The strange details of std::string at Facebook

Nicholas Ormrod

Software Engineer

CppCon 2016

Questions I have answers for

- What different string implementations exist?
- How are strings optimized?
- What goes wrong when hunting for improvements?

Questions I want answers for

- What is the most efficient string implementation?

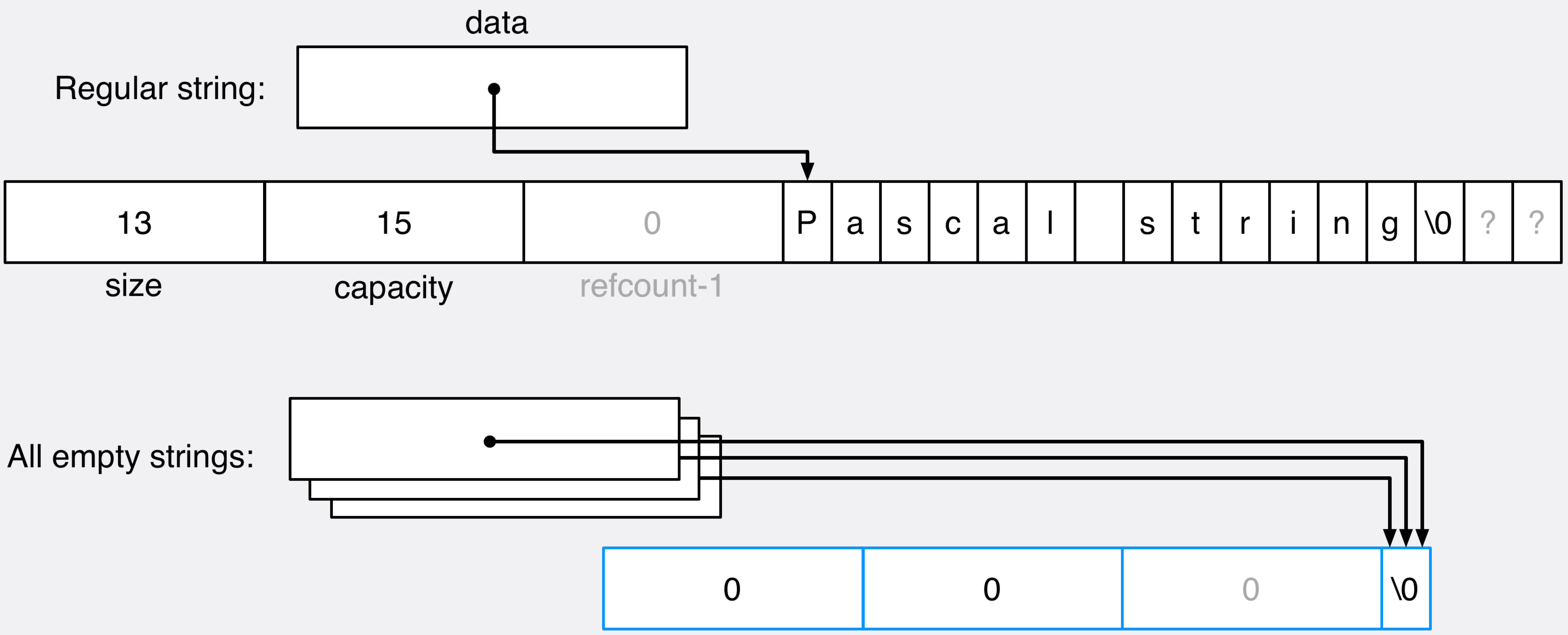
Why are strings important?

- `<string>` is the most-included file at Facebook
- Accounts for 18% of all CPU time spent in `std`
- There are simple ways to optimize strings

```c
struct string {
    int size;
    int capacity;
    char * data;
};
```

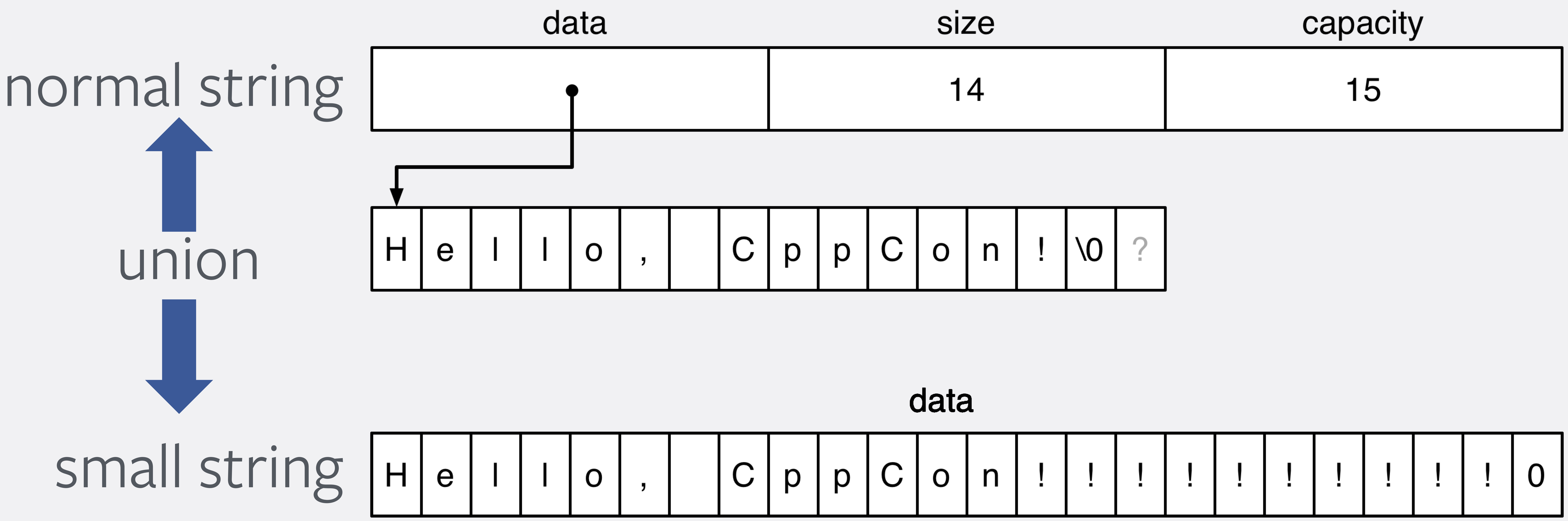
?

gcc string (version <5)



fbstring

@author Andrei Alexandrescu


data
size
capacity
normal string
14
15
union
H e l l o ,   C p p C o n ! \0 ?
data
small string
H e l l o ,   C p p C o n ! ! ! ! ! ! ! ! ! ! 0

Performance of fbstring

`gcc_string.size()`

```
movq     (%rdi), %rax
movq     -24(%rax), %rax
```

1.6ns

`fbstring.size()`

```
  movabsq $-4611686018427387904, %rax
  testq   %rax, 16(%rdi)
  je      .L7
  movq    8(%rdi), %rax
  ret
.L7:
  movsbq  23(%rdi), %rdx
  movl    $23, %eax
  subq    %rdx, %rax
  ret
```

is_small

0.9ns

http://gcc.godbolt.org/

std::string replacement

- We replaced `std::string`'s implementation with fbstring's
- `std::string` and `folly::fbstring` now have the same implementation, but are still different types

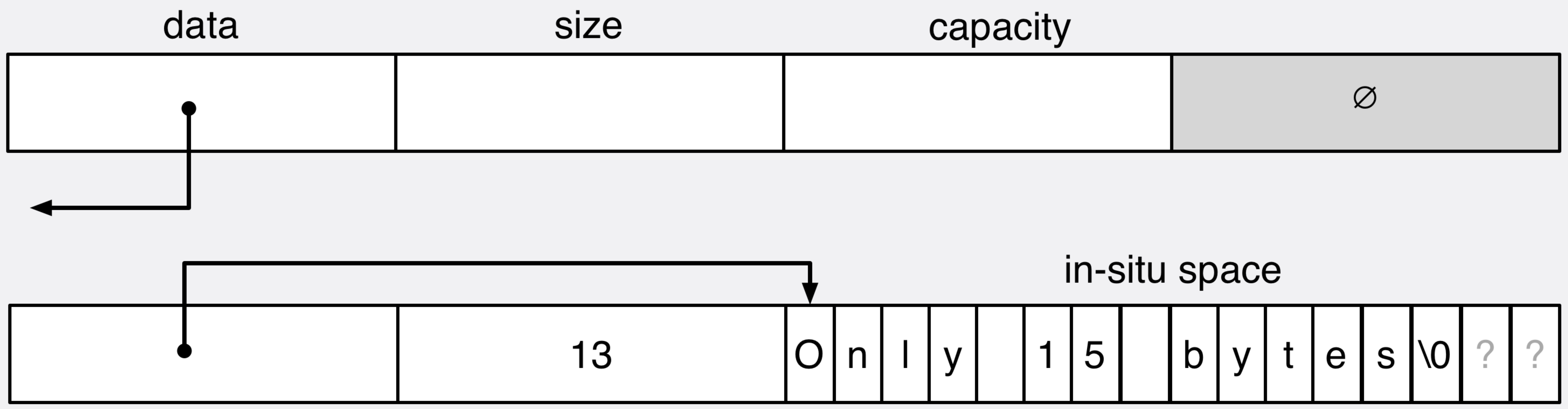
1% performance win

Killing the null terminator

- fbstring lazily wrote `\0`
- Added a mode that eagerly wrote `^` as terminator
- `c_str()`, `data()` will append `\0`

```
const Char * c_str() const {
    ...
    if (data[size()] != ‘\0’)
    data[size()] = ‘\0’;
    return data;
}
```

gcc string (version >=5)



+ Has SSO
+ `data()`, `size()` very fast
+ Size, 32, is power of 2

– Only 15-byte capacity
– Move is no longer memcopy
– Size is 33% larger than fbstring

Strangeness no more!

- There are lots of different ways to implement strings
- Small-String-Optimization is in, Copy-On-Write is out
- Memory layout is important

facebook