**Locally Repairable Codes**
Dimitris S. Papailiopoulos and Alexandros G. Dimakis

### Big Data
- Big Data Players (Facebook, Amazon, Google, Yahoo, ...)  
- FB has the biggest Hadoop cluster (B0PB)

- Failures are the norm.  
- We need to protect the data: **Introduce redundancy**

### Reliability: Replication vs. Codes
- \( (n,k) \)-MDS codes have optimal reliability for given storage  
- 8% of Facebook Archival storage uses coding (most is still 3x replication)  
- Plans to code 50% of archival data

### MDS-Codes: Pros & Cons

<table>
<thead>
<tr>
<th>Time</th>
<th>Repair Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fri</td>
<td>4 PB</td>
</tr>
<tr>
<td>Mon</td>
<td>6 PB</td>
</tr>
<tr>
<td>Wed</td>
<td>5 PB</td>
</tr>
<tr>
<td>Thu</td>
<td>4 PB</td>
</tr>
</tbody>
</table>

**Metrics of interest:**
- Bits communicated for repair  
- Bits read for repairs  
- Locality = Number of Nodes used during repair.

### The Code Repair Problem
- A node is lost: We need to exactly repair it.  
- Practice: ALL nodes are contacted,  
  **everything** is downloaded for repair (Hadoop)  
  (matrix inversions take place)

**Naive repair:**  
1. generates enormous communication  
2. accesses a great number of nodes

### Locally Repairable Codes

**Tradeoff between Locality- Reliability?**

**Implementing LRC**
[Sathiamoorthy, Asteris, P, Dimakis, facebook]  
- LRC was tested on Facebook clusters and Amazon ec2 clusters (100 machines).  
- Reduces disk IO and network bandwidth by approximately 2x  
- Available online (Apache licence)

Under testing for use in production at Facebook.