CS538: Advanced Topics in Information Systems
Black Box: Distributed Storage [GMM]

Consistent
Location transparency
Data Persistency
Secure
Available
Real-Time
Desirable Properties:
- Good query interface
- Power efficiency, long lifetime
- Scalability
- Adaptivity
- Low response time (high throughput)
Desirable Properties wrt Google [GMM]

- **Input**
  - Keyword(s)

- **Output**
  - Will return to the user what the user wants/needs and NOT what the search engine thinks you want/need.
Black Box: Privacy [GMM]

- The primary task in data mining: development of models about aggregated data.
- Can we develop accurate models without access to precise information in individual data records?

Diagram:
- User input
- Data mining server
- Aggregated data like mean and etc
Black Box: Infrastructure [MPV]

- Reliable
- Time-Efficient
- Cost-Efficient
- Robust
- Scalable
- Secure
Black Box: Distributed Interactive App. [MPV]

Consistent
Scalable
Secure
Robust
Available
Real-Time
Block Box: Embedded Software [Lym]

Main Features
- Timeliness
- Concurrency
- Liveness
- Heterogeneity
- Reactivity
- Robustness
- Low power
- Scaleable
## Some Black Boxes

<table>
<thead>
<tr>
<th>Distributed Storage</th>
<th>Sensor Database</th>
<th>Digital Archive</th>
<th>Internet Query; Google</th>
<th>Privacy Data mining</th>
<th>Grid</th>
<th>Communication Infrastructure</th>
<th>Smart Env.</th>
<th>Distributed Interactive Applications</th>
<th>Embedded Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location transparency</td>
<td>return what the user wants</td>
<td>aggregate data preserving privacy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Consistent</td>
<td></td>
</tr>
<tr>
<td>Consistent Data Persistency</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Timeliness</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Concurrency</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Cost-Efficient</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Liveness</td>
<td></td>
</tr>
<tr>
<td>Real-Time</td>
<td>Low response time (high throughput)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Real-Time</td>
<td>Reactivity</td>
</tr>
<tr>
<td></td>
<td>Adaptivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Heterogeneity</td>
</tr>
<tr>
<td></td>
<td>Power efficiency, long lifetime</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Low power</td>
</tr>
<tr>
<td></td>
<td>Scalability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Scalable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Good query interface</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Scalable</td>
<td>Scaleable</td>
</tr>
<tr>
<td>Secure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Secure</td>
<td></td>
</tr>
<tr>
<td>Robust</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Robust</td>
<td>Robustness</td>
</tr>
<tr>
<td>Available</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Reliable</td>
<td>Available</td>
</tr>
</tbody>
</table>
Some Specific Problems

- [dist. storage] What data should be stored by external providers? How to deal with partition (how to deal with consistency)? How to deal with the 3f + 1 assumption?

- [sensor database] What can sensor databases do? What are the differences compared with traditional distributed databases, with streaming databases?

- [digital archive] How to keep old files? How to search video databases (e.g., tell me all videos containing scenes where John Wayne rode in front of White House)?

- [search engine] How to search the Internet? How do you know that you have a good result?

- [privacy] How to preserve correlation in preserving privacy? What is privacy? What are the trade-offs between privacy and information access?
Some Specific Problems (cont')

- [ubicomp] What is an addressing scheme of an environment with a large number of small devices?
- [grid] How to build a Grid computing environment at a specific environment, say Yale CS?
- [distributed interactive application] What is the bandwidth requirement? How to build a high quality “teleconference” system?
- [embedded OS] What are embedded devices? How to program embedded devices?
- [optical] Where should the optical routers be, edge or backbone? How much bandwidth do we need?