Internet Services

Introduction to Internet Services

Richard Martin

Rutgers University
Department of Computer Science

January 2003

Outline

- Definition
- History
- Overview
- Fox Paper
- Brewer paper
- Open Problems

Definition

- What is an Internet Service?
- Any client server interaction over the Internet?
- What is a service?
  - Client is a "user" of the server
  - Shared resources
  - Information or "transactions" are the key abstractions, not packets or data

Examples

- Auctions
  - Ebay, yahoo auctions
- Trading
  - E*Trade, schwab
- Shopping
  - Amazon model
- Entertainment and news
  - Cnn, AOL
- Computing Services ("grid")
  - Seti @ home,
- Communication
  - Hotmail, AOL instant messanger

Internet Service History

- 1970's email+FTP
- 1980's email, ftp, news
- Early 1990's: +gopher wais (search)
- Mid 1990's: +web, (hypertext)
  - added search later
- Late 1990's: +DB access
- Today: +radio, movie/ TV shorts
- Tomorrow?

Lesions from Giant-Scale Internet Servics

- Author: Eric Brewer
  - professor at UC Berkeley and Chief Scientist of Inktomi

Challenges

- High availability
- On-line evolution
- Growth

Examples

Table A. Example clusters for giant-scale services.

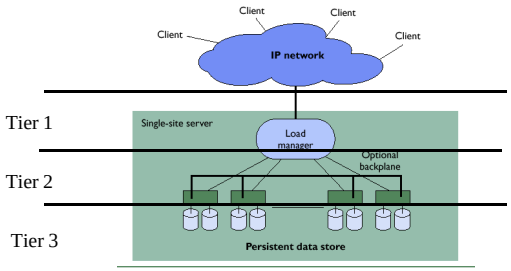
Service	Nodes	Queries	Nodes
AOL Web cache	>1,000	10B/day	4-CPU DEC 4100s
Inktomi search engine	>1,000	>80M/day	2-CPU Sun Workstations
Geocities	>300	>25M/day	PC Based
Anonymous Web-based e-mail	>5,000	>1B/day	FreeBSD PCs

google >10000 80+M day 1-2 processor PC's

Tier Functions

- Tier 1: Request Distribution
  - load balancers, front ends, TP monitors
- Tier 2: Web servicing, data aggregation
  - Web serves, distillers
- Tier 3: Persistent state
  - Databases and storage modules

Tiered Models



Availability Metrics

- Classic analysis:
  - Mean time to Failure (MTTF)
  - Mean time to Repair (MTTR)
  - Average Uptime

New Metrics

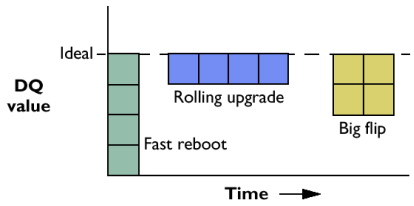
- New, 2 dimensional space:
- Yield: % of queries completed per second
- Harvest: % of data available

DQ Principle

- Total amount of data deliverable is constant
  - Bandwidth constraint
  - Data per query * Queries/sec = C
- Service quality defined by how much of total potential, C, is delivered

  New metric needed to reason about both "data" and "queries"

DQ In practice



Which method is a better upgrade strategy?

When is DQ useful

- Can the data be partitioned?
  - Different users -> different datasets
- Can it be replicated?
  - Auctions
  - Trading
  - Shopping
  - Entertainment and news
  - Computing Services ("grid")
  - Communication

Big Ideas

- Clustering and tiers architecture/clusters are the only way to go. (why?)
- Dual metrics needed to capture availability
- Constant = DQ is the best way to measure availability

Cluster based Scalable Network Services

- Authors:
  - Armando Fox: Prof. at Stanford
  - Steve Gribble: Prof. at Washington
  - Yatin Chawathe: ICIRI Research Lab
  - Eric Brewer
  - Paul Gauthier: CTO of Inktomi

Paper ideas

- Argues for cluster architecture
  - Seems natural today, what was the big deal?
- ACID vs. Base
- Presents scaling study

ACID vs. BASE

- ACID:
  - Atomic, consistent, isolated, durable
- BASE
  - Basically available, soft state, eventual consistency
- What do these really mean?

BASE Universe

- Stale data OK
- Use Soft state
  - Non-durable and eventually consistent
- Approximate answers are OK
- Compare to ACID:
  - How would a database designer react to the above statements?
  - Would you want your DBMS to behave in a BASE manner?

Layered Software Model

Service: Service-specific code
- Workers that present human interface to what TACC modules do, including device-specific presentation
- User interface to control the service

TACC: Transformation,Aggregation, Caching, Customization
- API for composition of stateless data transformation and content aggregation modules
- Uniform caching of original, post-aggregation and post-transformation data
- Transparent access to Customization database

SNS: Scalable Network Service support
- Incremental and absolute scalability
- Worker load balancing and overflow management
- Front-end availability, fault tolerance mechanisms
- System monitoring and logging

Two Services

- Transend web proxy service
  - Front end for devices, modems
- HotBot search engine

Scalability

Requests/ Second	# Front Ends	# Distillers	Element that saturated
0-24	1	1	distillers
25-47	1	2	distillers
48-72	1	3	distillers
73-87	1	4	FE Ethernet
88-91	2	4	distillers
92-112	2	5	distillers
113-135	2	6	distillers & FE Ethernet
136-159	3	7	distillers

Scalability Study

- Does a demonstration of 7 distillers and 3 front demonstrate scalability?
- "pilot plant" type of demonstration
  - build working scale model of the actual system
  - Develop ideas so that most real problems are uncovered

See "mythical man-month" by Fred Brooks

Summary

- Architecture panned out
- Software model a little fuzzy
  - is CGI scripts + a DB the same thing?
- Philosophy of of BASE used
  - But not well understood what that means compared to ACID