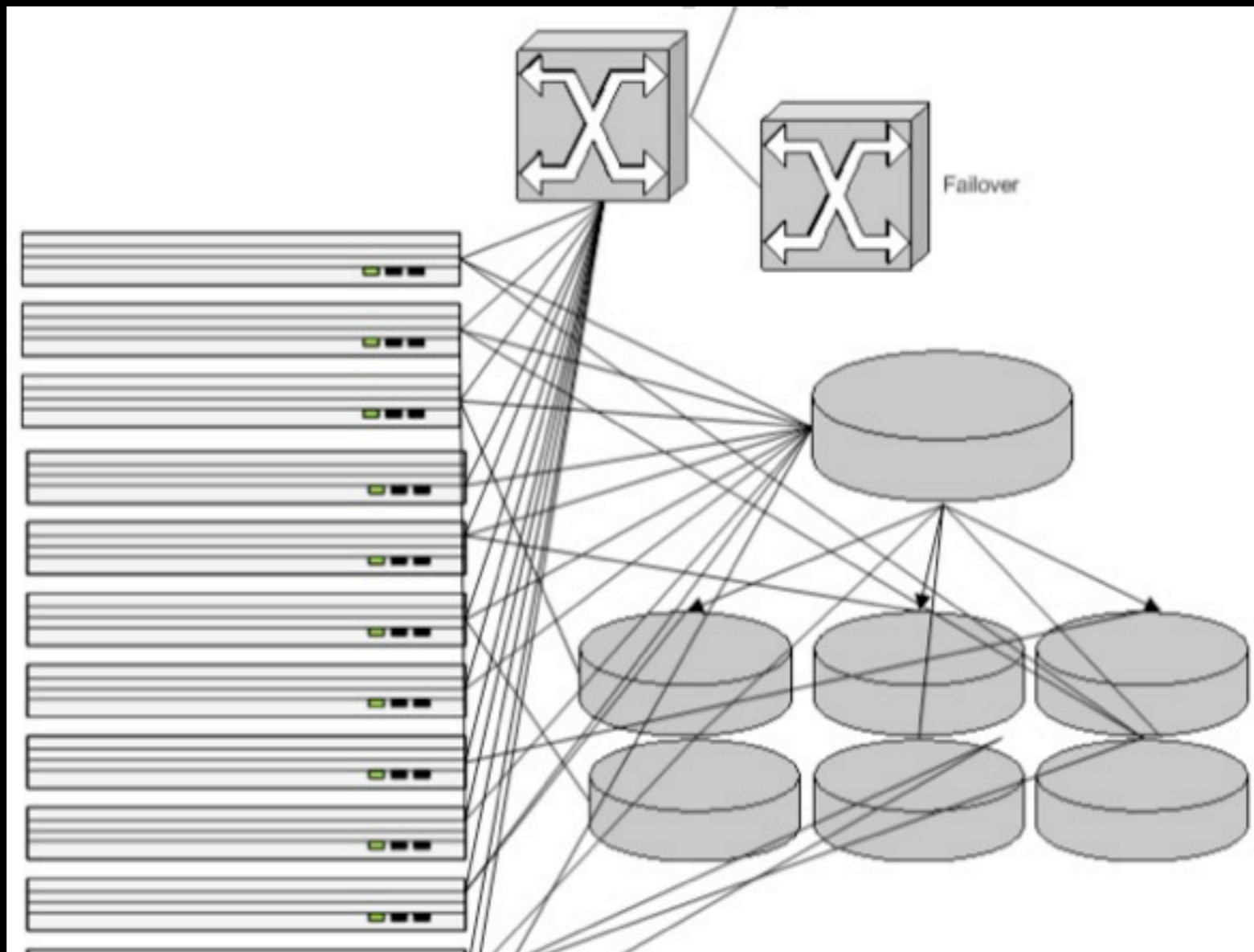


Tricking out your Memcached Setup





Chaos Factor

ASCII Protocol

```
[brian@gaz libmemcached]$ memcp foo --servers=localhost
```

```
[brian@gaz libmemcached]$ telnet localhost 11211
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.
```

```
Escape character is '^]'.  
get foo
```

```
VALUE foo 0 4
```

```
bar
```

```
END
```

```
version
```

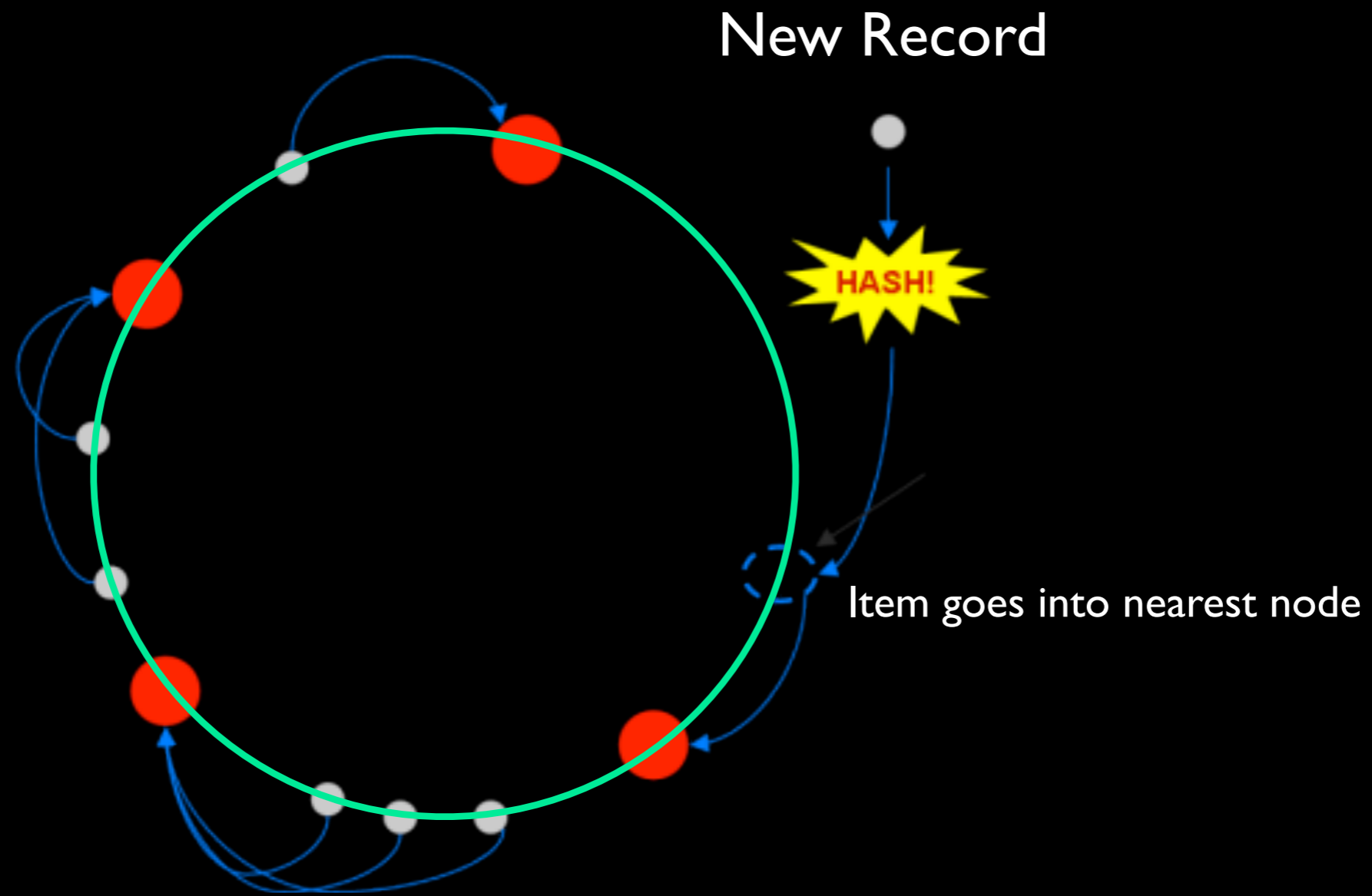
```
VERSION 1.4.5
```

Binary Protocol

MAGIC (1 byte)	Opcode (1 byte)	Key Length (2 bytes)
Extra Length (1 byte)	Data Type (1 byte)	Reserved (2 bytes)
Total Body Length (4 bytes)		
Opaque (4 bytes)		
CAS (Compare and Swap) (8 bytes)		

Pipelining!



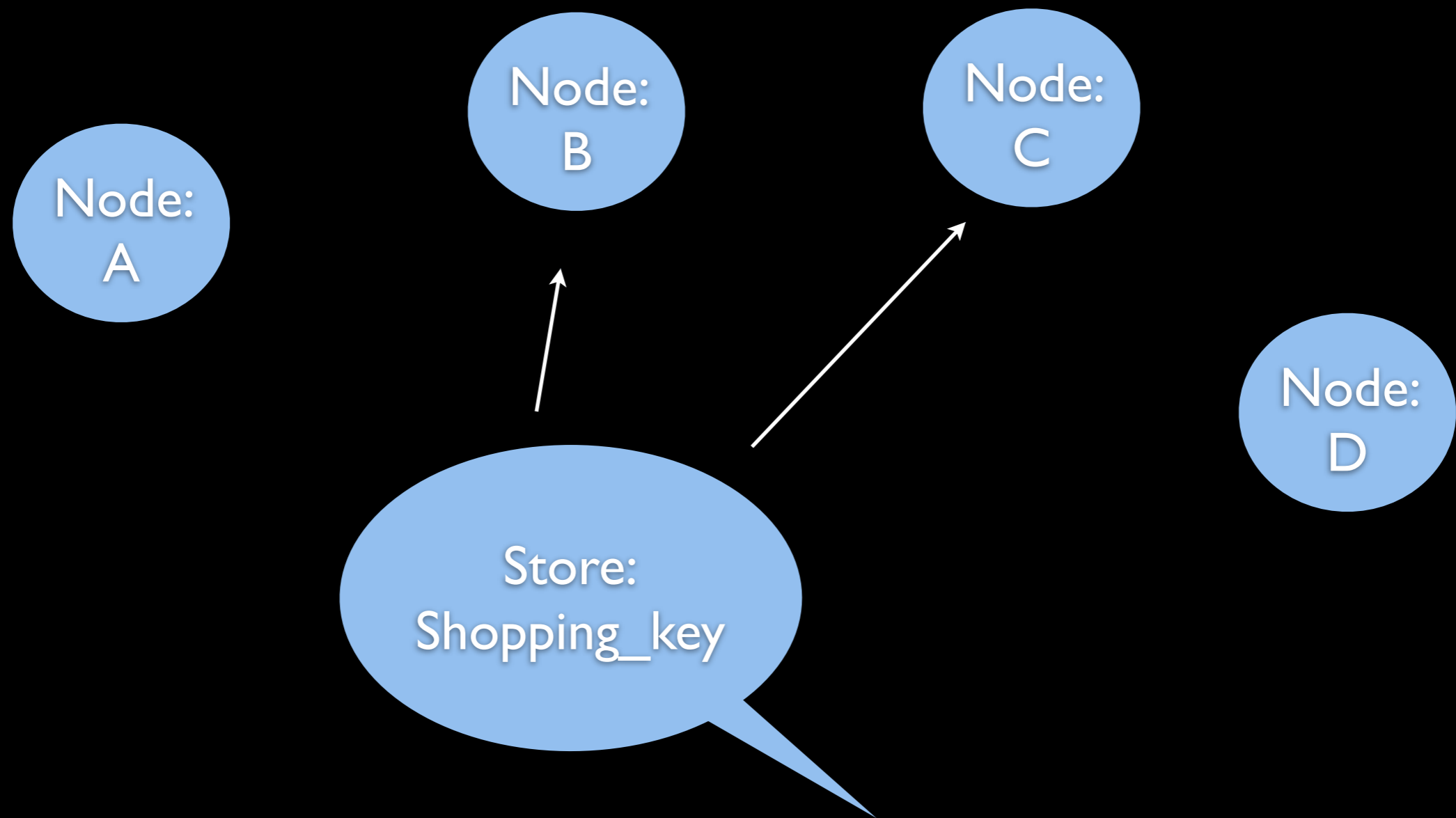


Consistent Hashing

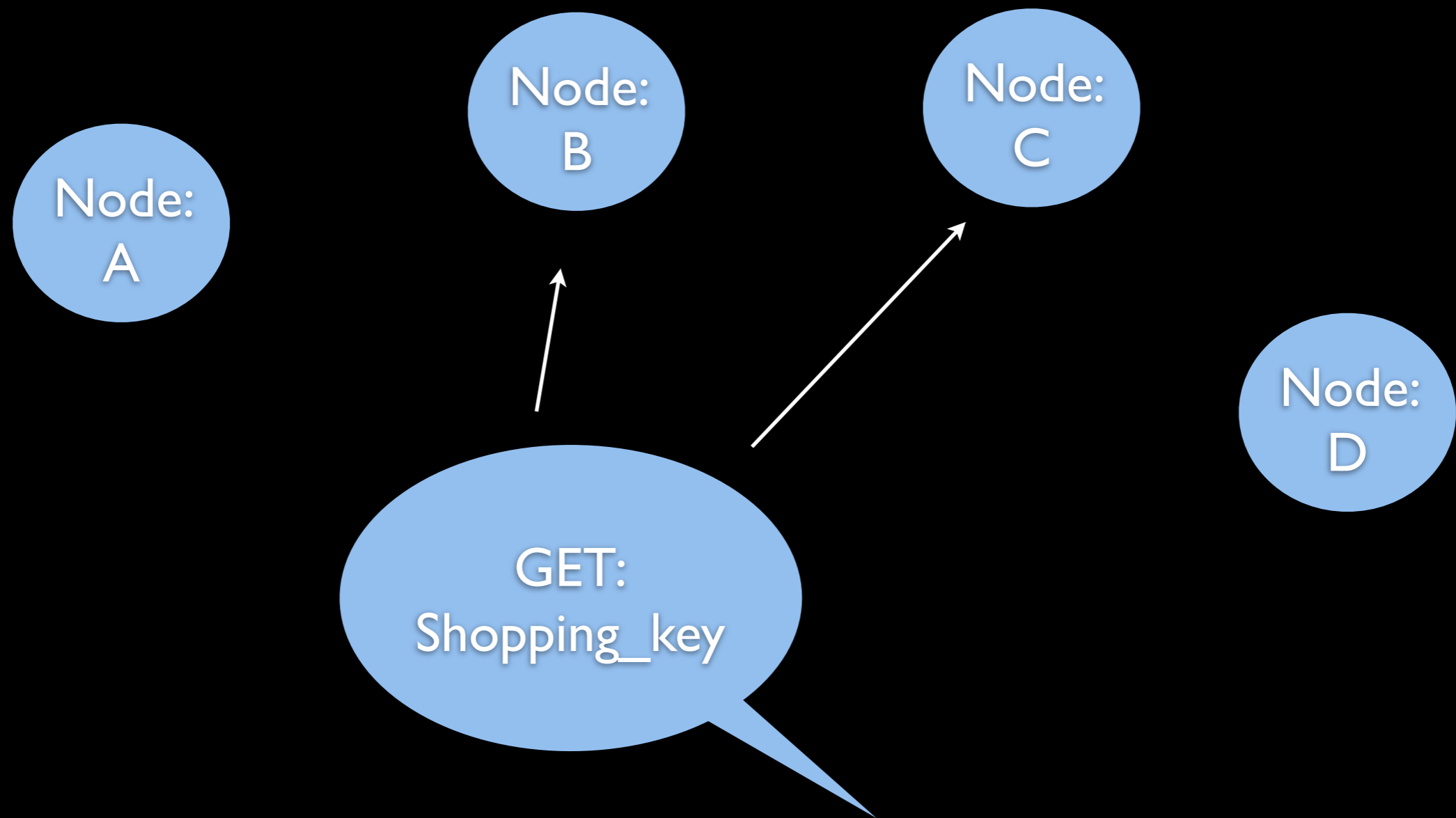
libhashkit

- One_at_a_time
- MD5
- HSIES (patents anyone?)
- CRC
- murmur (pretty simple)

Replica



Replica



STAT!

- `memstat --servers=localhost "args"`
 - `slab`
 - `items`
 - `sizes`

slab

```
STAT 1:chunk_size 104
STAT 1:chunks_per_page 10082
STAT 1:total_pages 1
STAT 1:total_chunks 10082
STAT 1:used_chunks 10081
STAT 1:free_chunks 1
STAT 1:free_chunks_end 10079
STAT 9:chunk_size 696
STAT 9:chunks_per_page 1506
STAT 9:total_pages 63
STAT 9:total_chunks 94878
STAT 9:used_chunks 94878
STAT 9:free_chunks 0
STAT 9:free_chunks_end 0
STAT active_slabs 2
STAT total_mallocated 67083616
END
```

items

```
STAT items:2:number 1
STAT items:2:age 452
STAT items:2:evicted 0
STAT items:2:evicted_nonzero 0
STAT items:2:evicted_time 2
STAT items:2:outofmemory 0
STAT items:2:tailrepairs 0
...
STAT items:27:number 1
STAT items:27:age 452
STAT items:27:evicted 0
STAT items:27:evicted_nonzero 0
STAT items:27:evicted_time 2
STAT items:27:outofmemory 0
STAT items:27:tailrepairs 0
```

sizes

96	35
128	38
160	807
192	804
224	410
256	222
288	83
320	39
352	53
384	33
416	64
448	51
480	30
512	54
544	39
576	10065

Dump your data?

- memdump
- memcached_dump(memcached_st *ptr, memcached_dump_fn *function, void *context, uint32_t number_of_callbacks);

Drain

```
for (items= memdump())  
{  
    store(items);  
    memcached_delete();  
}
```

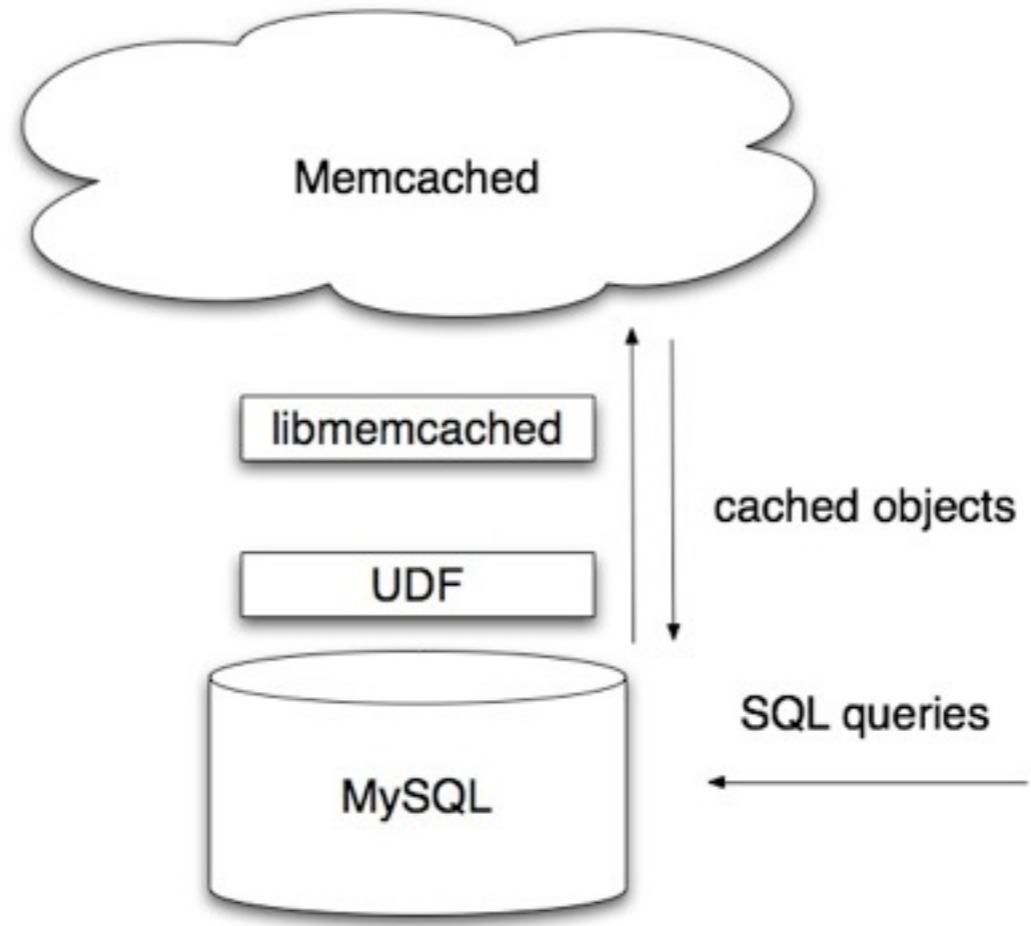
libmemcachedutil

- `libmemcached_util_ping();`
- `libmemcached_util_version_check();`
- `memcached_pool_st`

```
memcached_pool_st* pool= memcached_pool_create(memc, 5, 10);
memcached_st* mmc[10];

for (size_t x= 0; x < 10; ++x)
{
    mmc[x]= memcached_pool_pop(pool, false, &rc);
    ...;
}
memcached_pool_push(pool, mmc[x])
```

...and now for
databases



Setting via SELECT

```
select id, url, memc_set(concat('feeds', md5(url)), url) from feeds;
```

```
select memc_get(concat('feeds', md5(url))) from feeds;
```

```
+-----+
| memc_get(concat('feeds', md5(url))) |
+-----+
| http://feeds.feedburner.com/littlegreenfootballs/Ilds |
| http://del.icio.us/rss/jacomien |
| http://del.icio.us/rss/tags/rmj20 |
| http://www.jihadwatch.org/index.rdf |
| http://www.connotea.org/rss/user/rmj20 |
| http://devblog.grazr.com/?feed=rss2x |
| http://feeds.feedburner.com/littlegreenfootballs/kyeH |
+-----+
```

Trigger

```
DROP TRIGGER IF EXISTS feed_insert;
CREATE TRIGGER feed_insert BEFORE INSERT ON feeds FOR EACH ROW
BEGIN    SET @mm= memc_set(concat('feeds:',md5(NEW.url)),
NEW.url);END |
```

```
insert into feeds (url) values ('http://grazr.com/feedlist.xml');
```

```
select memc_get(concat('feeds:', md5('http://grazr.com/feedlist.xml')));
+-----+
memc_get(concat('feeds:', md5('http://grazr.com/feedlist.xml'))) |
+-----+
http://grazr.com/feedlist.xml |
+-----+
```

Memcached Replication via MySQL

```
INSERT INTO table_a VALUES ("key", "value", "values");  
INSERT INTO blackhole_table VALUES (memc_delete("key"));
```

```
drizzle> select table_name
-> from information_schema.tables
-> where table_name like '%MEMCACHED%';
+-----+
| table_name |
+-----+
| MEMCACHED_STATS |
| MEMCACHED_ANALYSIS |
+-----+
2 rows in set (0 sec)
```

```
drizzle> desc information_schema.memcached_stats;
```

Field	Type	Null	Key	Default	Extra
NAME	varchar(32)	NO			
PORT_NUMBER	bigint	NO		0	
PROCESS_ID	bigint	NO		0	
UPTIME	bigint	NO		0	
TIME	bigint	NO		0	
VERSION	varchar(8)	NO			
POINTER_SIZE	bigint	NO		0	
RUSAGE_USER	bigint	NO		0	
RUSAGE_SYSTEM	bigint	NO		0	
CURRENT_ITEMS	bigint	NO		0	

```
drizzle> desc information_schema.memcached_analysis;
```

Field	Type	Null	Key	Default	Extra
SERVERS_ANALYZED	bigint	NO		0	
AVERAGE_ITEM_SIZE	bigint	NO		0	
NODE_WITH_MOST_MEM_CONSUMPTION	varchar(32)	NO			
USED_BYTES	bigint	NO		0	
NODE_WITH_LEAST_FREE_SPACE	varchar(32)	NO			
FREE_BYTES	bigint	NO		0	
NODE_WITH_LONGEST_UPTIME	varchar(32)	NO			
LONGEST_UPTIME	bigint	NO		0	
POOL_WIDE_HIT_RATIO	bigint	NO		0	

libmemcachedprotocol

- Ascii
- Binary

```
memcached_binary_protocol_callback_st interface_vl_impl= {  
    .interface_version= MEMCACHED_PROTOCOL_HANDLER_V1,  
    .interface.vl= {  
        .add= add_handler,  
        .append= append_handler,  
        .decrement= decrement_handler,  
        .delete= delete_handler,  
        .flush= flush_handler,  
        .get= get_handler,  
        .increment= increment_handler,  
        .noop= noop_handler,  
        .prepend= prepend_handler,  
        .quit= quit_handler,  
        .replace= replace_handler,  
        .set= set_handler,  
        .stat= stat_handler,  
        .version= version_handler  
    }  
};
```

./examples/memcached_light

(everyone needs their own persistent Memcached server)

Examples!

- `tests/mem_functions.c`
- `man "libmemcached_examples"`

Brian Aker (brian.aker@datadifferential.com)

twitter:brianaker

url: Krow.net

