

# Data Partitioning for Private LOCKSS Networks

Tobin Cataldo  
Information Systems Manager  
Birmingham Public Library  
Birmingham, Alabama  
[tcataldo@bham.lib.al.us](mailto:tcataldo@bham.lib.al.us)

Scaling out by adding additional commodity nodes to a PLN introduces greater (potentially heterogeneous) hardware redundancy and increases network storage capacity when employing a collection policy that establishes a maximum number of AU replicas.

Utilizing per-node title lists to populate custom Add AU screens is the most sustainable model for PLNs with high AU counts.

## **Assertions**

PLNs should establish a baseline number of AU replicas

PLNs should favor low-cost, commodity hardware and scale out

PLNs should establish a sustainable model by utilizing custom title lists

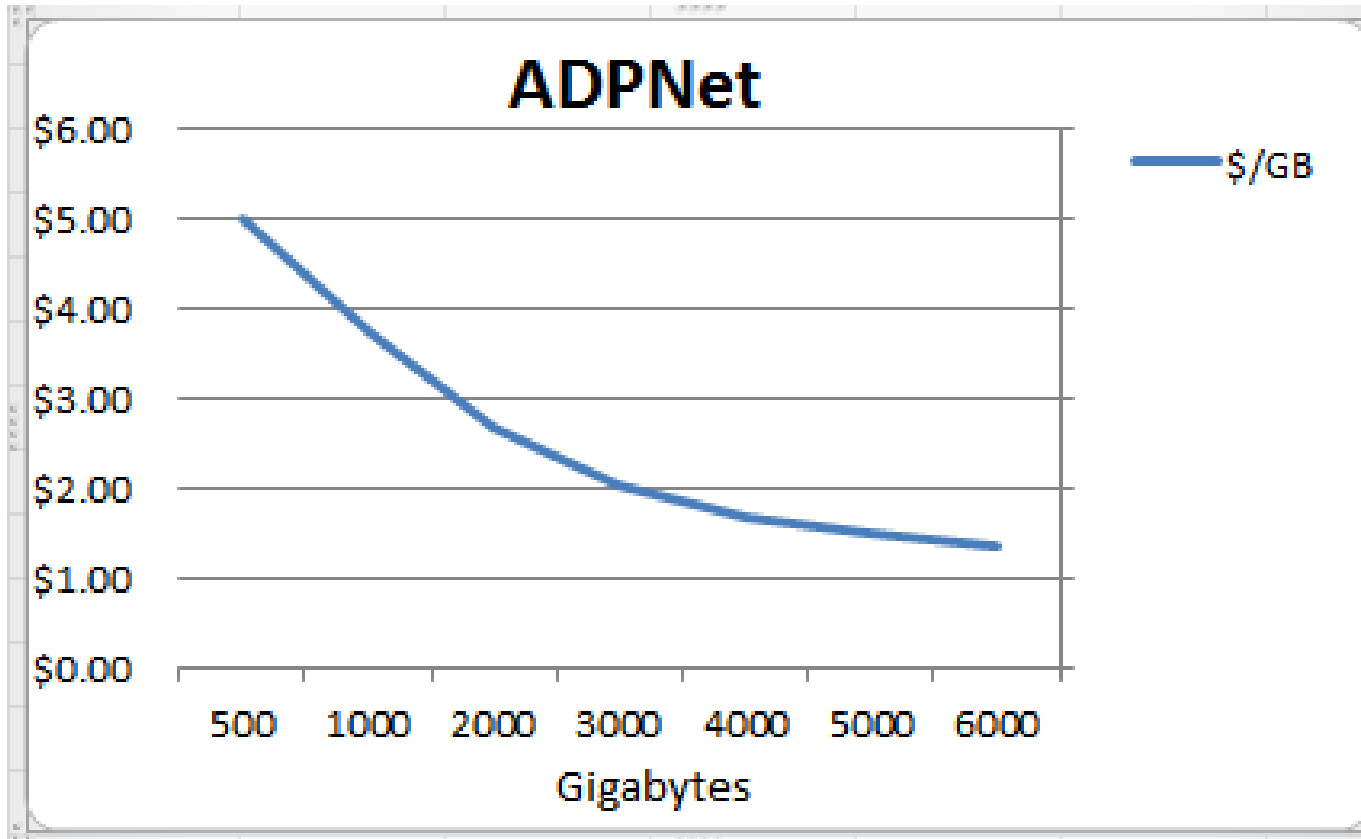
PLNs should shelve the perceived immutability of cache data

## Alabama Digital Preservation Network (ADPNet) Quick Facts

- 8 nodes
- Released AUs ~ 1000
- Capacity 10.6 TB (Raw 16 TB)
- Consumed space ~ 6 TB
- Informal [anticipated] growth (through 2014) 15-20 TB

# Birmingham Public Library

## ADPNet Pricing Model

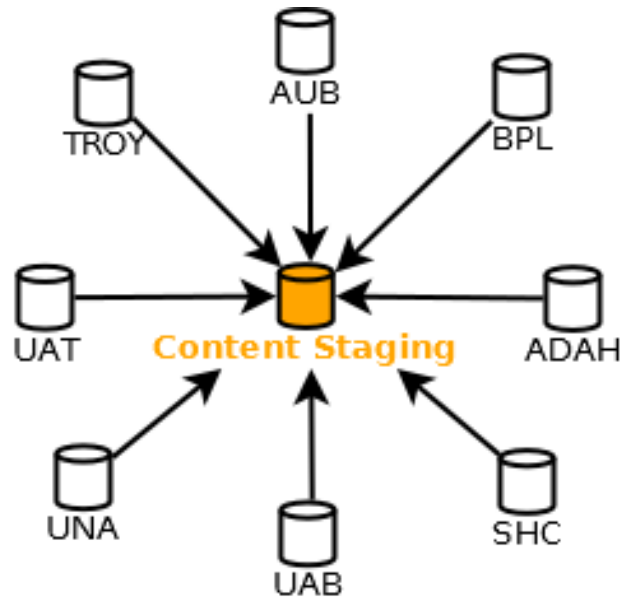


Pricing favors higher consumption

# Birmingham Public Library

## ADPNet Collection Policy

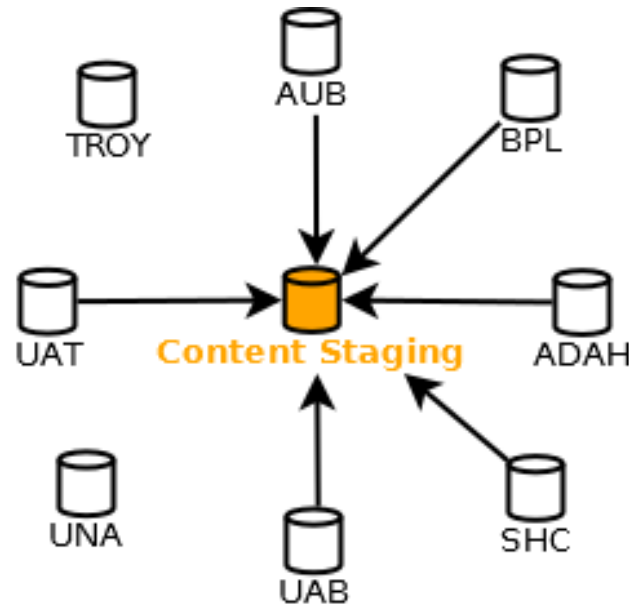
All Nodes Collect Every AU



# Birmingham Public Library

## Data Partitioning 1 + 5

AU Owner + 5 random (distinct) nodes (not AU owner)



In the case where publisher is not a peer, 6 random nodes

BPL (89) : ADAH, AUB, BPL, SHC, UAB, UAT1

# Birmingham Public Library

## Data Partitioning 1 + 5

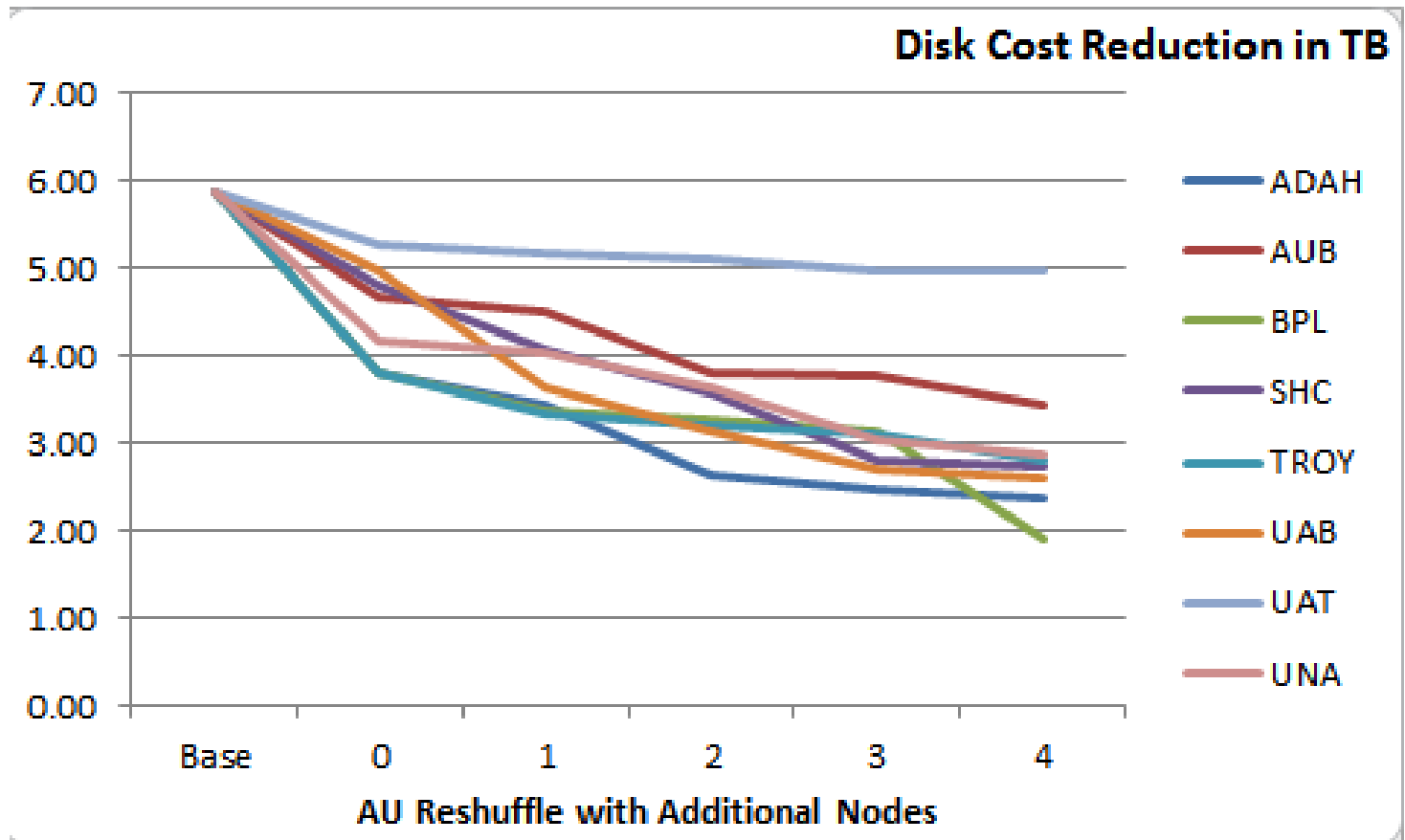
Extant Storage Burden in TB		
	Base	1 + 5
ADAH	5.87	3.79
AUB	5.87	4.66
BPL	5.87	3.79
SHC	5.87	4.79
TROY	5.87	3.80
UAB	5.87	4.96
UAT	5.87	5.26
UNA	5.87	4.18

25% average per node cost reduction



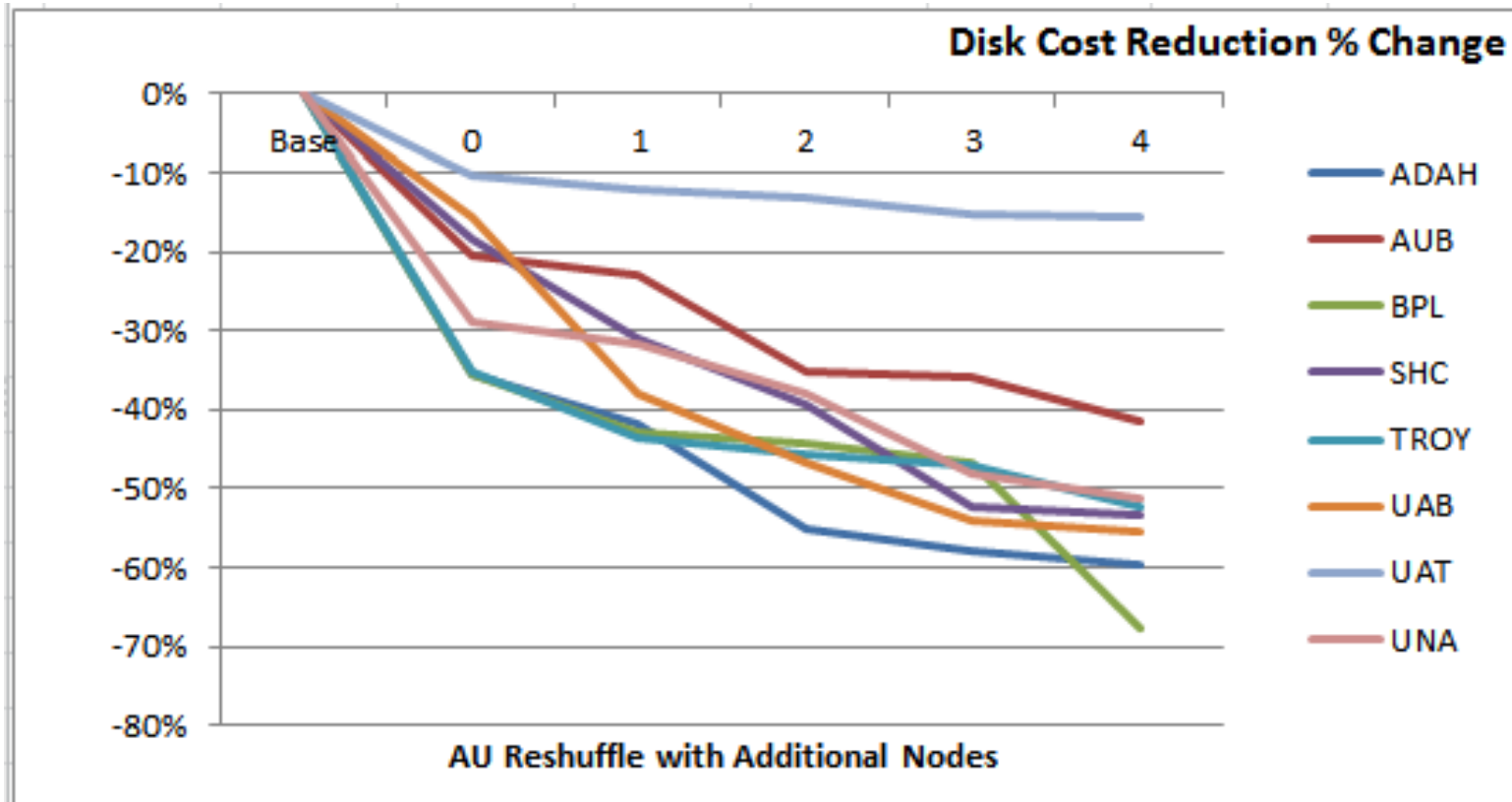
# Birmingham Public Library

## Node Cost Reduction in TB



# Birmingham Public Library

## Cost Reduction as a Percent



# Birmingham Public Library

## Implementation

### **Initial**

~1000 AUs for base allocation

### **On-going**

AUs typically released in batches (50-75 typically)

Customize the Title List

Add AUs only lists allocated AUs

# Birmingham Public Library

## LOCKSS Title List Snippet (1 cluster)

```
<lockss-config>
  <property name="org.lockss.titleSet">
    <property name="Birmingham Public Library">
      <property name="name" value="All Birmingham Public Library AUs" />
      <property name="class" value="xpath" />
      <property name="xpath" value="[attributes/publisher='Birmingham Public Library']" />
    </property>
  </property>
  <property name="org.lockss.title">
    <property name="BirminghamPublicLibraryBasePluginBirminghamPublicLibraryCartographyCollectionMaps000400000599">
      <property name="attributes.publisher" value="Birmingham Public Library" />
      <property name="journalTitle" value="Birmingham Public Library Cartography Collection" />
      <property name="type" value="journal" />
      <property name="title" value="Birmingham Public Library Cartography Collection: Maps (000400-000599)" />
      <property name="plugin" value="org.bplonline.adpn.BirminghamPublicLibraryBasePlugin" />
      <property name="param.1">
        <property name="key" value="base_url" />
        <property name="value" value="http://bpldb.bplonline.org/adpn/load/" />
      </property>
      <property name="param.2">
        <property name="key" value="group" />
        <property name="value" value="Cartography" />
      </property>
      <property name="param.3">
        <property name="key" value="collection" />
        <property name="value" value="000400-000599" />
      </property>
    </property>
  </property>
</lockss-config>
```

## XML Deserialization with XmlAnyElement

```
[XmlRoot("lockss-config")]  
public class LockssTitleDb  
{  
    [XmlAnyElement()]  
    public XmlElement[] LockssTitleSet;  
}
```

## XML Title Cluster Repetition

```
<lockss-config>
<property name="org.lockss.titleSet">
  ...
</property>
<property name="org.lockss.title">
  ...
</property>
<property name="org.lockss.titleSet">
  ...
</property>
<property name="org.lockss.title">
  ...
</property>
</lockss-config>
```

## Property Cluster in Groups of 2

```
using (FileStream _f = new FileStream(Server.MapPath(@"titledb.xml"), FileMode.Open, FileAccess.Read))
{
    XmlReader _xml = XmlReader.Create(_f);
    XmlSerializer _xs = new XmlSerializer(typeof(LockssTitleDb));
    LockssTitleDb _titles = (LockssTitleDb)_xs.Deserialize(_xml);

    for (int i = 0; i < _titles.LockssTitleSet.Length; i++)
    {
        if (i % 2 == 0)
        {
            // even number check
            // org.lockss.titleSet
        }
        else
        {
            // org.lockss.title
        }
    }
}
```

## Property Cluster Expanded

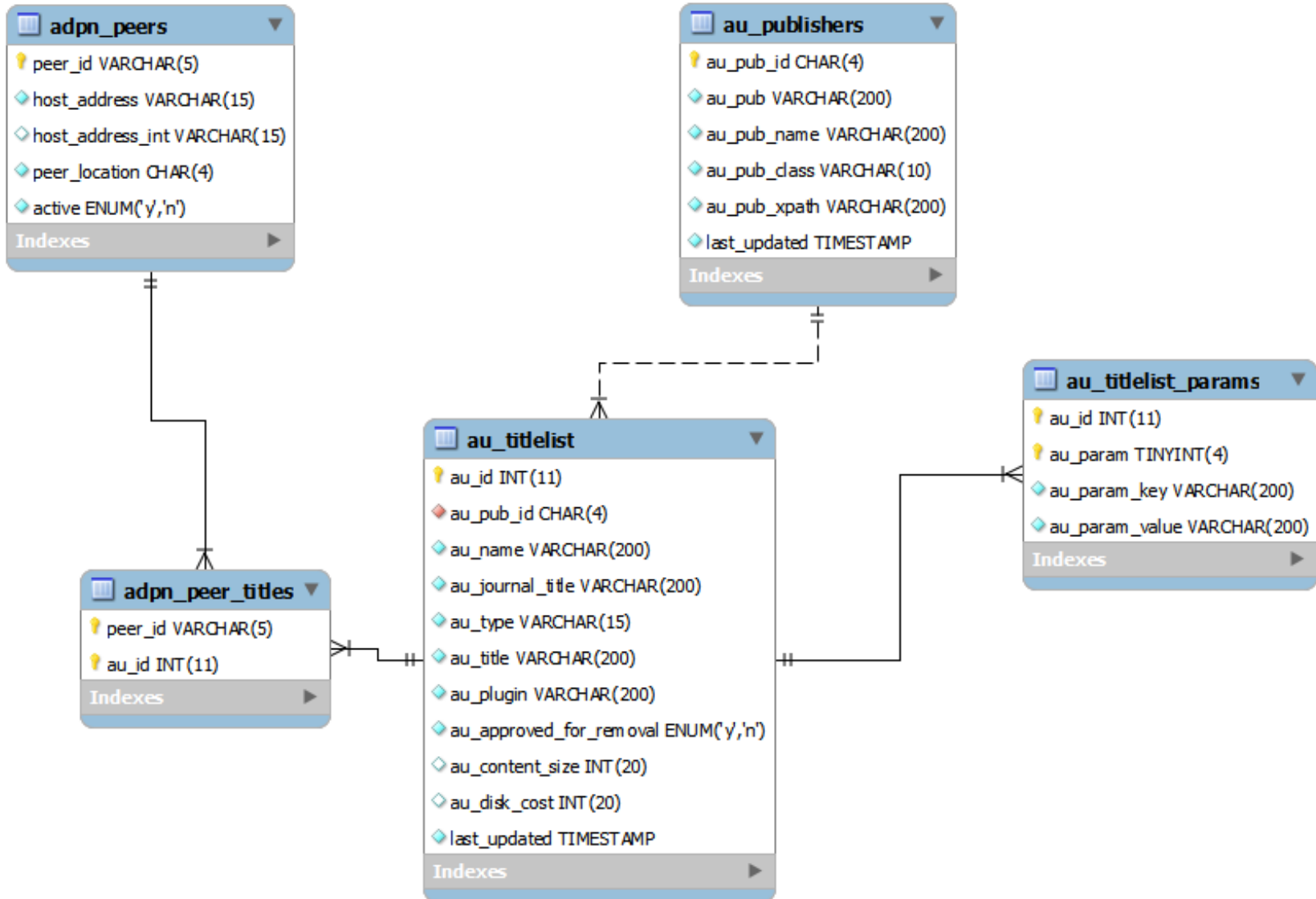
```
using (FileStream _f = new FileStream(Server.MapPath(@"titledb.xml"), FileMode.Open, FileAccess.Read))
{
    XmlReader _xml = XmlReader.Create(_f);
    XmlSerializer _xs = new XmlSerializer(typeof(LockssTitleDb));
    LockssTitleDb _titles = (LockssTitleDb)_xs.Deserialize(_xml);

    for (int i = 0; i < _titles.LockssTitleSet.Length; i++)
    {
        if (i % 2 == 0)
        {
            // even number check
            // org.lockss.titleSet
        }
        else
        {
            // org.lockss.title
            // check for attribute value org.lockss.title
            string _au_id = _titles.LockssTitleSet[i].ChildNodes[j].Attributes[0].Value;
            string _journal_title = null;
            string _type = null;
            string _title = null;
            string _plugin = null;
            System.Collections.ArrayList _params = new System.Collections.ArrayList();
            bool _removal = false;
        }
    }
}
```



# Birmingham Public Library

## Local Data Store



# Birmingham Public Library

## Combined Title list and AU DaemonStatusService Detail

### **AU Detail**

Selected AU ID : 89

Pub ID : BPL

AU Normalized Name : BirminghamPublicLibraryBasePluginBirminghamPublicLibraryCartographyCollectionMaps000400000599

AU Title: Birmingham Public Library Cartography Collection: Maps (000400-000599)

AU Journal Title : Birmingham Public Library Cartography Collection

AU Plugin : org.bplonline.adpn.BirminghamPublicLibraryBasePlugin

AU Params :

param.1 key=base\_url value=http://bpldb.bplonline.org/adpn/load/

param.2 key=group value=Cartography

param.3 key=collection value=000400-000599

AU Approved For Removal : n

AU Content Size : 12,562.26 MB

AU Disk Cost : 12,606.86 MB

Assigned Peers : ADAH, AUB, BPL, SHC, UAB, UAT1

## Allocating Archival Units (Basic)

### 1. Base Distribution

Randomize array, inject least used nodes into array  
Includes initial and on-going allocation consideration

### 2. New Node Reshuffle

Randomize array, only update data when new node is “up”

### 3. Dead Node Reshuffle

Remove dead node from `adpn_peer_titles`, create an enumerable where count is less than baseline

## Allocating Archival Units (Basic)

### 1. Base Distribution

Randomize array, inject least used nodes into array

- Shuffle Array
- Swap array[0] with least used by au count
- Swap array[1] with least used by disk cost
- Loop for 5 if publisher is peer else 6

## Allocating Archival Units (Basic)

### 2. New Node Reshuffle

Randomize array, only update data when new node is “up”

- Shuffle Array
- Loop for 5 if publisher is peer else 6,
  - if new node is up, random change one of existing AU assignments

```
UPDATE `adpn_peer_titles` AS `a`,  
      (SELECT `b`.`peer_id`  
       FROM `adpn_peer_titles` AS `b`  
       WHERE `b`.`au_id` = ?  
       AND `b`.`peer_id` != ?  
       ORDER BY RAND() LIMIT 1) AS `b`  
SET `a`.`peer_id` = ?  
WHERE `a`.`peer_id` = `b`.`peer_id`  
AND `a`.`au_id` = ?
```

## Allocating Archival Units (Basic)

### 3. Dead Node Reshuffle

Remove dead node from `adpn_peer_titles`, create an enumerable where count is less than baseline (6)

- For each AU, insert a random new peer not already assigned to AU.

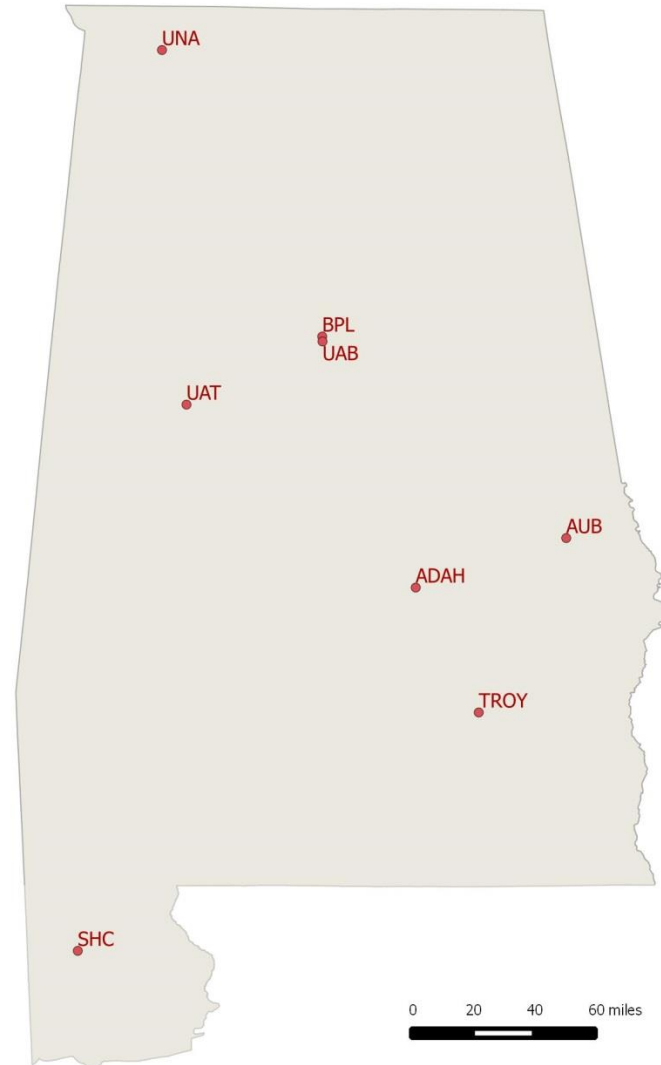
```
INSERT INTO `adpn_peer_titles` (`peer_id`, `au_id`)
  (SELECT `adpn_peers`.`peer_id`, (SELECT `au_id` FROM `au_titlelist` WHERE `au_id` = ?) AS 'au_id'
   FROM `adpn_peers`
  WHERE `adpn_peers`.`active` = 'y'
  AND `adpn_peers`.`peer_id` NOT IN (SELECT `peer_id` FROM `adpn_peer_titles` WHERE `au_id` = ?)
  ORDER BY RAND() LIMIT 1
```

# Birmingham Public Library

Alabama Digital Preservation Network Nodes

## Elaborate Allocations

- Geo-location
- Uber-nodes
- Multi-node sites



## Expert Config

`org.lockss.titleDBs = http://.../GenerateXml.aspx?peer=BPL&stype=0&ext=.xml`

Ending parameter `.xml` is required as LOCKSS uses a string match to check config type.



# Birmingham Public Library

## AU Title Administration

Select one or more collections of AUs to remove, then click "Select AUs".

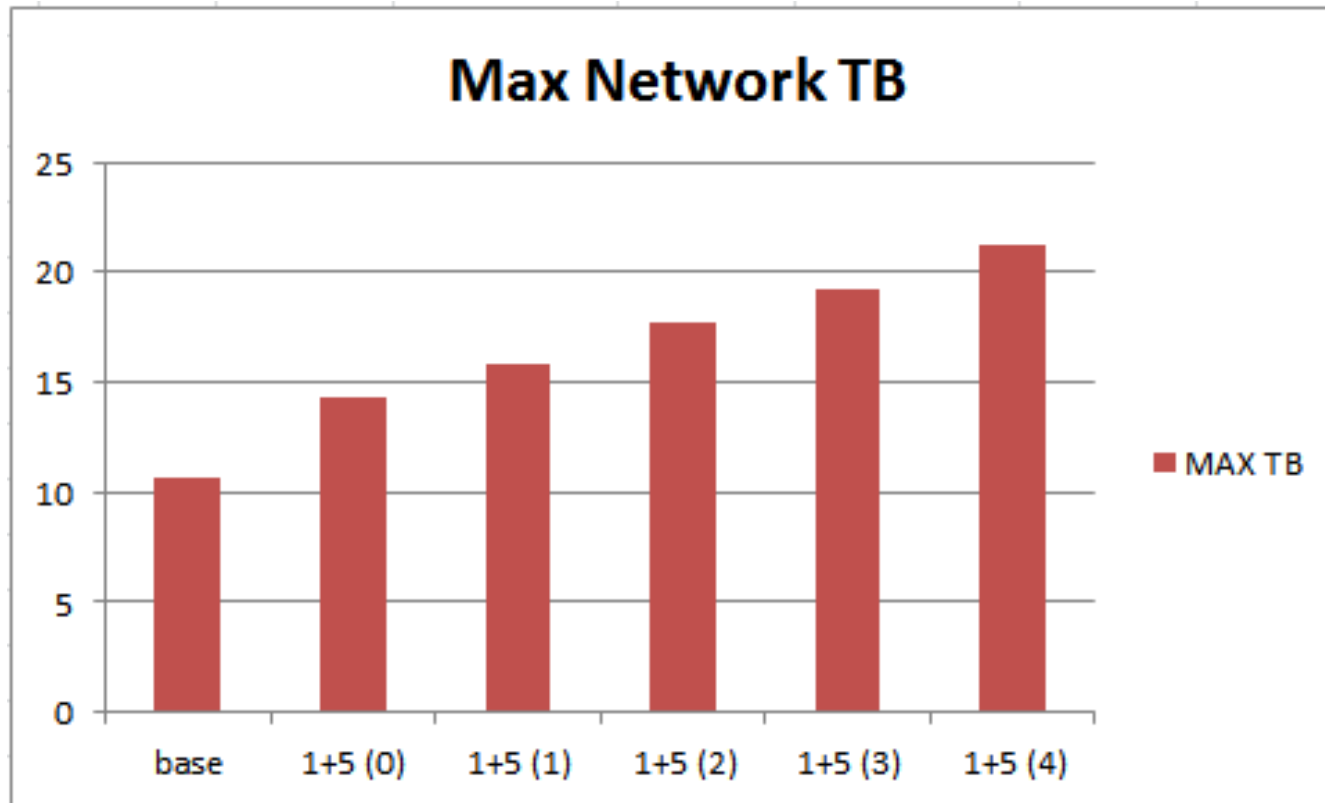
- All Alabama Department of Archives and History AUs (1)
- All Auburn University AUs (83)
- All Birmingham Public Library AUs (11)
- All Huntsville-Madison County Public Library AUs (1)
- All Spring Hill College AUs (1)
- All University of Alabama at Birmingham AUs (2)
- All University of Alabama AUs (791)
- All University of North Alabama AUs (2)
- AUs that have been approved for removal (470)

Select AUs

[Back to Journal Configuration](#)

# Birmingham Public Library

## Storage Capacity Max in Network



based on average cost reduction

Questions?