Data on Air: Organization and Access

Imielinski, Viswanathan, Badrinath

Randy Chong

CS525mc S04

PDA's …

- Are almost everywhere…

- At least for the technologically-inclined.

Pilot vs. Newton

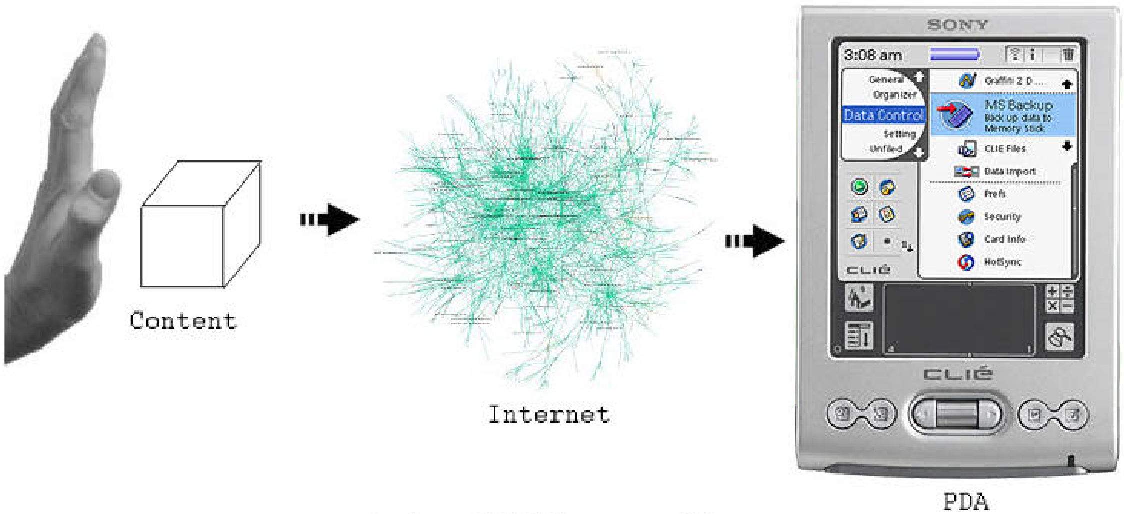
Clié

2,500,000+ PDA's Shipped Globally in Q3 2003

http://www.dataquest.com/press_gartner/quickstats/personal.html

Enter the Internet

- We have lots of very portable computers.
- We have a global network.
- We have to keep up with the news, weather, stocks, etc.
- We should use this infrastructure to push content to our PDA's so we can keep up with information from anywhere!



Brilliant!

But how are We Going to Do This?

- We must use a protocol.
- But, there is not an accepted protocol for such a medium.
- So we must create a protocol.
- The protocol must facilitate the mobility of the devices.
  - Devices are battery-powered, so power is in “short” supply.
  - Receiving and transmitting data is expensive.
  - Transmitting data is really expensive
  - Probability of a mangled/lost packet on a wireless network is an order high than that of wired network.

Reducing Power Consumption

- Let the PDA transmit data as little as possible.
- Use an asymmetric “pipe” where the downstream is larger than the upstream.
  - Remember, lots of power required for the upstream.
  - Not as much for the downstream.
  - Exploit the asymmetry!
    - Do not let the client upload at all.
    - The transmission of the downstream is handled by the server, which can be a powered, stationary computer.

The Crux of the Problem

- Use broadcasting, continuously send content to any device that is listening.
  - The redundancy helps solve the mangled/lost packet issue inherent in wireless communications.
- Receiving content still consumes a considerable amount of power.
- Listen only to content that needs to be downloaded.
- The best way to do this is via indexing and broadcast disks.

Broadcast Disks

- Since content is constantly being broadcast, it is as if there is a persistent disk in the airwaves.
- To access data on any disk, we need:
  - A reference.
  - A file system.
  - Or simply an index.

Indices

- A listing of when certain content will be broadcast.
- Generally, should be transmitted (at least) at the beginning of each broadcast.
- Overhead, which leads to more power consumption on the PDA.
- Indices must be optimized to minimize overhead.
- However, indices should also be optimized to minimize latency and tuning time.
  - Latency is the amount of time elapsed for all of the desired content to be retrieved.
  - Tuning time is the amount of time that is spent listening to the broadcast.

Clustered Indexing

- Optimized Solutions
  - Latency_opt gives the lowest latency, but a large tuning time.
  - Tuning_opt gives the lowest tuning time, but a large latency.
- (1, m) Indexing
  - Transmit the entire index m times for each broadcast.
  - Each data “bucket” contains a pointer to the next index transmission.
  - Procedure
    - Tune in and get the pointer to the next index.
    - Sleep until the next index is available.
    - Retrieve the index and get the pointer to the desired content.
    - Sleep until the desired content is available.
    - Retrieve the desired content.

Clustered Indexing

- Distributed Indexing
  - Instead of sending the entire index multiple times, only send the index of the data that will be immediately available.
    - A tree-type index.
    - Minimizes index overhead.
  - Three types
    - Non-replicated distribution.
    - Entire path replication.
    - Partial path replication.
  - Partial path replication is middle ground for the other two.

Clustered Indexing

- Comparison
  - Both (1, m) and distributed indexing have less latency than tune_opt.
  - Distributed indexing has mush less latency than (1, m).
  - Distributed indexing almost achieves the latency of latency_opt.
  - Latency_opt has the largest tuning time of all of the algorithms.
  - (1, m) has a tuning time almost the same as tune_opt.
  - Distributed indexing has a tuning time is just slightly larger than that of tune_opt.

Nonclustered Indexing

- Optimized solutions
  - Noncluster_latency_opt provides the best latency but a large tuning time for non-clustered indexing.
  - Noncluster_tune_opt provides the least tuning time but a large latency.
- Nonclustered Indexing Algorithm
  - The entire broadcast is divided into meta-segments.
  - Meta-segments are further divided into data segments.
  - Data segments contain only data buckets of the same content attribute.
  - The end of each data segment contains a pointer to the next available meta-segment that contains a data segment for the desired content attribute.

Nonclustered Indexing

- Comparison
  - The nonclustered indexing algorithm always has a better tuning time than noncluster_latency_opt.
  - The nonclustered indexing algorithm has a tuning time that is nearly the same as the tuning time achieved with noncluster_tune_opt.
  - The latency of nonclustered indexing depends on the scattering factor.
    - If the scattering factor is small, then nonclustered indexing has a latency that is almost that of nonclustered_latency_opt.

Multiple Indices

- Multiple Indices Algorithm
  - Let each bucket in the broadcast have n attributes.
  - Attributes with lower values of n are accessed more frequently while the attribute with the value n is accessed least frequently.
  - Divide the broadcast into meta-segments based on each attribute.
  - Additionally, the buckets of indexed attributes are divided into meta-segments as well.
  - There are now LOTS of meta-segments in the broadcast now.

Conclusions

- “Adding indexes increases the latency but provides radical improvement in terms of tuning time and consequently improves battery utilization.”
- “The resulting latency and tuning time [of distributed indexing] is close to the optimum as our analyses indicate.”
- Not much concluded on the multiple indexing algorithm.

Future Work

- Further analyze the multiple indexing algorithm.
- Implement prototype clients and servers.
- Investigate broadcasting content over sub-channels ala channels on cordless phone system.