

Outline for today

- Feasibility of a Serverless Distributed File System deployed on an Existing set of Desktop PCs – Microsoft research. ACM SIGMETRICS 2000

MSR Serverless Distributed File System

- They've actually implemented this system within Microsoft and hence have real results
- Assumption 1: not-fully-trusted environment
- Assumption 2: Disk space is not that free
- Each disk is partitioned into three areas:
 - Scratch area – for local computations
 - Global storage area
 - Local cache for global storage

Efficiency consideration

- Compress data in storage
- Coalesce distinct files that have identical contents
 - Probably an artifact of Windows environment that stores files in specific locations e.g. c:\windows\system\
- File are replicated
 - Machines that are topologically close
 - Machines that are lightly loaded
 - Non-cache reads and writes to prevent buffer cache pollution

Replica management

- Files in a directory are replicated together
- When new machines join, its data is replicated to other machines
- Replicas of other files are moved into the new machine
- When machine leaves, the data in that machine is replicated in other machines from other replicas

Security

- File updates are digitally signed
- File contents are encrypted before replication
- Convergent encryption to coalesce encrypted file
- Encryption:
 1. Hash(file contents) -> uniqueHash
 1. Encrypt(unencryptedfile, uniqueHash)->encryptedfile
 1. User1: encrypt(UserKey1, uniqueHash) -> Key1
 2. User2: encrypt(UserKey2, uniqueHash) -> Key2
- Decryption
- User1: decrypt(UserKey1, Key1) -> uniqueHash
- Decrypt(encryptedfile, uniqueHash) -> unencryptedfile

Application API

- Related read, write operations to objects form a session (defined by the application developer)
- Users specify the session guarantees required for each session
- Applications can register call back functions for exceptions

Transactions (Database technology)

- A **transaction** is a program unit that must be executed atomically; either the entire unit is executed or none at all. The transaction either completes in its entirety, or it does not (or at least, nothing appears to have happened).
- A transaction can generally be thought of as a sequence of reads and writes, which is either **committed** or **aborted**. A committed transaction is one that has been completed entirely and successfully, whereas an aborted transaction is one that has not. If a transaction is aborted, then the state of the system must be **rolled-back** to the state it had before the aborted transaction began.

ACID semantics

- Atomicity – each transaction is atomic, every operation succeeds or none at all
- Consistency – maintaining correct invariants across the data before and after the transaction
- Isolation - either has the value before the atomic action or after it, but never intermediate
- Durability – persistent on stable storage (backups, transaction logging, checkpoints)

Relaxed semantics

- Relax the ACID constraints
- We could relax consistency for better performance (ala Bayou) where you are willing to tolerate inconsistent data for better performance. For example, you are willing to work with partial calendar update and are willing to work with partial information rather than wait for confirmed data. More on this later on in the course.

Naming and Location Management

- The Anatomy of a Context-Aware Application – Andy Harter, Andy Hopper, Pete Steggles, Andy Ward and Paul Webster. AT&T Labs, Cambridge, UK



History of AT&T Cambridge Lab

- It used to be Olivetti Research Lab
- ORL developed active badges, VNC etc..
- VNC allows you to
- teleport across machines
- by moving the display
- to another terminal
- near you



Motivation for context-aware application

- Users application should be available where-ever the user goes, in a suitably adapted form
- Context aware application is one which adapts its behavior to a changing environment
 - E.g. Follow-Me applications
- Context aware applications need to know the location of users and equipment, and the capabilities of the equipment and networking infrastructure

Components of Context-Aware Applications

- A fine-grained location system, which is used to locate and identify objects
- A detailed data model, which describes the essential real-world entities that are involved in mobile applications
- A persistent distributed object system, which presents that data model in a form accessible to applications

Components of Context Aware Applications

- Resource monitors, which run on networked equipment and communicate status information in a centralized repository
- A spatial monitoring, which enables event-based location-aware applications
- Important lesson: It is very important to present location information in a form suitable to the application

Location system

- Out-door: Global Positioning Satellite (GPS)
 - Developed by US military, available for civilian use
 - Locates using triangulation with multiple satellites
 - Accurate up to a few meters. (Military version more accurate)
- In-door: Radio-based, Infra-red, Ultra-sound
 - For indoor systems, you have to worry about interference from multipath sources, obstructions and devices (such as fluorescent bulb ballasts)
 - Ultrasound is more immune to these interferences

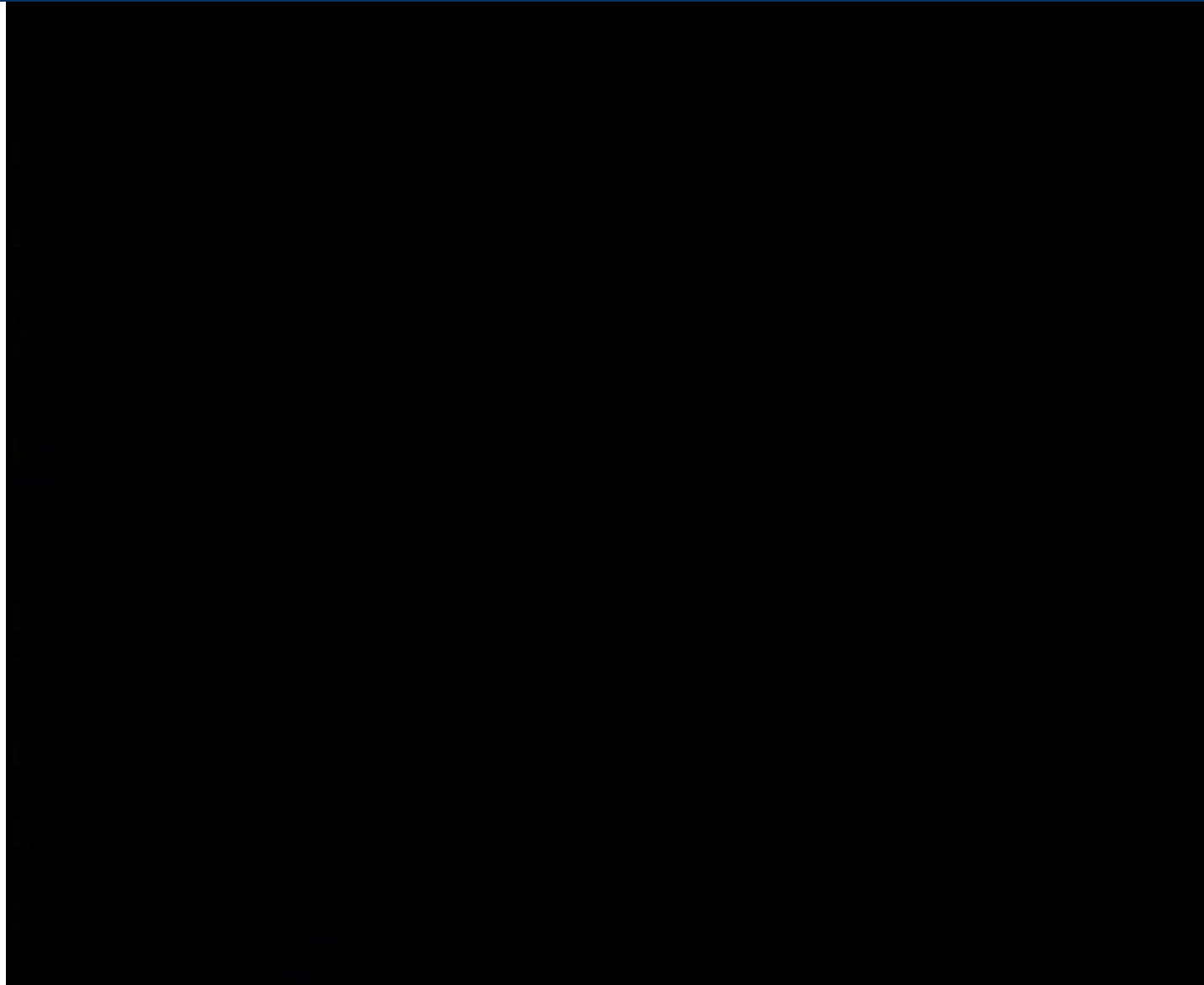
Bat Unit

- Radio transceiver, ultrasonic transducer and control logic
- Each bat has a GUID
- Use the radio, ultrasonic transducer and the speed of sound in air (estimated from ambient temperature) to estimate location
- Use multiple receivers to get 3D location using multilateration
- Reflections of ultrasonic waves – statistical outlier elimination

Bats



Sentient Computing



Discussion

