

Repair-driven file system design

Val Henson

val_henson@linux.intel.com

Open Source Technology Center

Intel Corporation

Will fsck time grow as capacity
grows?

Disk hardware trends

- Between 2006 and 2013:

Capacity increases by 16x

Bandwidth increases by 5x

Seek time stays nearly flat

One sample fsck time change

Fsck time on my laptop /home using ext3, scaled to 2013 disk size, bandwidth, rotational latency, and seek time:

2006: 450 s (cpu 15 s, data transfer 42 s,
seek/latency 390 s)

2013: 4800 s (cpu ? s, data transfer 130 s,
seek/latency 4700 s)

7.5 minutes vs. 80 minutes = **10x** increase!

Fsck frequency

Are sources of file system inconsistency growing faster than improvements in consistency?

Sources of file system inconsistency

- Per-**disk** I/O error rate
- Administrator error
- Failure of layers below block device interface
- File system bugs

fsck time crunch: fsck takes longer
and happens more often

Existing solutions reduce likelihood of fsck, not duration of fsck

- Journaling, copy-on-write, soft updates, all designed to complete interrupted file system updates
- Checksums, RAID, replication make fsck less common but still happens (and may cost days)
- In face of I/O error or file system bug, the entire file system must be checked with fsck

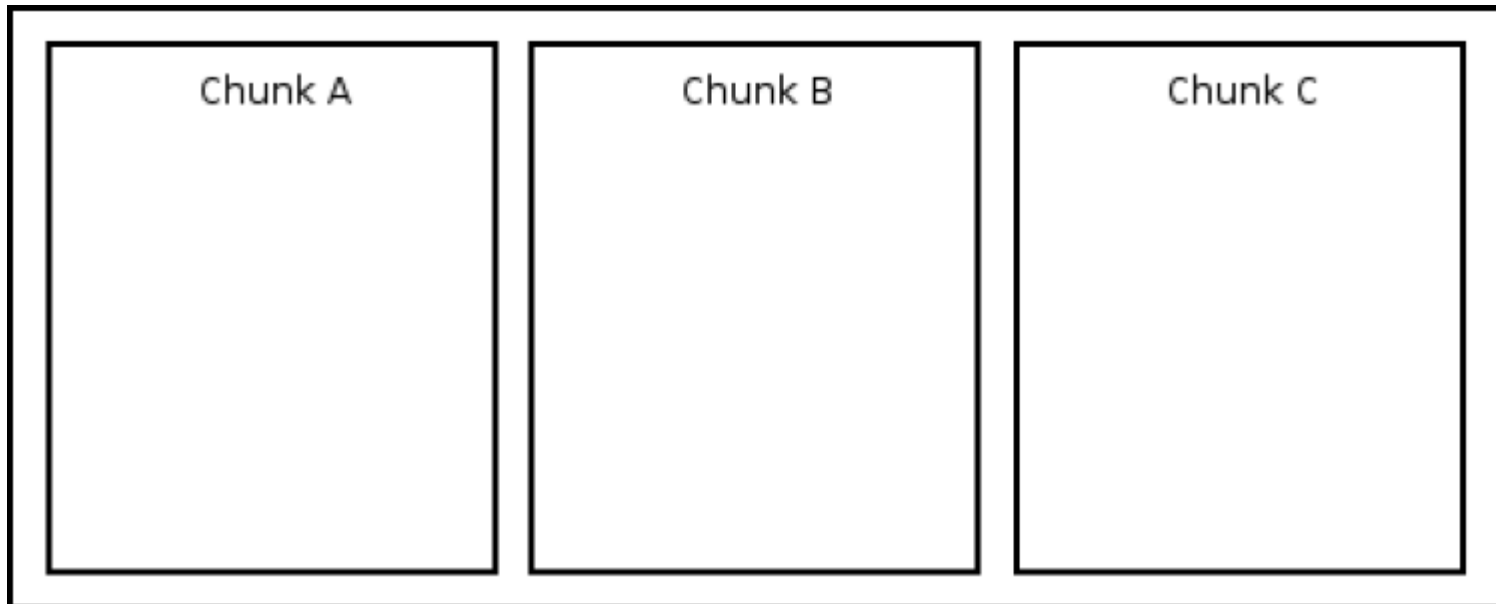
Repair-driven file system design

Repair-driven file system design

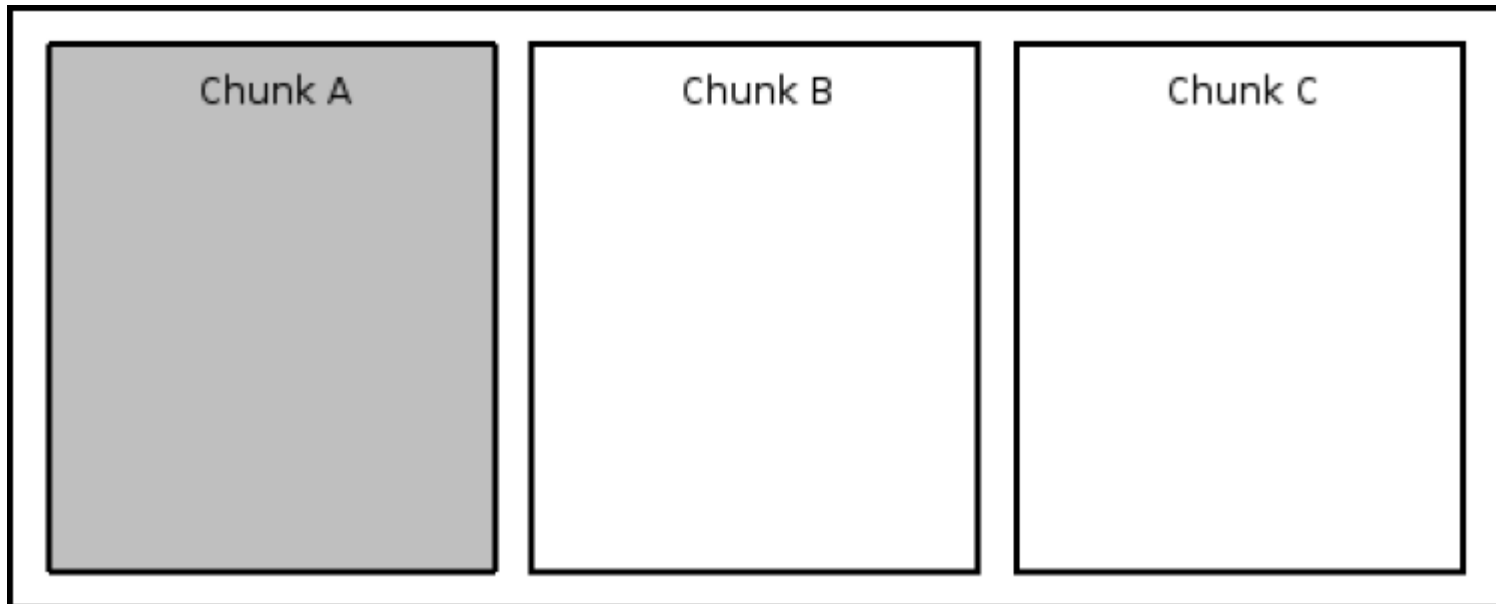
- On-disk format designed with repair in mind
- Simple data structures
- Optimizations for reading data for repair (e.g., metadata bitmap)
- Fast, incremental file system check
- Checksums, redundancy, scrubbing, etc.
- Metadata isolation

Chunkfs

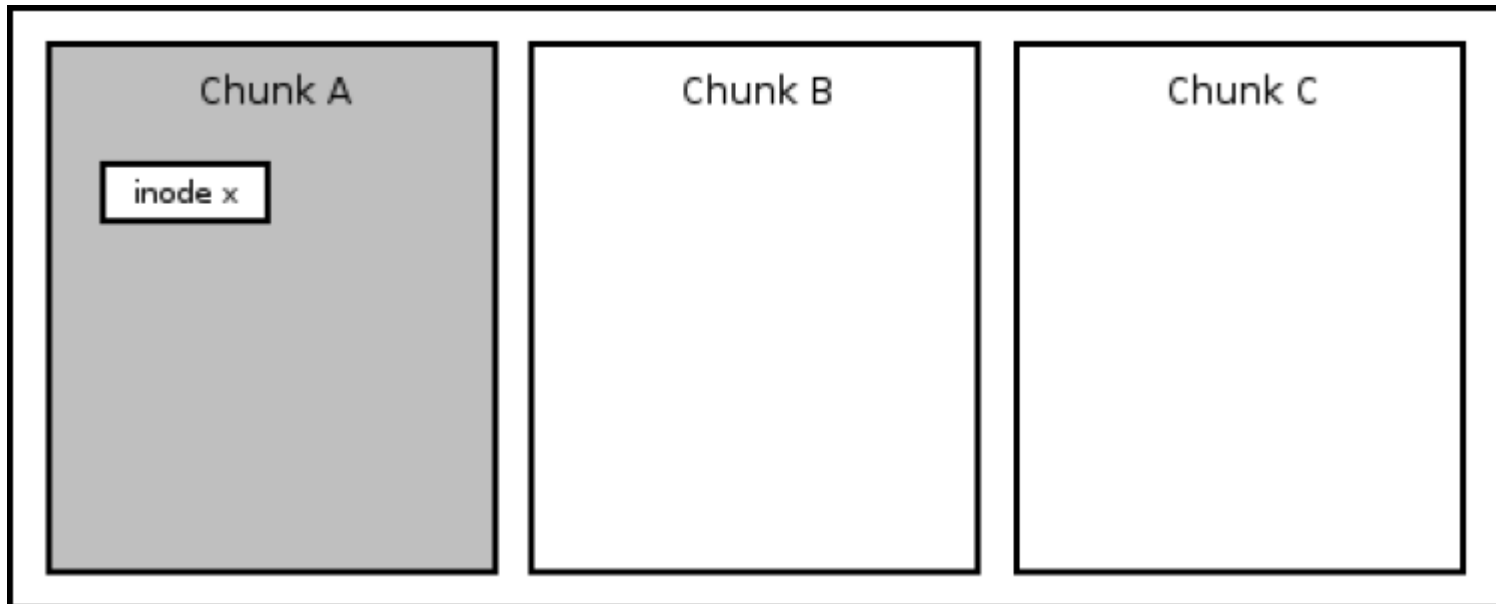
Divide fs into chunks



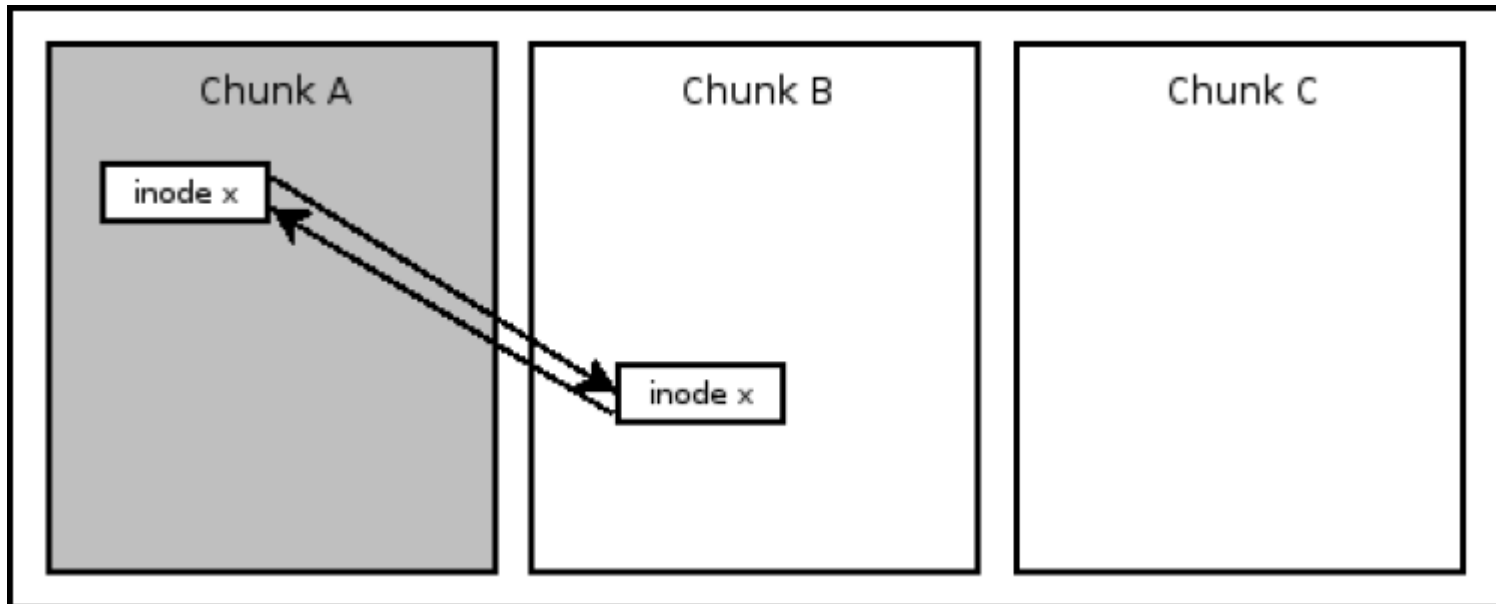
How to glue it back together?



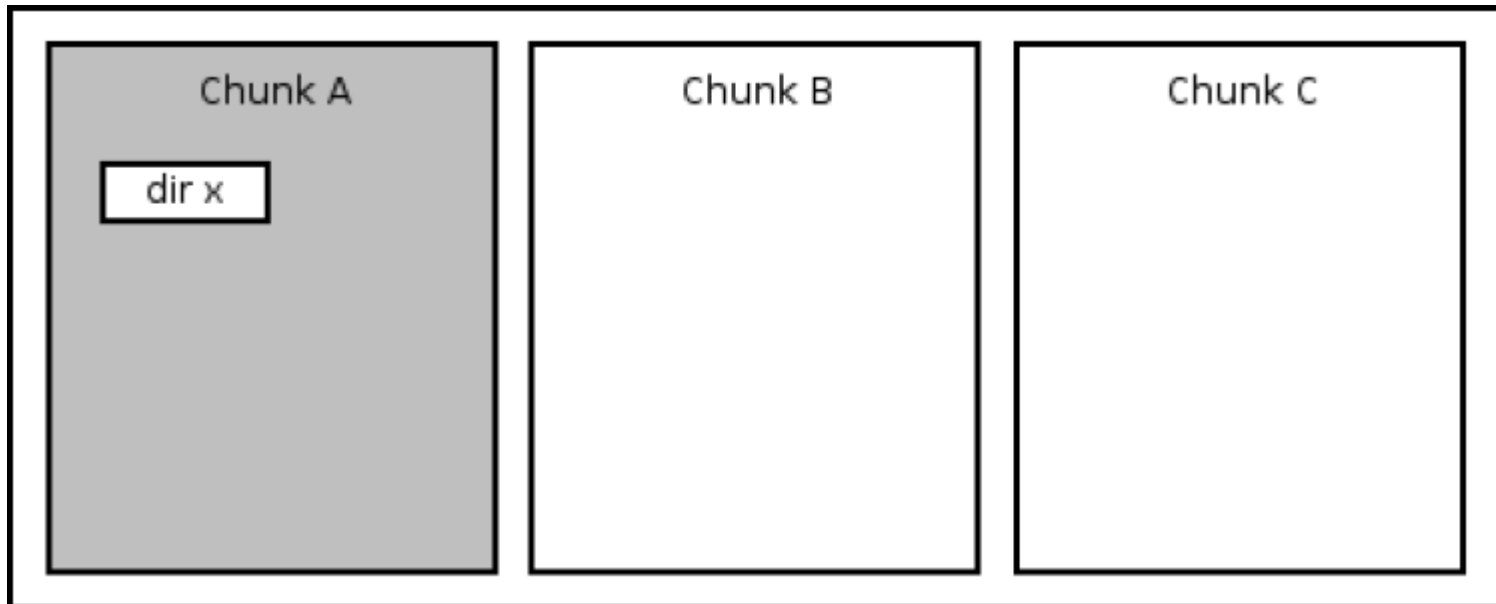
Problem: File data outgrows chunk



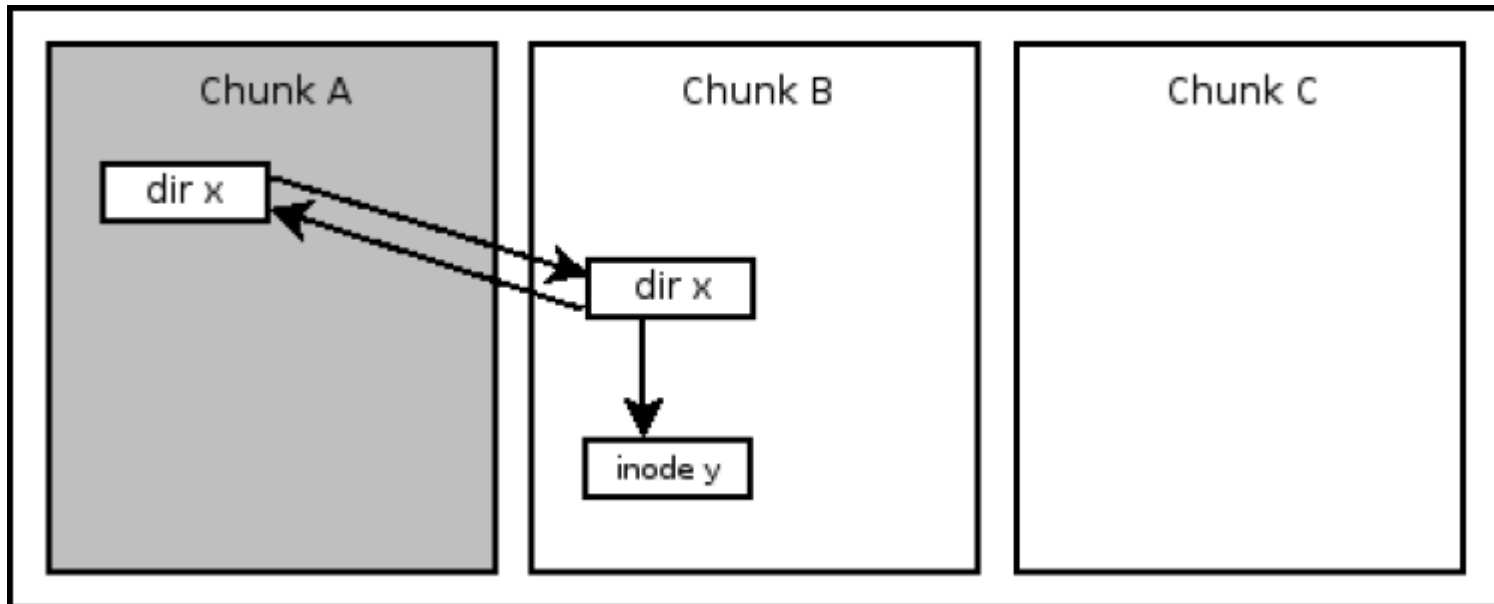
Solution: Continuation inodes



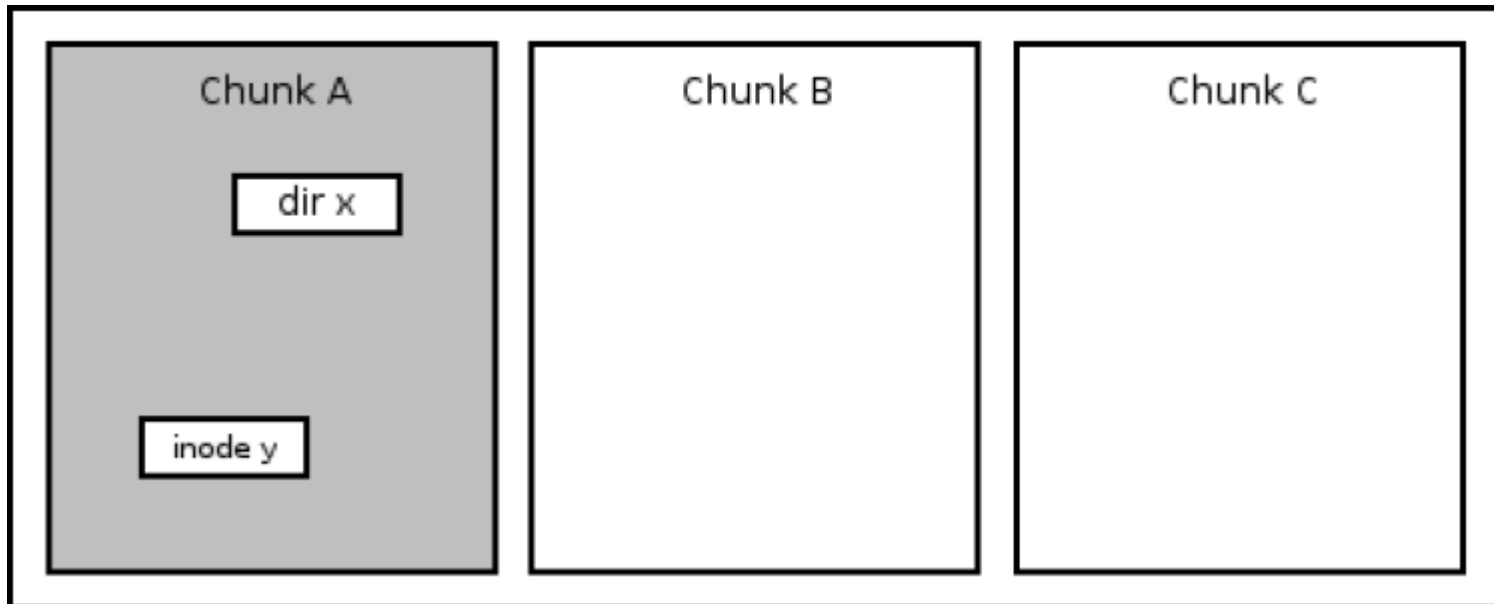
Problem: Directory outgrows chunk



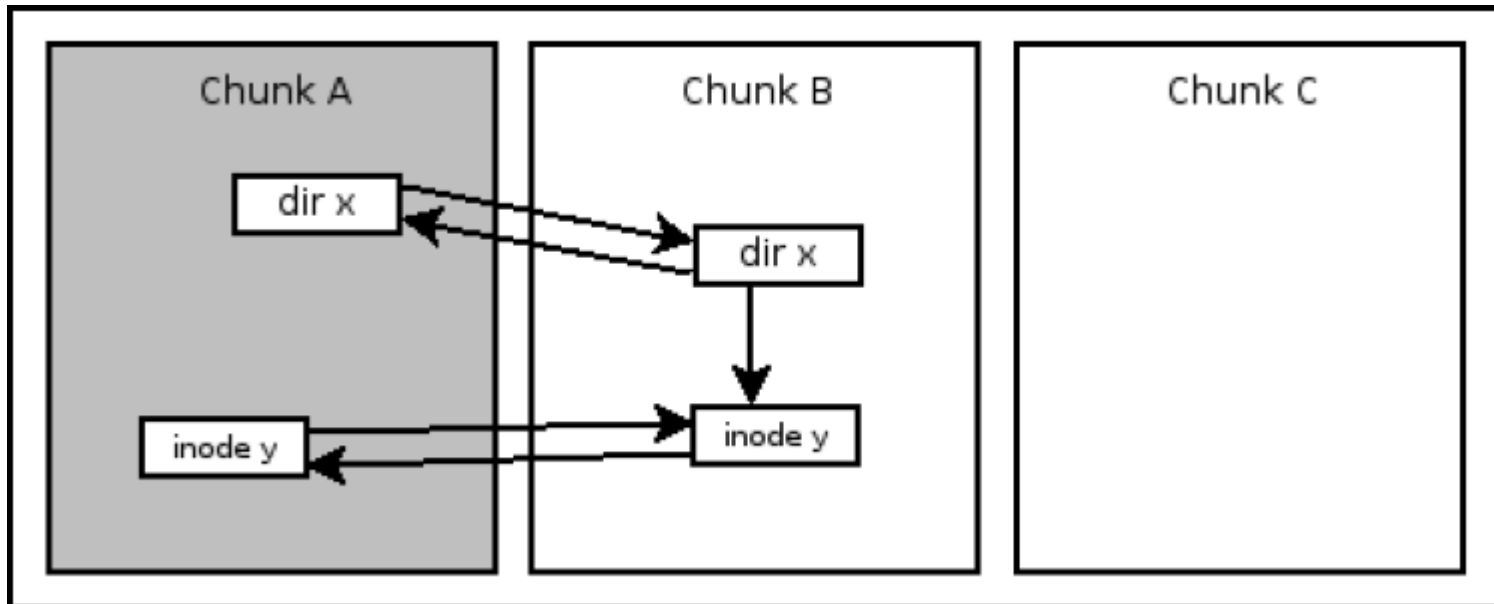
Solution: Continuation inodes



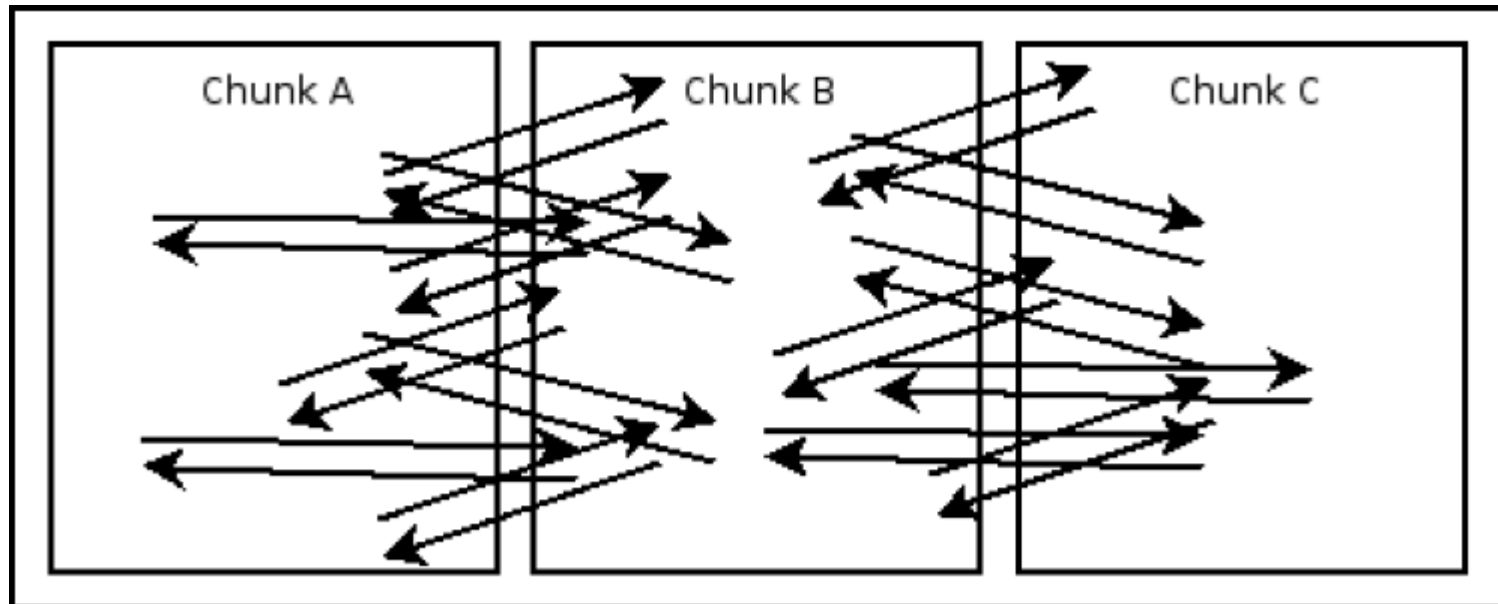
Problem: Hard link to inode in full chunk



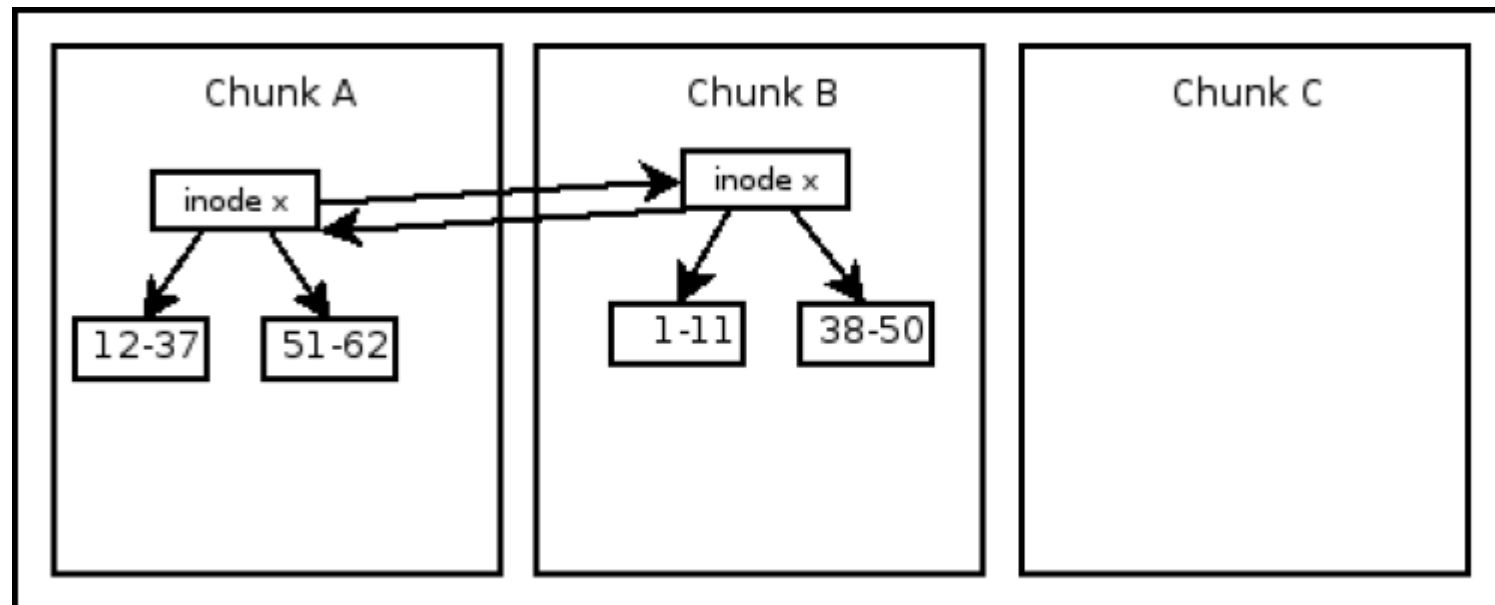
Solution: Continuation inodes



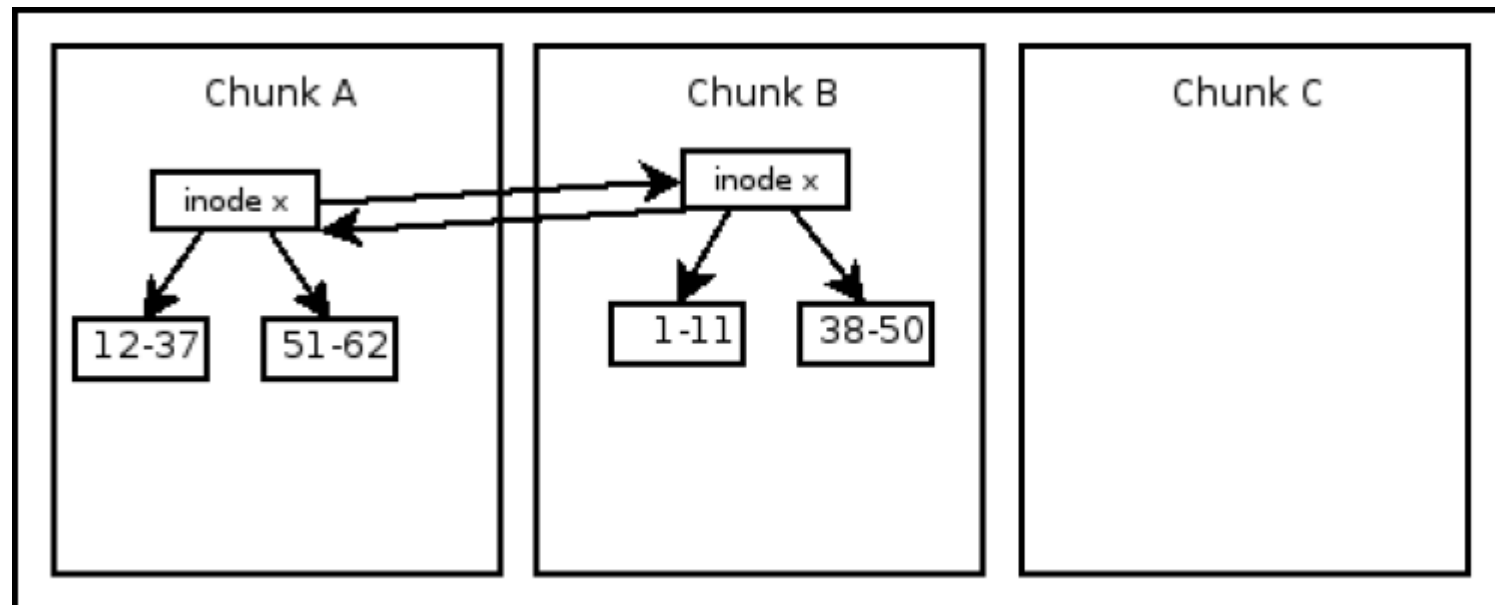
Problem: Too many continuation inodes



Solution: Smart allocation & sparse files



Problem: Quickly finding file offset



Solution: Embedded lookup structure in continuation inode

```
struct chunkfs_inode {  
    ...  
    struct continuation_data;  
}
```

```
struct continuation_data {  
    ...  
    struct tree_node;  
}
```

Neat stuff about chunkfs

- Built-in multi-threaded scalability
- Incremental, on-line check and repair
- Shrink, grow, defrag easier
- Per-chunk formats
- Crash-only software ethos