - Probabilistic delivery with multiple paths
 - Free multicasting (for RF)
 - Tension between compression and FEC

HotInterconnects Keynote, August 2002

Property: Network Growth?



- ◆ **Limit the maximum network size ahead of time?**
 - SANS say 'yes'
 - Allows overengineering for max size
 - Namespace, bisection bandwidth, "routing" table size, limits on max delay, ...
- ◆ **Without this limit, you need:**
 - Dynamic congestion control
 - Admission control (?) -- in general networks need to be able to say "no" to ensure any property
 - Very large namespace plus routing complexity

HotInterconnects Keynote, August 2002

Property: Software in the Network?



- ◆ **Goal: add capabilities to the network over time with easy deployment**
 - Phone system has some of this (caller ID, call waiting)
 - Proxies: computing/storage nodes in the middle
 - Active Networks: computing on every hop
- ◆ **Many challenges:**
 - Security, stability, federation are the top three
 - But, major new capabilities have been delivered

HotInterconnects Keynote, August 2002

Property: Federation?



- ◆ **Does network have multiple administrative domains?**
- ◆ **Single owner is *MUCH* simplerf**
 - Single set of goals
 - Enough control to deploy changes
- ◆ **Claim: hardest problem in networking is the evolution of a multi-owner network (new functionality)**
 - Current mechanism is IETF with limited results
 - Players rarely have the same goals...

HotInterconnects Keynote, August 2002

Agenda



- ◆ **Simple Properties**
- ◆ **Complex Properties**
- ◆ **Overlay Networks**

HotInterconnects Keynote, August 2002

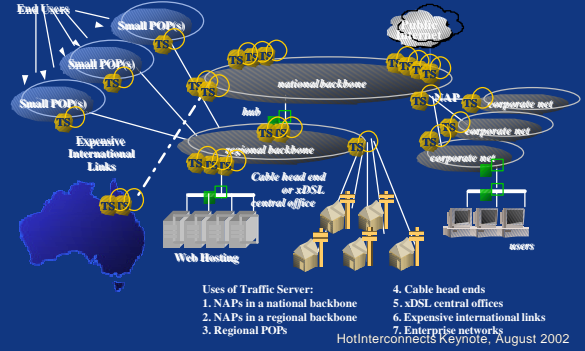
The "Old" Internet



- ◆ Local networks with local names and switches
- ◆ IP creates global namespace and links the local networks
- ◆ Routers connect IP names worldwide, and move packets
- ◆ Big networks share routing tables (BGP)

HotInterconnects Keynote, August 2002

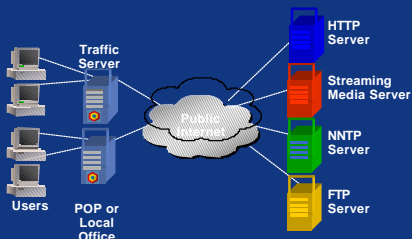
Caching Throughout the Network



What Is Network Caching?



Caching stores content objects from an origin server closer to the user, where they can be retrieved more quickly



HotInterconnects Keynote, August 2002

The New Internet



- ◆ There's a whole new layer on top...
- ◆ Redefines naming/routing
- ◆ Stores data in the middle of the network
- ◆ Adds new services for fault tolerance
- ◆ Adds transformation of data on the fly
- ◆ Adds distribution control (push/freshness)
- ◆ ... and it appeared silently almost overnight

HotInterconnects Keynote, August 2002

This is a New Internet

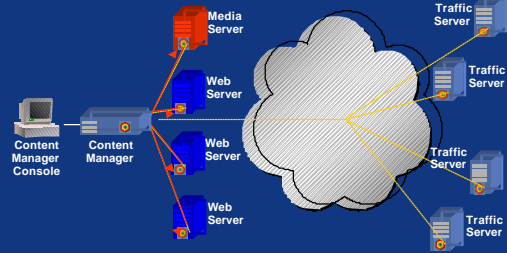


- ◆ The **action** is between the caches, **NOT** between the routers
- ◆ There are new protocols in use to:
 - push content to the edge
 - invalidate remote content for freshness
 - collate remote logs into a single log
 - A/V streaming that works

HotInterconnects Keynote, August 2002

Information Backflow

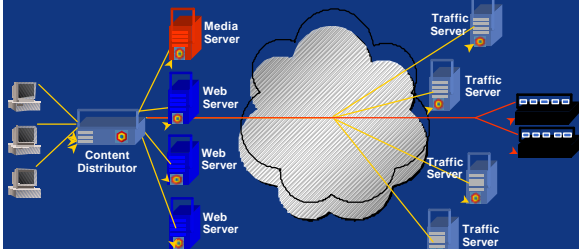
Aggregates data about content usage and performance
 Tracks whether service level requirements are being met
 Dynamically adjusts content availability to meet those requirements



Content Delivery



Retrieves content from development or management source
 Replicates and distributes content to destination servers and caches
 Provides content routing information to load balancer

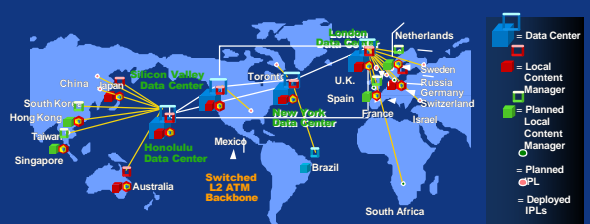


HotInterconnects Keynote, August 2002

Example: Digital Island



Global content distribution network provides fast access to content in every country



HotInterconnects Keynote, August 2002

Where is it Going?



- ◆ Right now we have **many** new Internets
 - Digital Island, Akamai & others compete
 - Easy progress for wholly-owned networks...
- ◆ Prediction: repeat the past
 - Large networks will need to peer at the **content level** not the packet level
 - BGP did this at the packet level (IP)
 - But this is the federation problem... very hard.

HotInterconnects Keynote, August 2002

Transforming Content at the Edge



Web Content is designed for PCs with high speed connections:

- 24-bit color graphics
- Animated GIFs
- Applets
- Frames, tables and other formatting
- Mouse-oriented navigation



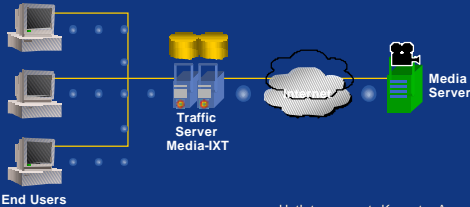
Single copy of content capable of supporting multiple needs

HotInterconnects Keynote, August 2002

Edge Delivery of Streaming Media



- ◆ Reduces network traffic
- ◆ Optimizes content quality
- ◆ Leverages high local bandwidth



HotInterconnects Keynote, August 2002

Conclusions



- ◆ A single top-down network is easy
- ◆ Harder:
 - Partitions, security, multiplexing
- ◆ Much harder:
 - Unknown topology and growth
 - Evolution
 - Federation
 - Computing/storage in the network
- ◆ Open problems:
 - Federation + evolution
 - Software in the network
 - Network interface?

HotInterconnects Keynote, August 2002