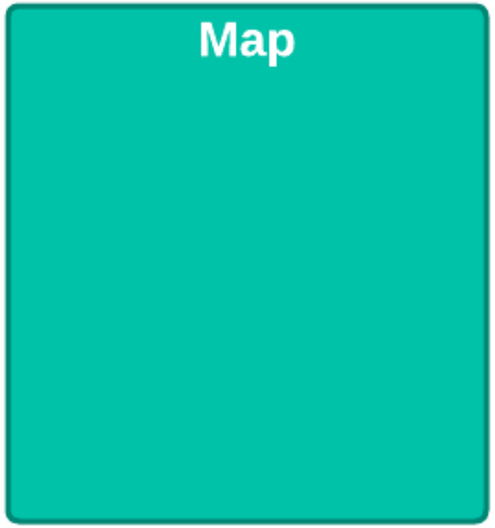
JAVA

МНОГОПОТОЧНОСТЬ.

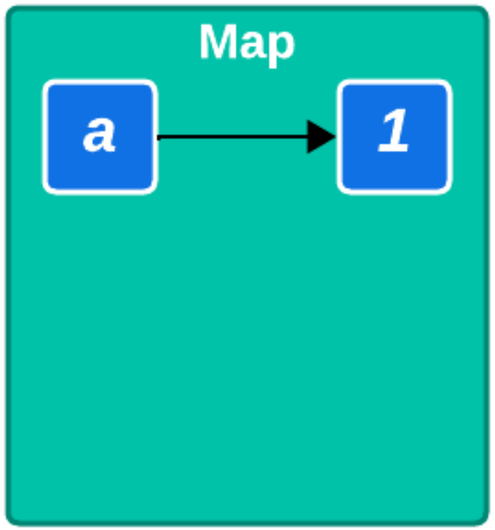
ConcurrentHashMap

A a k i u l s w S k i e o 1 2 D d

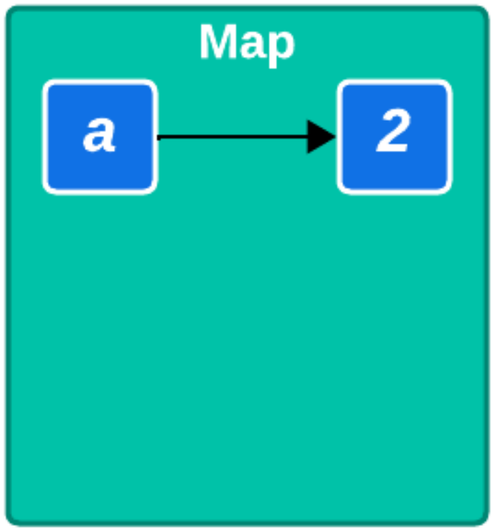
A a k i u l s w S k i e o 1 2 D d



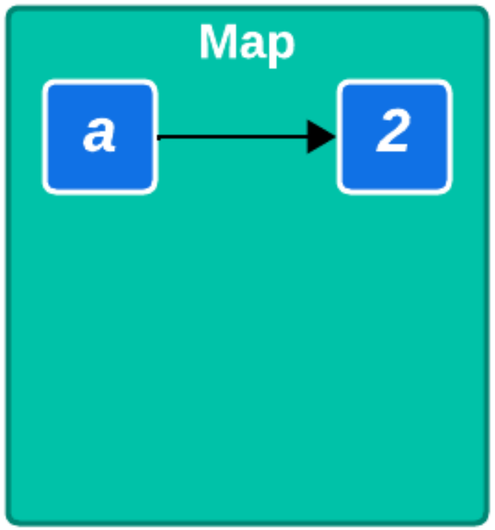

A a k i u l s w S k i e o 1 2 D d


Map
a
1

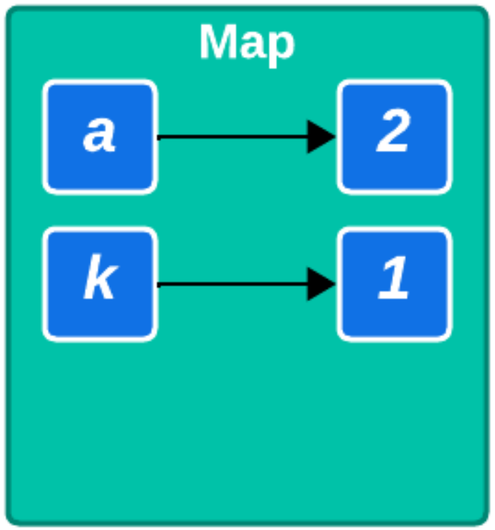
A	a		k	i	u			l		s	w		S	k	i	e	o	1	2	D	d

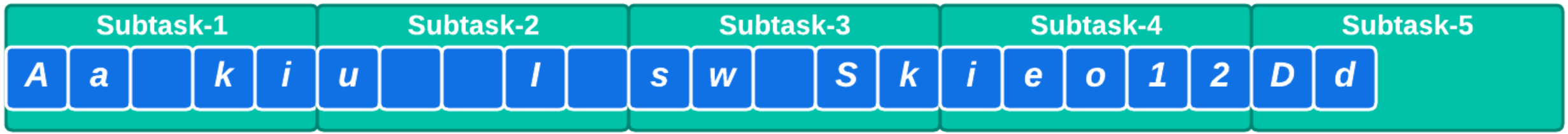


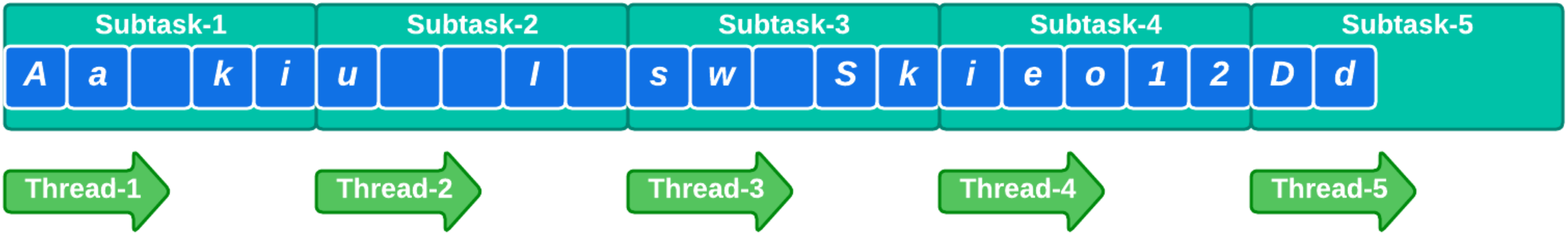

A a k i u l s w S k i e o 1 2 D d

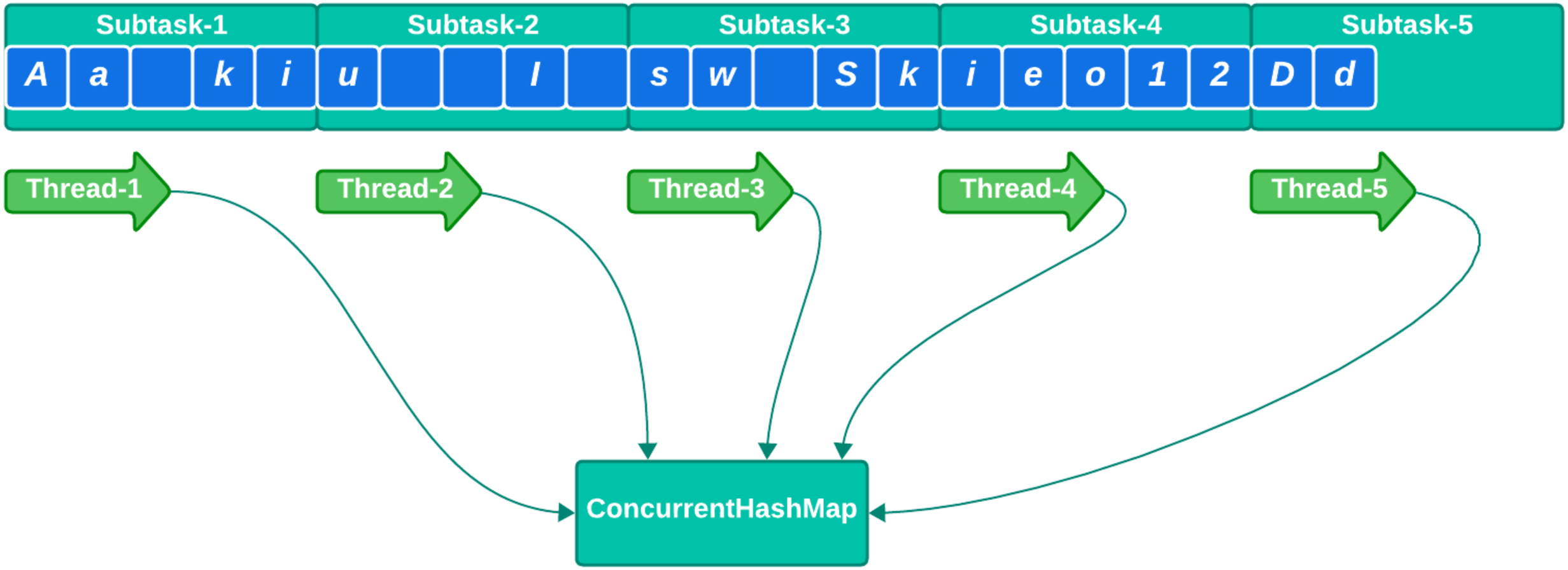

Map
a
2

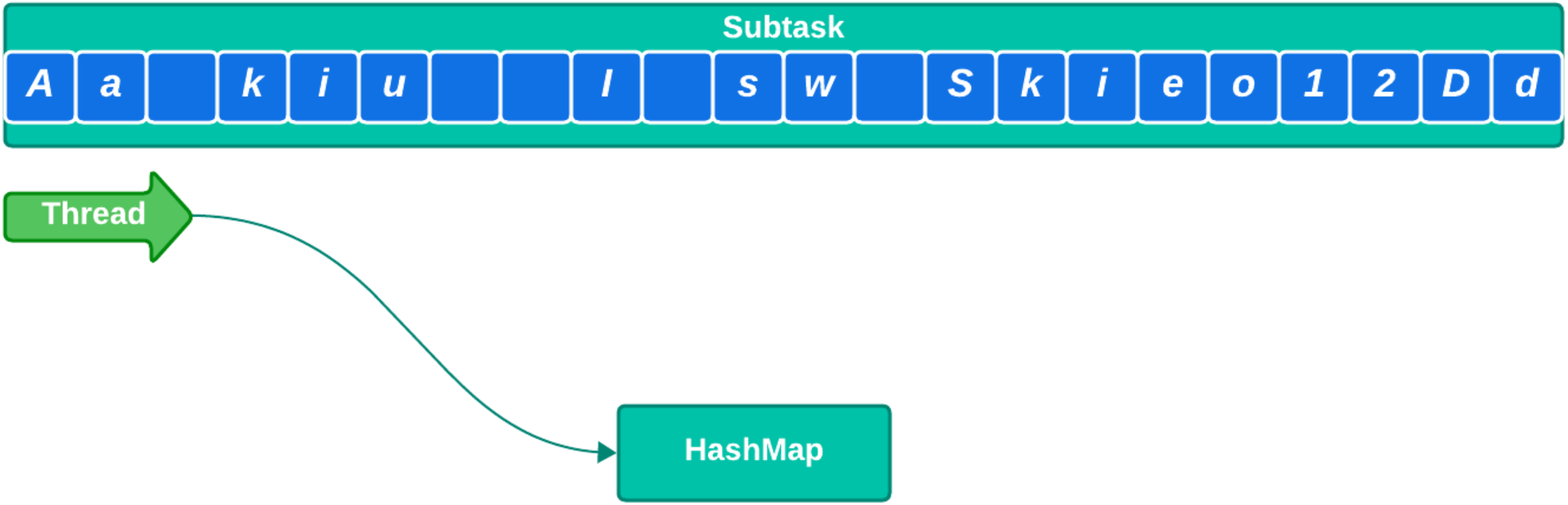

A a k i u l s w S k i e o 1 2 D d


Map
a
2
k
1


Subtask-1
Subtask-2
Subtask-3
Subtask-4
Subtask-5
A a k i u l s w S k i e o 1 2 D d


Subtask-1
Subtask-2
Subtask-3
Subtask-4
Subtask-5
A a k i u l s w S k i e o 1 2 D d
Thread-1
Thread-2
Thread-3
Thread-4
Thread-5

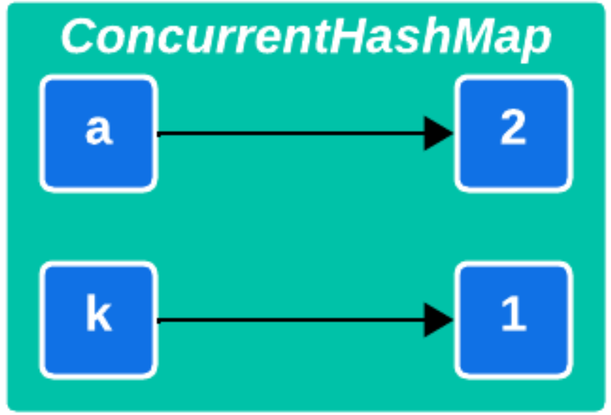

Subtask-1
Subtask-2
Subtask-3
Subtask-4
Subtask-5
A a k i u l s w S k i e o 1 2 D d
Thread-1
Thread-2
Thread-3
Thread-4
Thread-5
ConcurrentHashMap


Subtask
A a k i u l s w S k i e o 1 2 D d
Thread
HashMap

```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-1

codePoint	letter	frequency
-	-	-



codePoint	letter	frequency
-	-	-

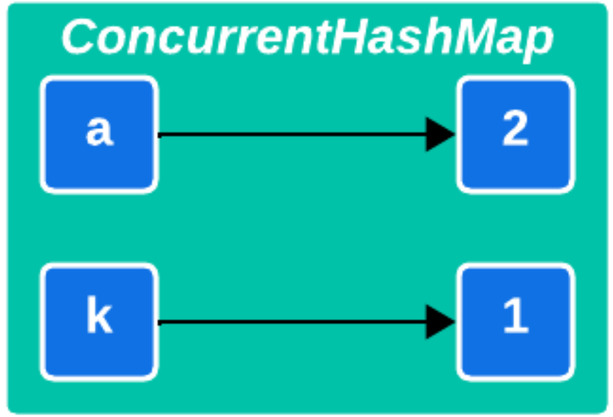
```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-2

```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-1

codePoint	letter	frequency
97	-	-



codePoint	letter	frequency
97	-	-

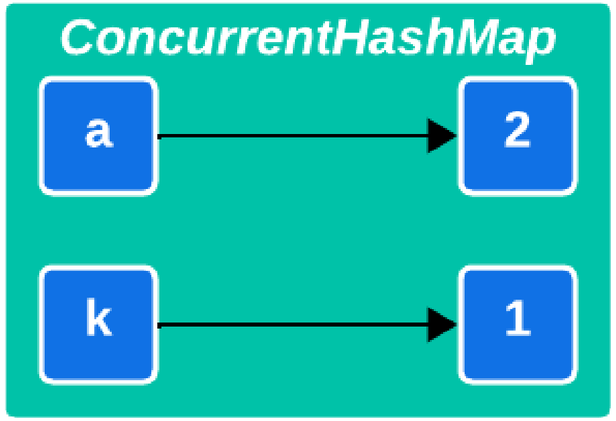
```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-2

```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-1

codePoint	letter	frequency
97	a	-



codePoint	letter	frequency
97	a	-

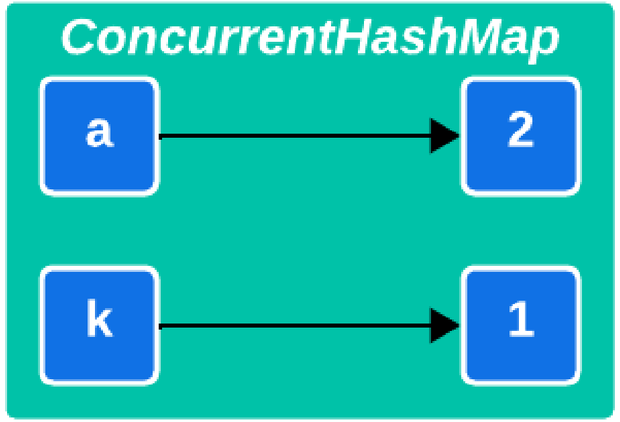
```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-2

```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-1

codePoint	letter	frequency
97	a	2



codePoint	letter	frequency
97	a	2

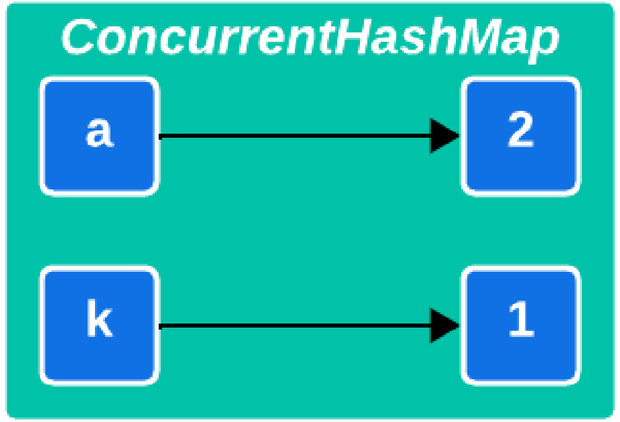
```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-2

```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-1

codePoint	letter	frequency
97	a	2



codePoint	letter	frequency
97	a	2

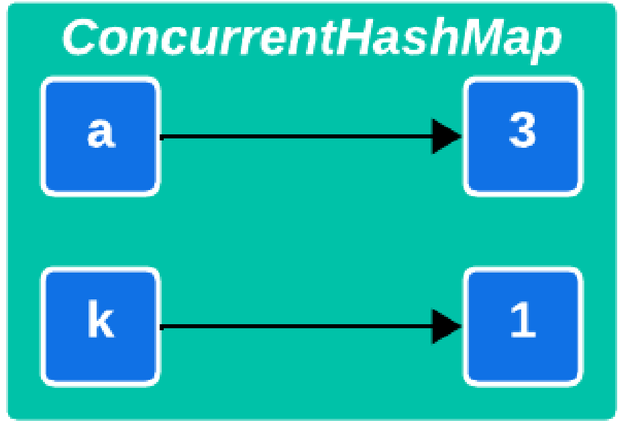
```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-2

```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-1

codePoint	letter	frequency
97	a	2



codePoint	letter	frequency
97	a	2

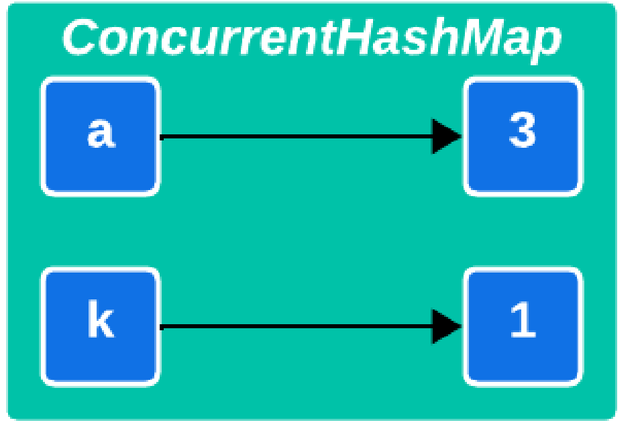
```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-2

```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

Thread-1

codePoint	letter	frequency
97	a	2



codePoint	letter	frequency
97	a	2

```java
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

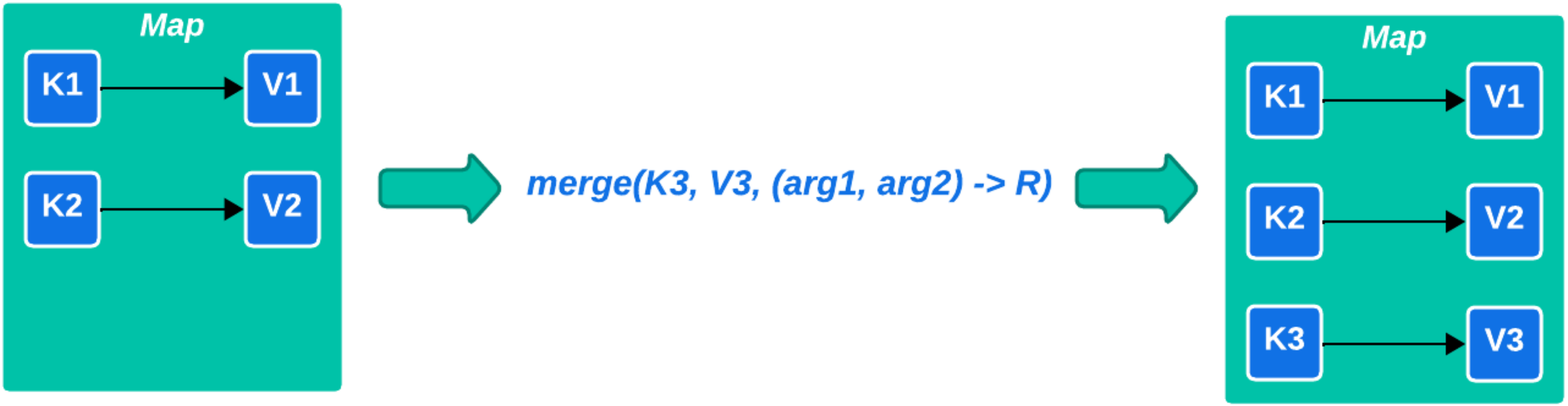
Thread-2

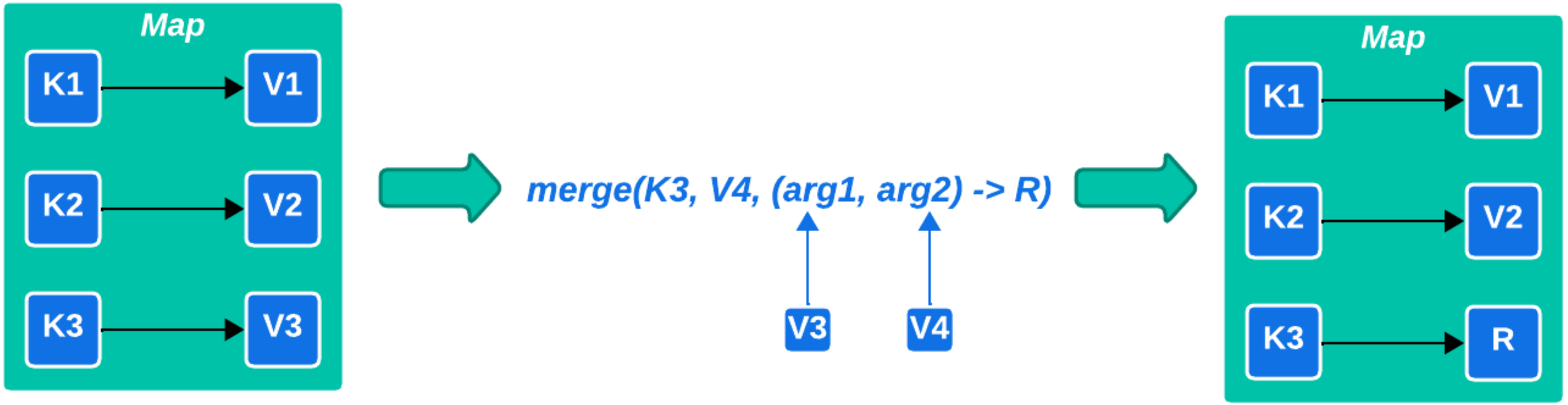
non-thread-safe

```
private void accumulate(final int codePoint) {
    final Character letter = (char) codePoint;
    final Integer frequency = accumulator.get(letter);
    if (frequency != null) {
        accumulator.put(letter, frequency + 1);
    } else {
        accumulator.put(letter, 1);
    }
}
```

thread-safe

```
private void accumulate(final int codePoint) {
    accumulator.merge((char) codePoint, value: 1, Integer::sum);
}
```


Map
K1
V1
K2
V2
merge(K3, V3, (arg1, arg2) -> R)
Map
K1
V1
K2
V2
K3
V3


Map
K1
V1
K2
V2
K3
V3
merge(K3, V4, (arg1, arg2) -> R)
V3
V4
Map
K1
V1
K2
V2
K3
R