

adaptTo()

APACHE SLING & FRIENDS TECH MEETUP
BERLIN, 26-28 SEPTEMBER 2012

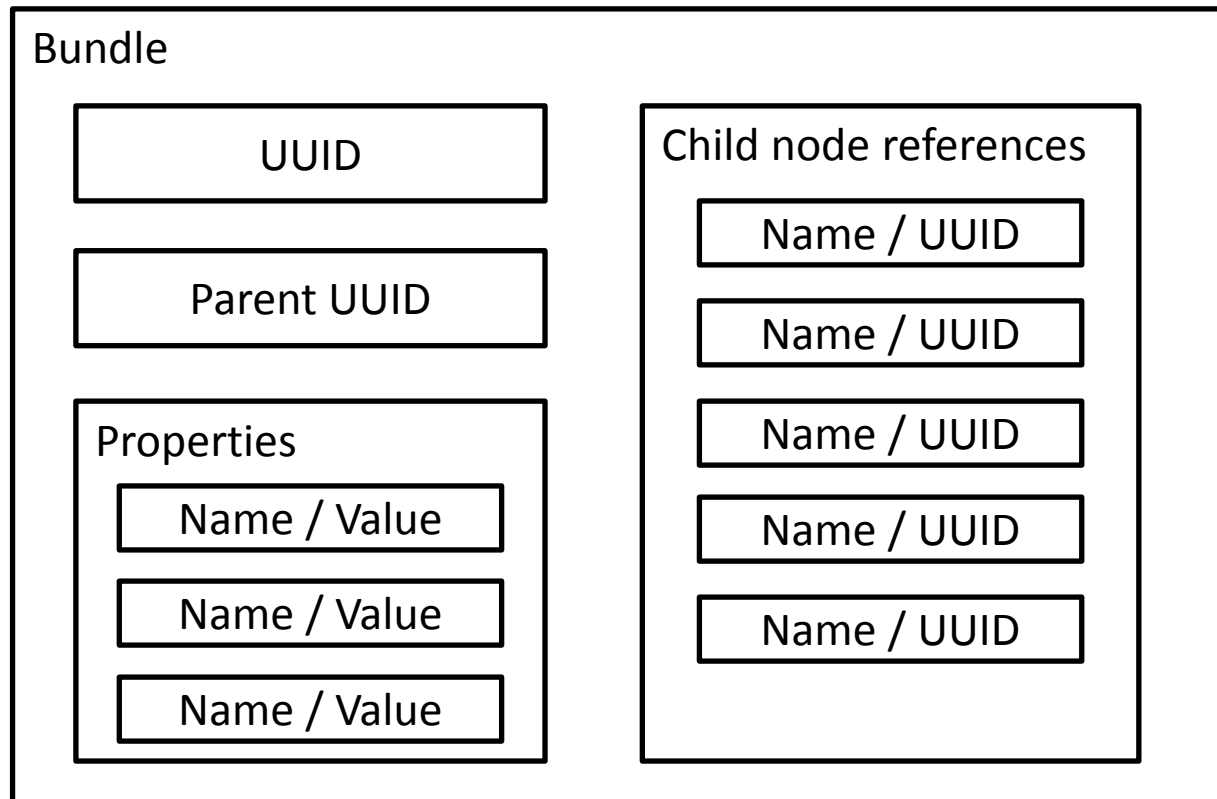
Efficient content structures and queries in CRX

Marcel Reutegger

- Jackrabbit & CRX basics
- Efficient content structures and limitations of current implementation
- Query performance analysis and optimization

- Nodes & properties stored in one entity -> bundle
- Every node/bundle has a UUID (random)
- Child nodes are linked from the parent node
- Binaries go into the DataStore

■ Bundle structure

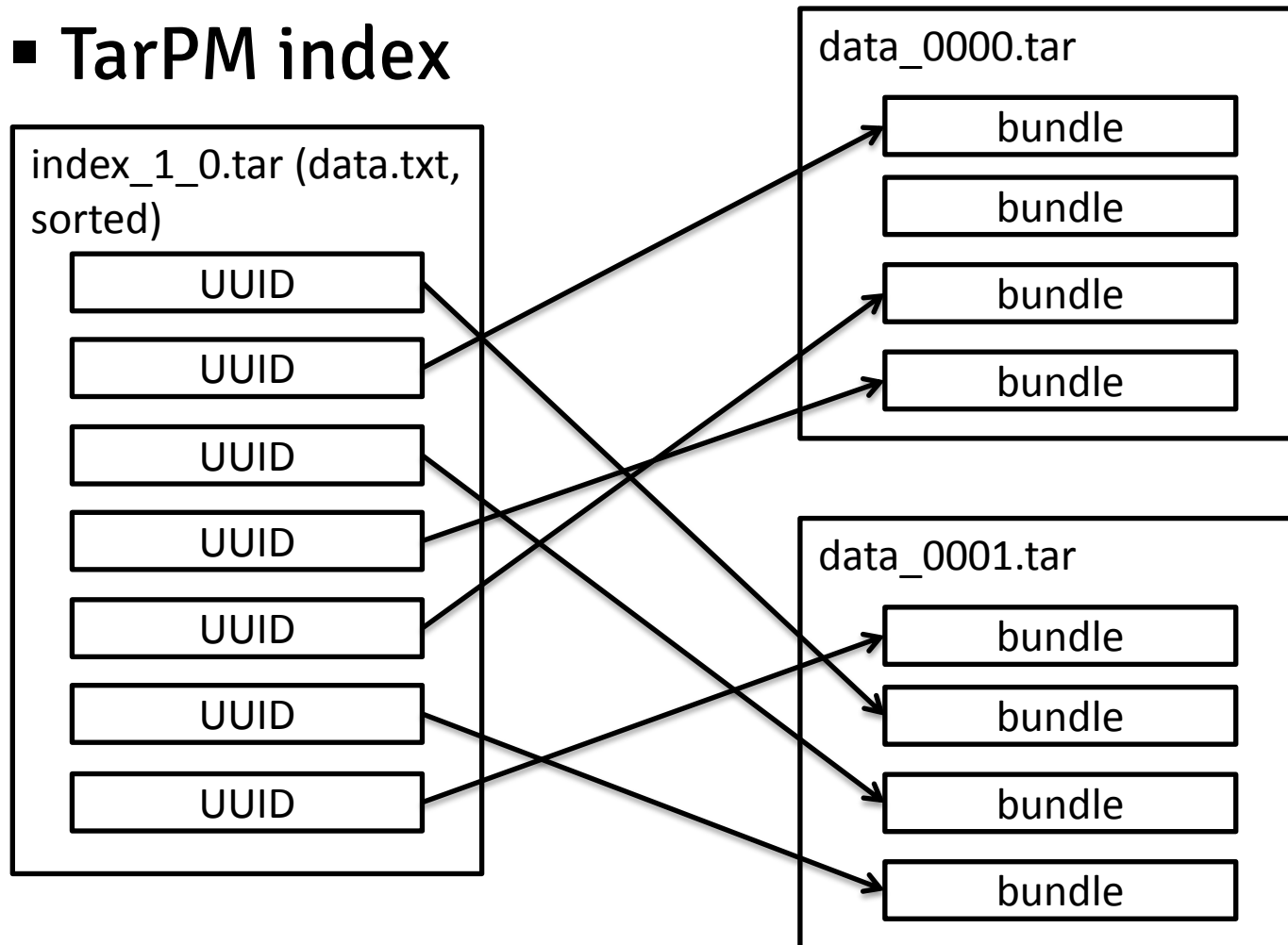


Jackrabbit basics

- Binaries go into the DataStore
- Size threshold $\geq 4\text{kB}$, otherwise inlined in bundle
- Content addressable storage, hash of content identifies binary
- DataStore garbage collection
 - Cost to run is linear to the number of nodes in repository

- Nodes & Properties (bundles) stored in tar files
- Tar files are append only
- Data is never overwritten
- Garbage is removed by TarPM optimization (scheduled, incremental)

■ TarPM index

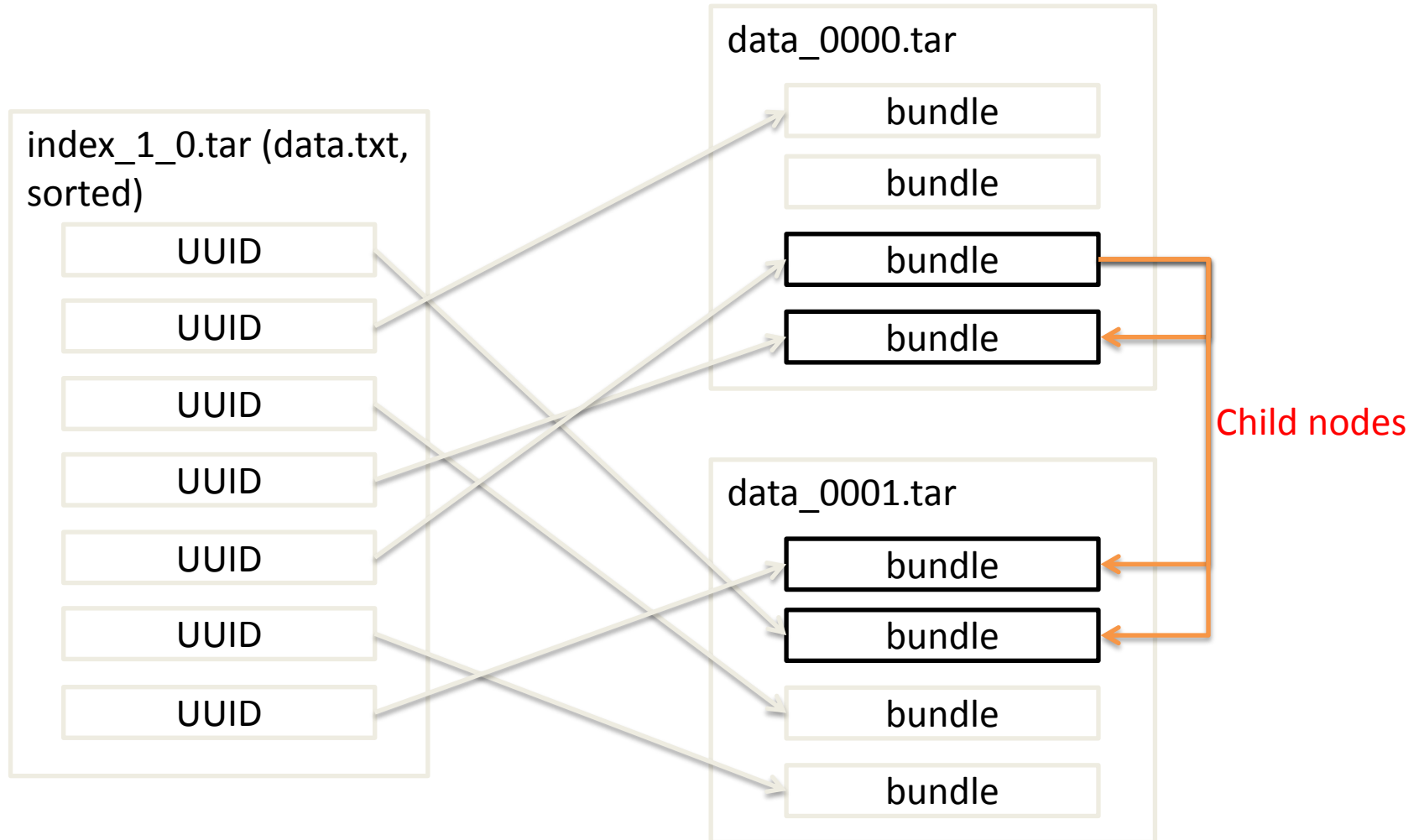


Number of nodes

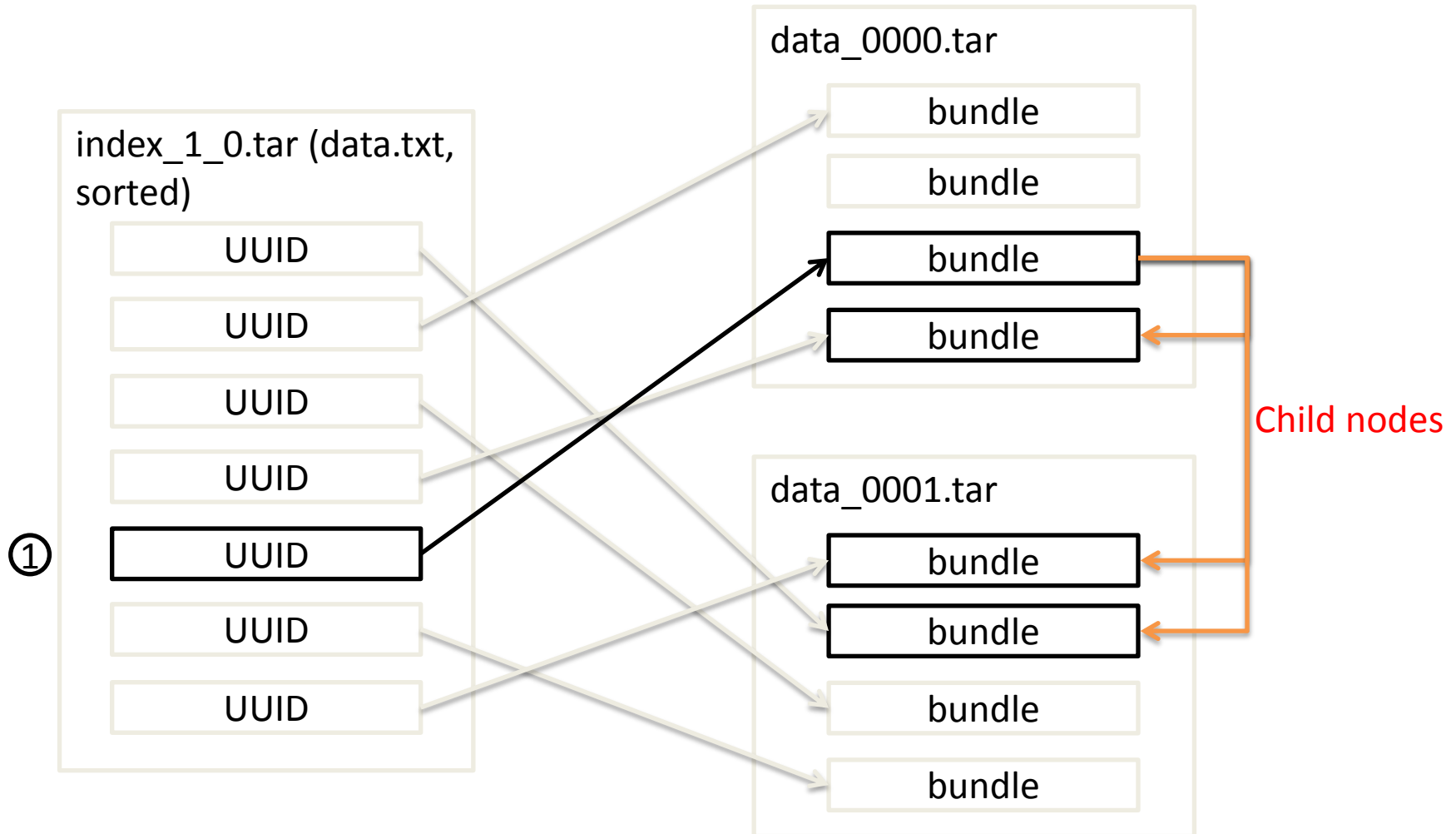
Number of nodes

- Keep number of nodes low
- Performance degrades with increasing number of nodes
- Random UUIDs cause random I/O -> Jackrabbit design
 - 15k rpm drive: 200-400 IOPS
- What about locality in data tar files?

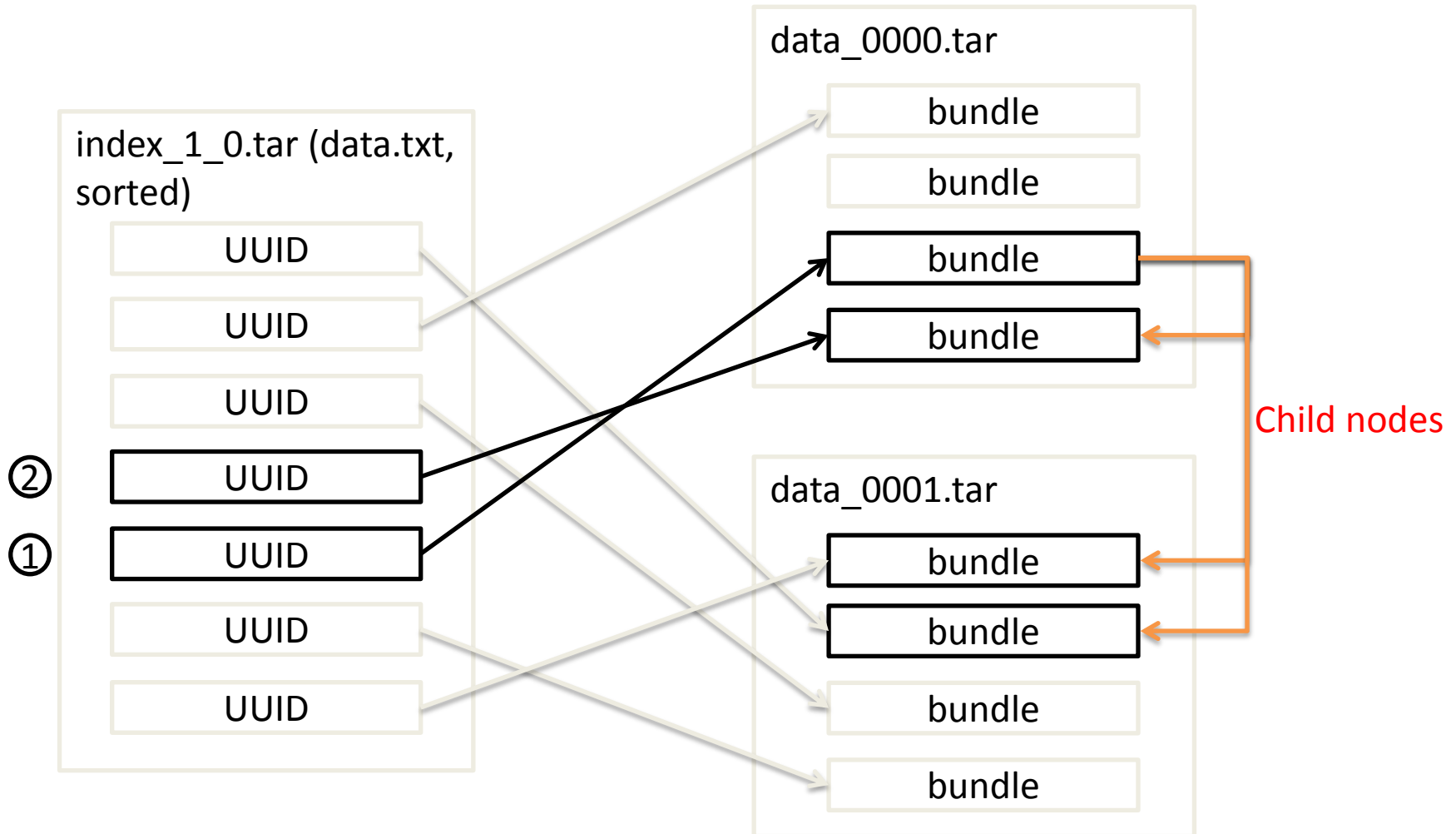
Number of nodes



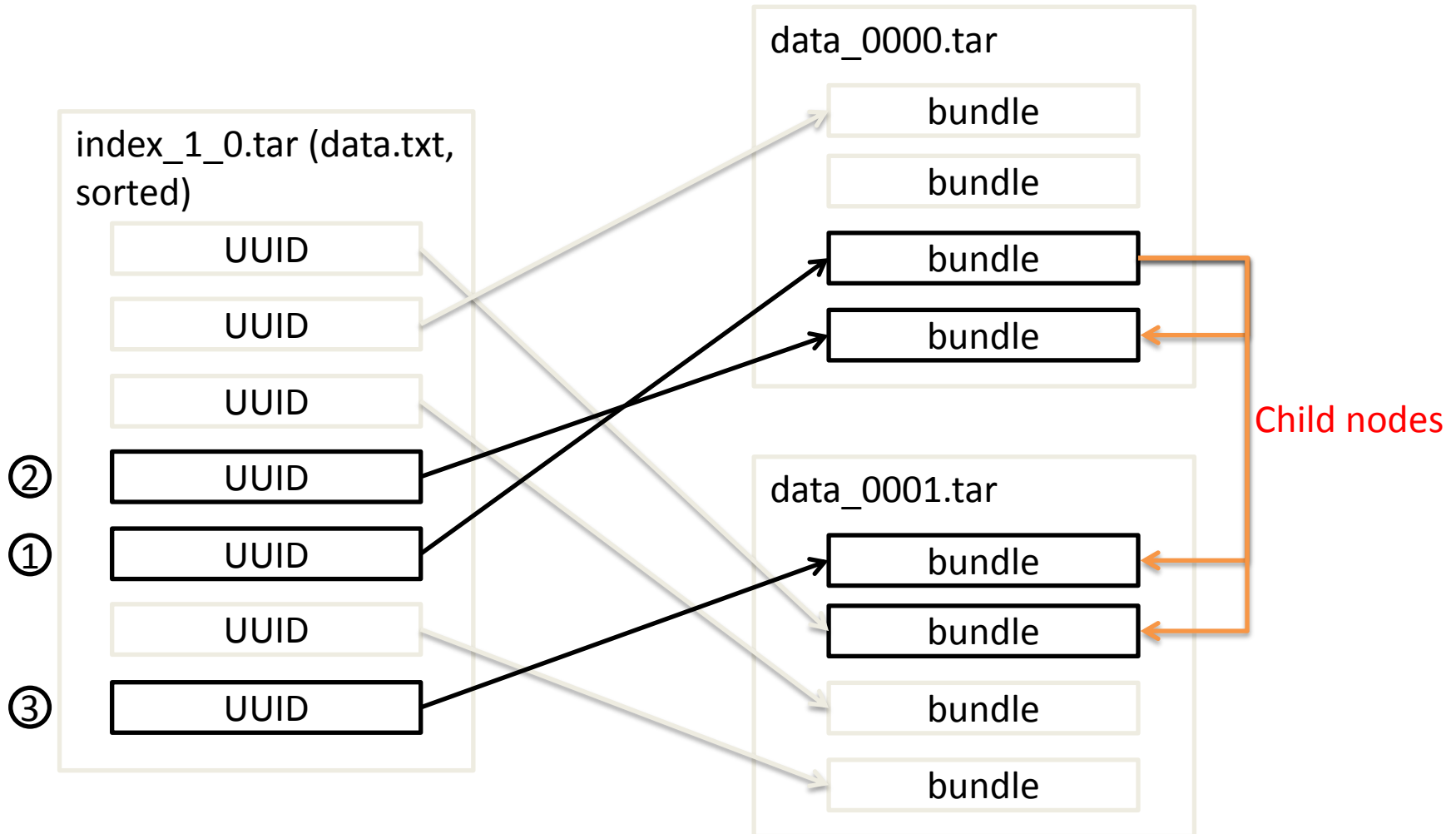
Number of nodes



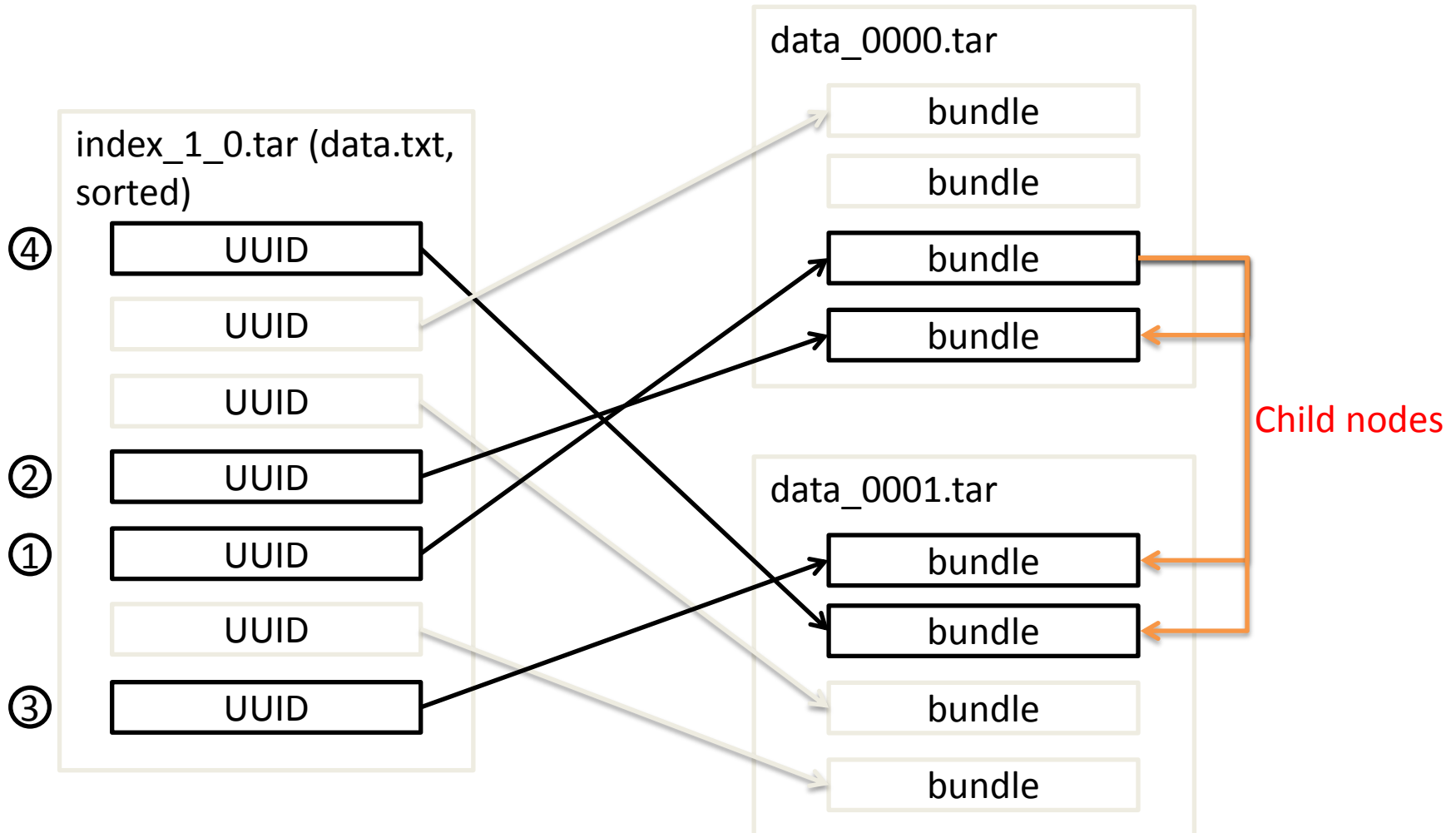
Number of nodes



Number of nodes



Number of nodes



Number of nodes

- What about OS buffer cache?
 - Cache is filled on demand
 - Only helps to some degree

- Tar index file sizes (64 bytes per bundle)
 - 1 million nodes: 70 MB
 - 10 million nodes: 700 MB
 - 100 million nodes: 7 GB

Number of nodes

- How to reduce number of nodes
 - Use version purge tool
 - Remove archived workflow instances
 - Purge audit events
- Application specific
 - Bad: document view 'import' of XML
 - Good: Pack properties on few nodes
- Other benefits: DataStore GC will be faster

Number of nodes

- Other options:
- Solid state drive (~100k IOPS)
- Force OS to cache TarPM index files

Number of child nodes

Number of child nodes

- Frequently asked questions:
- «What is the maximum supported number of child nodes?»
- «I have X number of child nodes. Will performance be OK?»

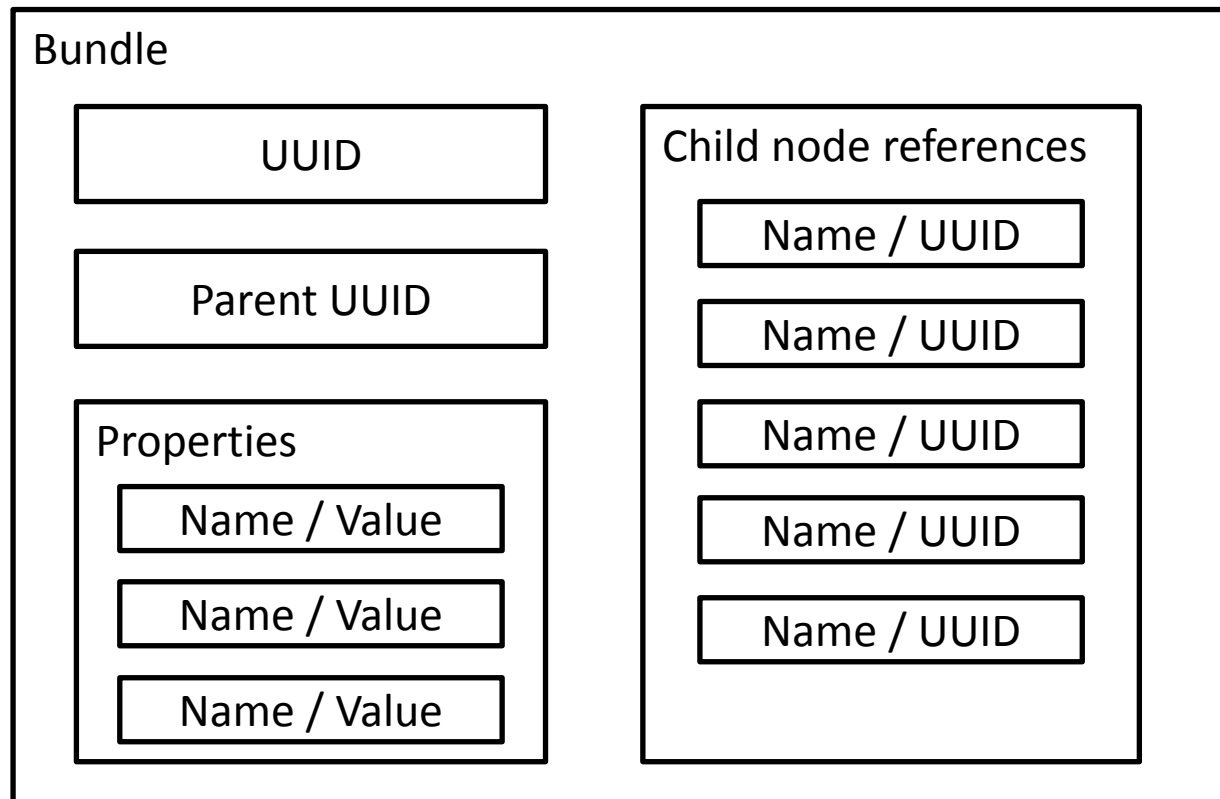
Number of child nodes

- Frequently asked questions:
- «What is the maximum supported number of child nodes?»
- «I have X number of child nodes. Will performance be OK?»

It depends!

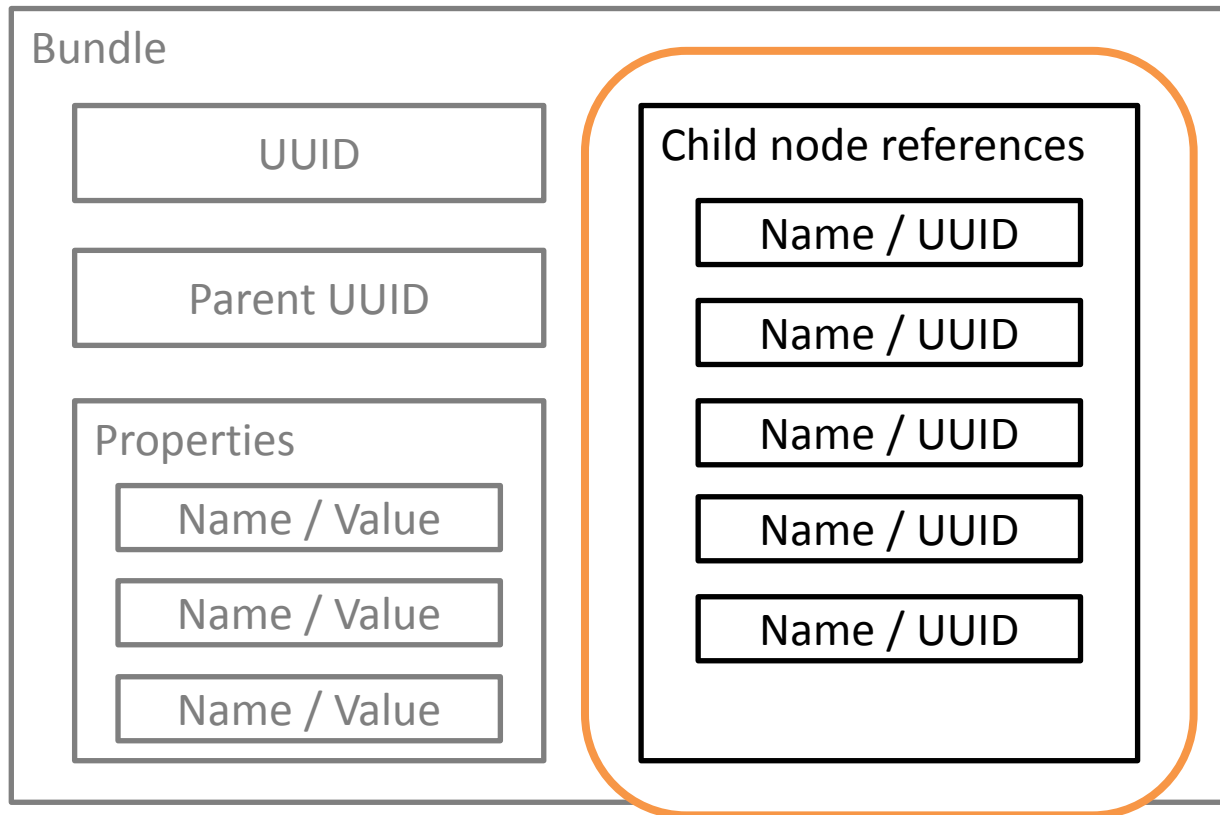
Number of child nodes

- Maximum number of child nodes



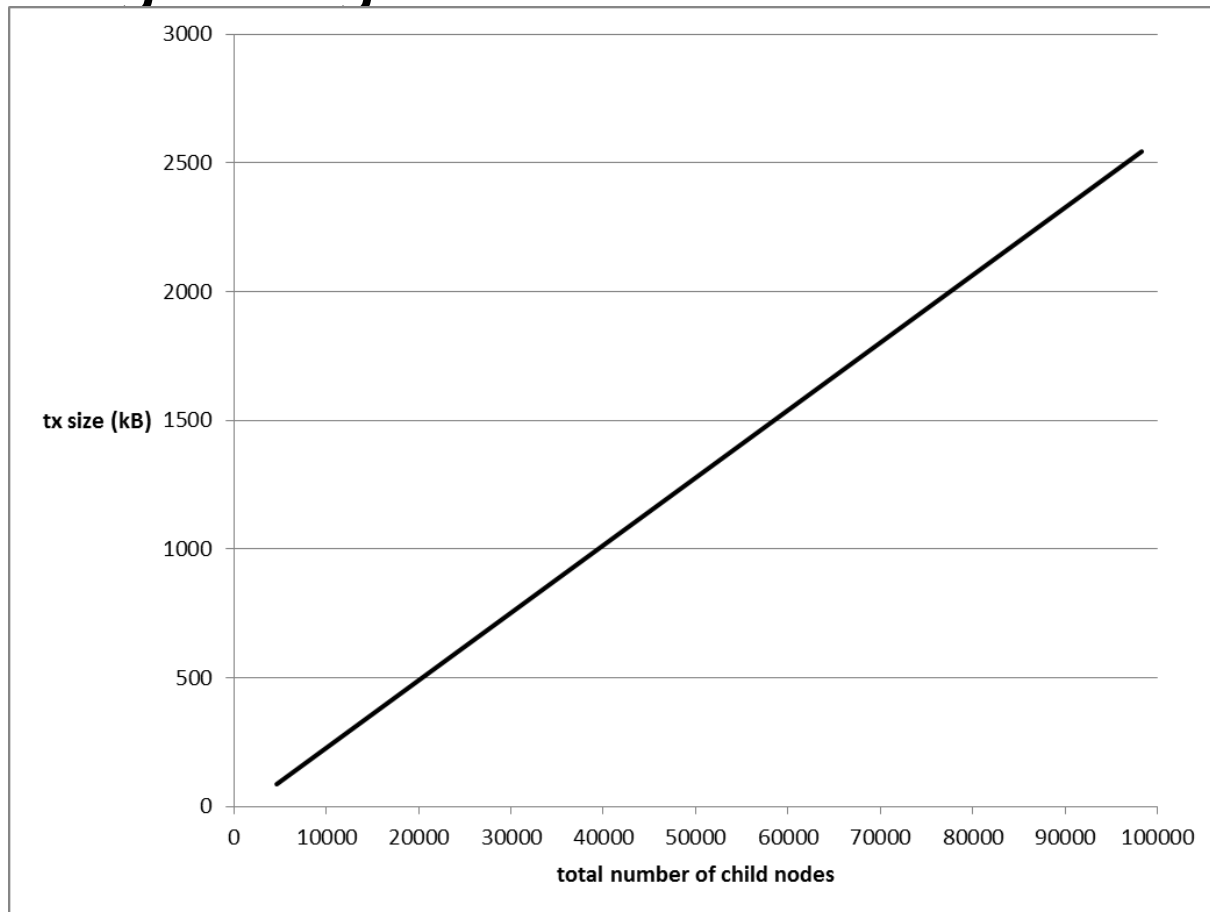
Number of child nodes

- Maximum number of child nodes



Heap is the limit

■ Adding a single child node



Number of child nodes

- Large number of child nodes
- OK for:
 - Static content
 - /libs/wcm/core/i18n/de has ~4k child nodes
- Not OK for:
 - Dynamic content
 - authentication pins, replication items, user generated content

Number of child nodes – Recommendations

- **Structure content**
 - E.g. date/time based: 2012/09/26
 - Use utilities like Jackrabbit BTreeManager
- **Make sure application keeps number of child nodes within limits (e.g. 1000)**
- **Save in batches when possible**

Number of child nodes

- What about performance?
- Usually repository growth is the major concern, but...
- Unfortunate combination of *application* and *content* design may result in bad performance

David's Model

David's Model: A guide for content modeling

- Rule #1: Data First, Structure Later. Maybe.
- Rule #2: Drive the content hierarchy, don't let it happen.
- Rule #3: Workspaces are for clone(), merge() and update().
- Rule #4: Beware of Same Name Siblings.
- Rule #5: References considered harmful.
- Rule #6: Files are Files are Files.
- Rule #7: ID's are evil.

- **Avoid features not used in CQ**
 - XA transactions
 - Shareable nodes
 - Lifecycle Management
 - Retention and Hold

Query analysis

- Query debug log
 - <http://dev.day.com/kb/home/Crx/Troubleshooting/HowToDebugJCRQueries.html>
 - “executed in <time> ms. (<query>)”
- JMX (CQ 5.5)
 - QueryStat: slow and most frequent queries
 - TimeSeries: count, duration, average

- Fast: simple comparison
 - **`sling:resourceType = 'my/type'`**
- Fast: node type match
 - **`//element(*, nt:hierarchyNode)`**
- Fast: simple fulltext search
 - **`jcr:contains(@jcr:title, 'crx')`**
- Fast: like on few distinct values
 - **`jcr:like(@jcr:mimeType, '%/plain')`**

- Slower: path constraints
 - **`content/geometrixx/en//*[...]`**
 - Alternative: turn path into property constraint. E.g. keep language property on every page and write:
`//*[@language = 'en']`
- Slower: relative path in predicate
 - **`//element(*, cq:Page)[jcr:contains(jcr:content, 'crx')]`**
 - Alternative: shorten path in predicate and post process result:
`//element(*, cq:PageContent)[jcr:contains(., 'crx')]`

- Slower: `jcr:contains` with wildcards
 - `jcr:contains(., 'sing*')`
 - Alternative: Implement Lucene analyzer with appropriate stemmer

- Slow: `jcr:contains` with initial wildcard
 - `jcr:contains(., '*rabbit')`
 - Alternative: don't do it, unless you know exactly what you are doing!

- Slow: `jcr:like` on many distinct values
 - `jcr:like(@email, '%@gmail.com')`
 - Alternative: store data you want to query in separate property, then you can write: `@email-host = 'gmail.com'`

- Slow: ranges matching many distinct values
 - **@jcr:lastModified > xs:dateTime('2001-09-17T18:17:13.000+02:00')**
 - Alternative: reduce resolution (e.g. only store date and not time)

- Query result does lazy loading of nodes
- `Query.execute()` may return quickly even if result size is big
- Looping over `QueryResult.getNodes()/getRows()` will load data from TarPM
- Reading a large result set completely is always slow
- Time to get result: query execution time + node retrieval time

- Recommendations
 - Test with real content
 - Structure content to avoid queries
 - Denormalize
 - Avoid path constraints
 - Replace frequent queries with initial query + event listener

Thank you