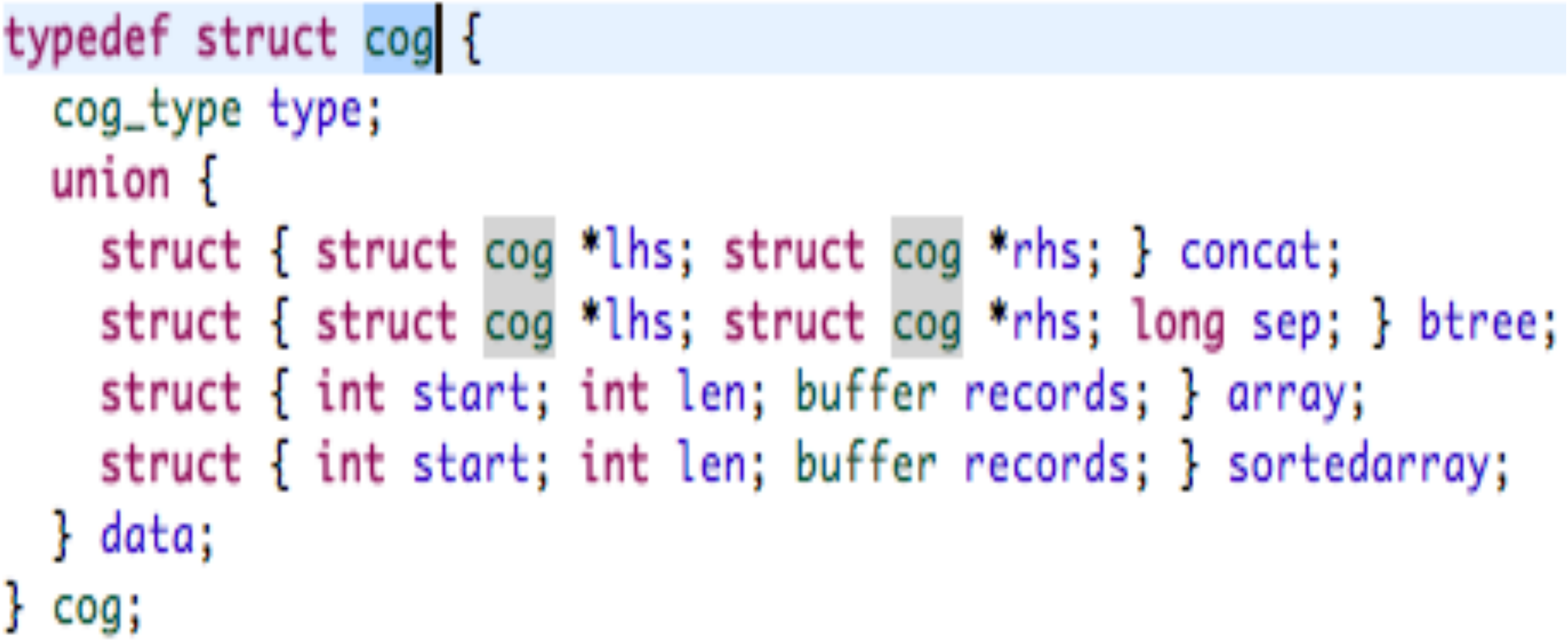
JITDs Java and C Comparison

Teams Twinkle & Datum

Cogs in C

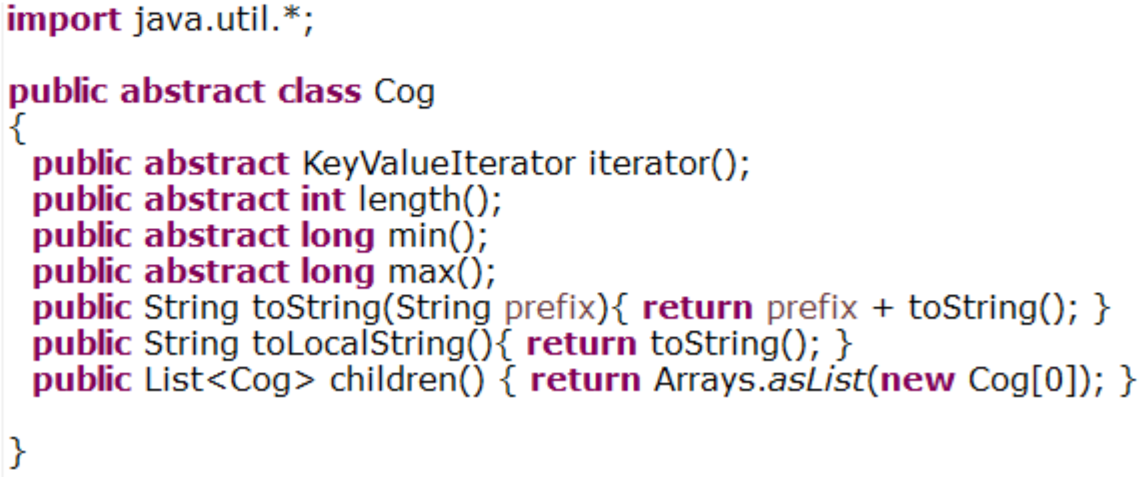


```
typedef struct cog {
  cog_type type;
  union {
    struct { struct cog *lhs; struct cog *rhs; } concat;
    struct { struct cog *lhs; struct cog *rhs; long sep; } btree;
    struct { int start; int len; buffer records; } array;
    struct { int start; int len; buffer records; } sortedarray;
  } data;
} cog;
```

Cogs in C

- Basic struct which abstracts cogs based on type
- Acts sort of as an interface
- Each type of cog has its own basic structure
- Operations on cogs provided in header files
  - Cracking
  - Adaptive Merge
  - Related Operations/Etc.
- Memory management built into basic functions

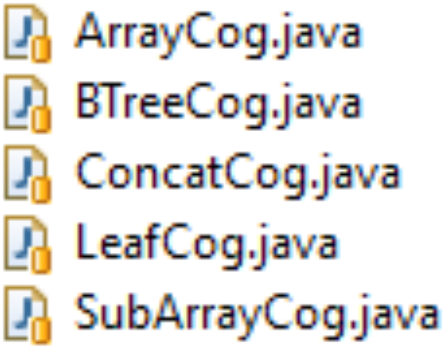
Cogs in Java



```
import java.util.*;

public abstract class Cog
{
  public abstract KeyValueIterator iterator();
  public abstract int length();
  public abstract long min();
  public abstract long max();
  public String toString(String prefix){ return prefix + toString(); }
  public String toLocalString(){ return toString(); }
  public List<Cog> children() { return Arrays.asList(new Cog[0]); }

}
```



- ArrayCog.java
- BTreeCog.java
- ConcatCog.java
- LeafCog.java
- SubArrayCog.java

Data in C

- Operation on data similarly provided in header files
  - Sort
  - Search
  - Min/Max
  - Iteration
  - Concatenate
  - Related Operations/Etc.
- Memory management built into basic functions

Delving deeper into C Data & Memory Management

- Main memory allocation happens on buffer creation/deletion
- There is some memory management done for helper structs such as iterators

- What is a buffer?

  - contains an array of data records [key/value pairs]

  - holds a reference count to itself

  - also contains a size value (number of records)

Cracking and Merging C vs Java Comparison

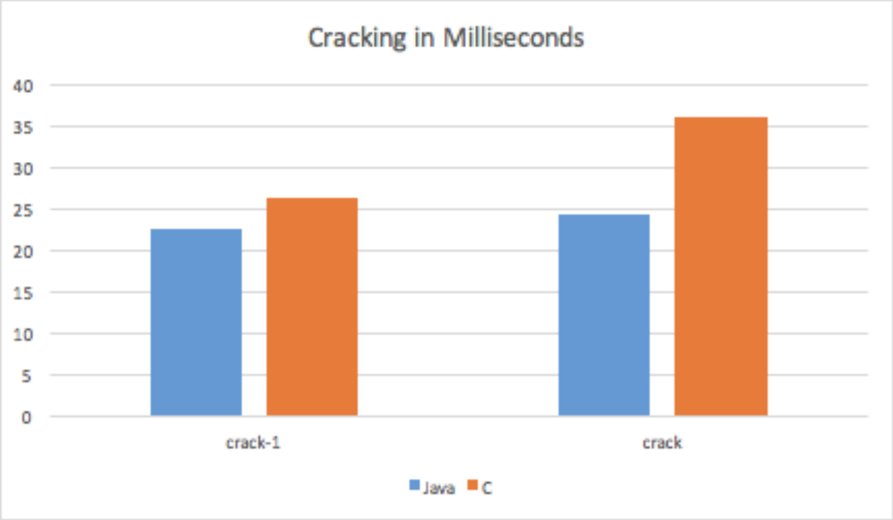
- Standard algorithm are used
- The implementations of cracking and merging is basically the same
- The only difference is that C uses structs and pointers
- C is proactive in memory management through the use of malloc and free
- Java uses the garbage collector for memory management

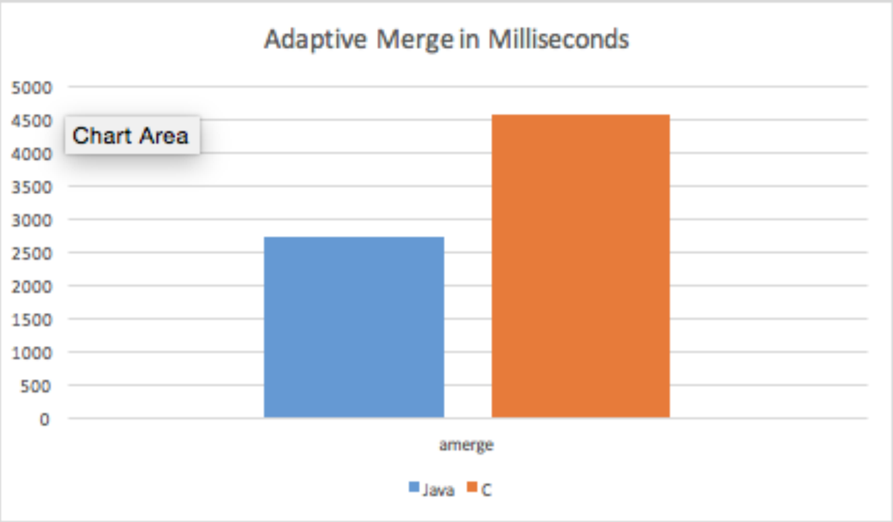
Single Read Comparison on Java vs C

All tested 1000000 data key range 1000000 crack-1 single split at 500000 and crack and amerge are single split at 333333 and 666666:

	Java	C
crack-1	22.4407 ms	26.276 ms
crack	24.426775 ms	36.057 ms
merge	2730.864164 ms	4568.007 ms

Single Read Comparison on Java vs C





Cracking - C vs Java

Tested with :

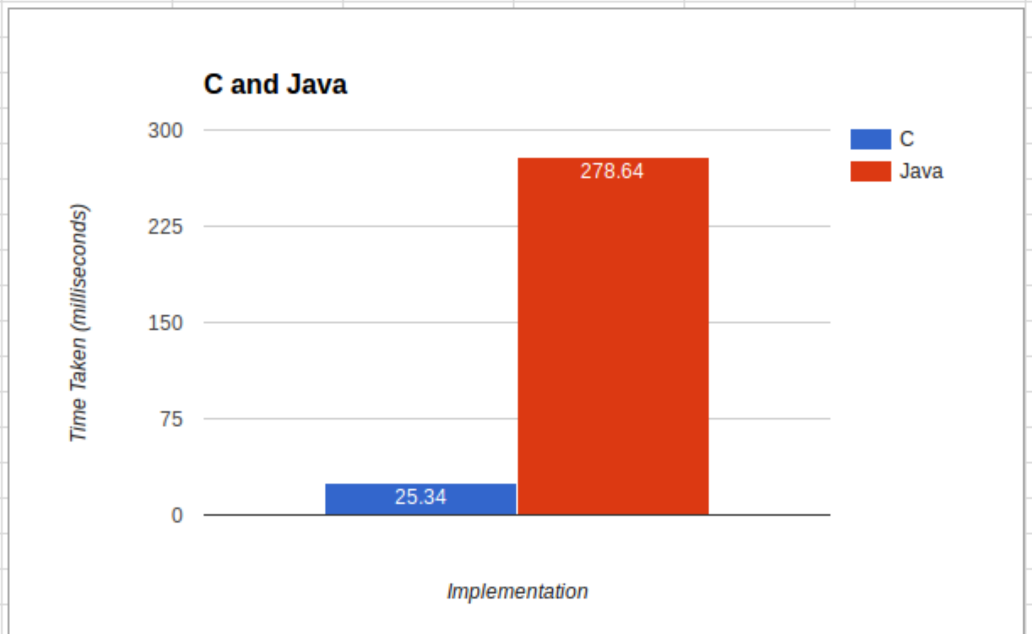
Data: 1000000,

Key Range: 1000000

Random Reads: 1000

Environment:

Ubuntu 14.04 / CPU 1.7 GHz / Processor Intel i5



Adaptive Merging

:(

Questions??

JITDs on Disk

Team Warp

Animesh, Archit, Rishabh, Rohit

CODE WRITTEN FOR DIFFERENT FILE FORMATS

Data , Separator, Data

Data,2,Data	Null,5,Null	Data,6,Data

File Pointer, Separator, File Pointer

File,2,File	Null,5,Null	File,6,File

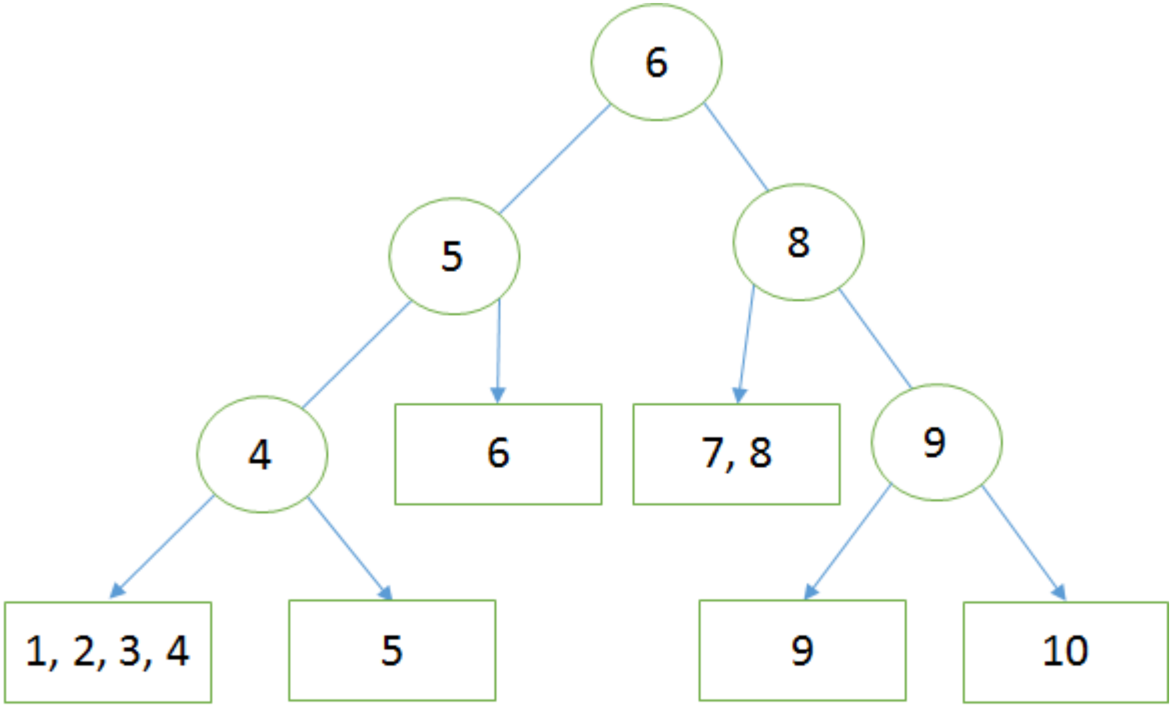
Saving and restoring trees

- Considering memory constraints, previously we were only restoring a part of the entire index tree for further indexing based on the incoming query
- Also, saving that indexed sub-tree on the disk accordingly
- This introduced problems while merging partial trees together to create updated index structure

Paging

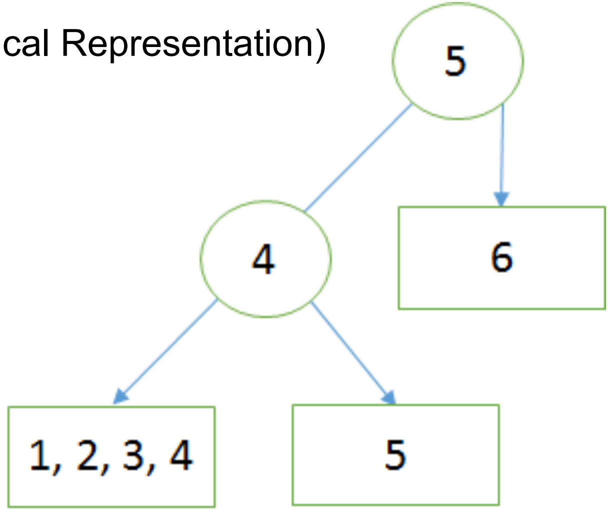
- Page-in and Page-out specific portions of the index tree based on the incoming query
- Each page will have a file structure similar to that of an index file

An Index Tree



An Index Tree With pages

P1 (Logical Representation)



File,4,File	Null,5,File

P1 (Physical Representation)

In memory logical representation