
# Virtual function calls are not that slow, but...

Simon Heybrock

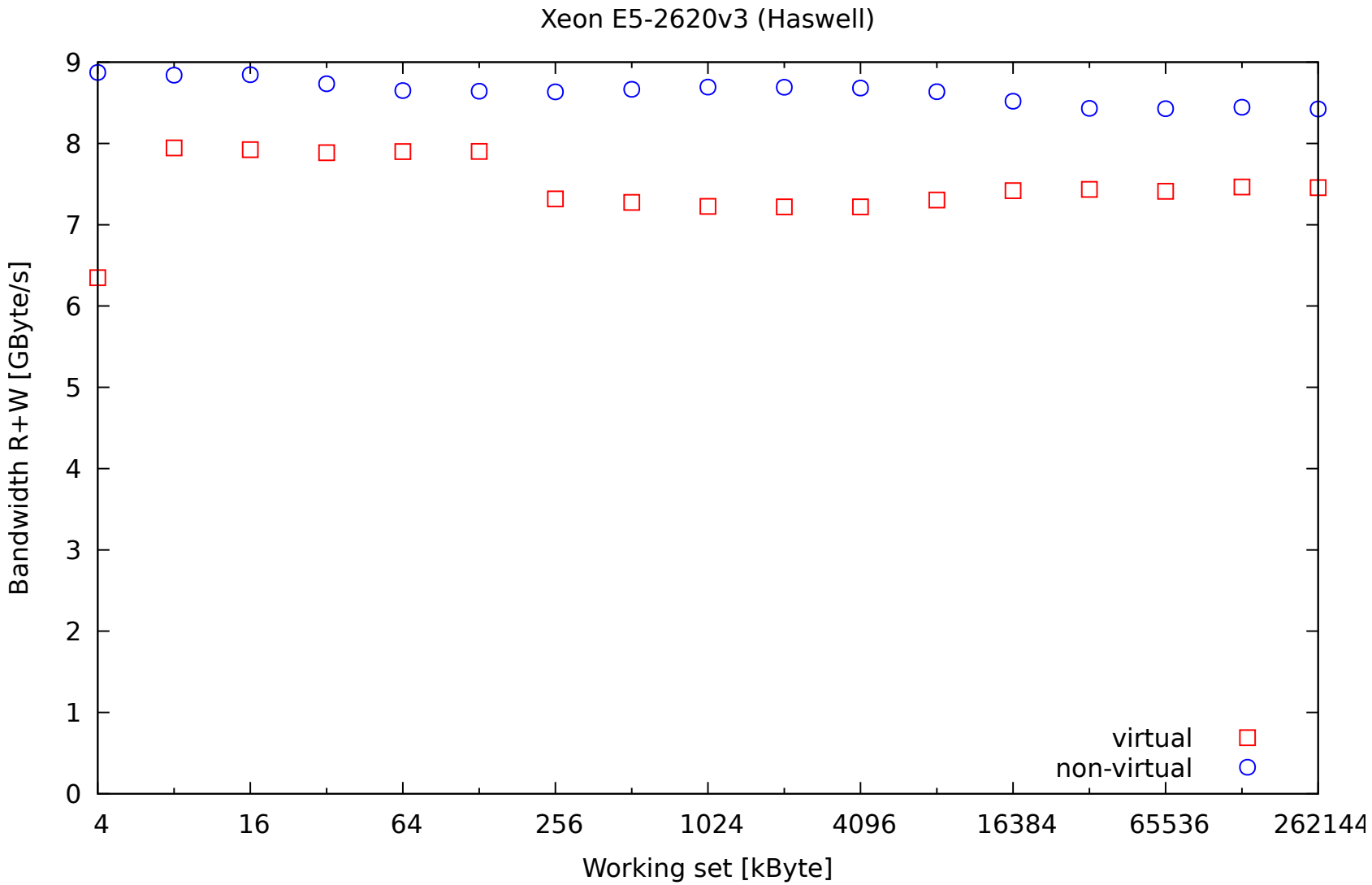
# Unit conversion

```
1  class Unit {
2  public:
3    void toTOF(std::vector<double> &xdata, /* ... */);
4    virtual double singleToTOF(const double x) const = 0;
5  };
```

```
1  void Unit::toTOF(std::vector<double> &xdata, /* ... */) {
2    initialize(/* instrument params */);
3    size_t numX = xdata.size();
4    for (size_t i = 0; i < numX; ++i)
5      xdata[i] = singleToTOF(xdata[i]);
6  }
```

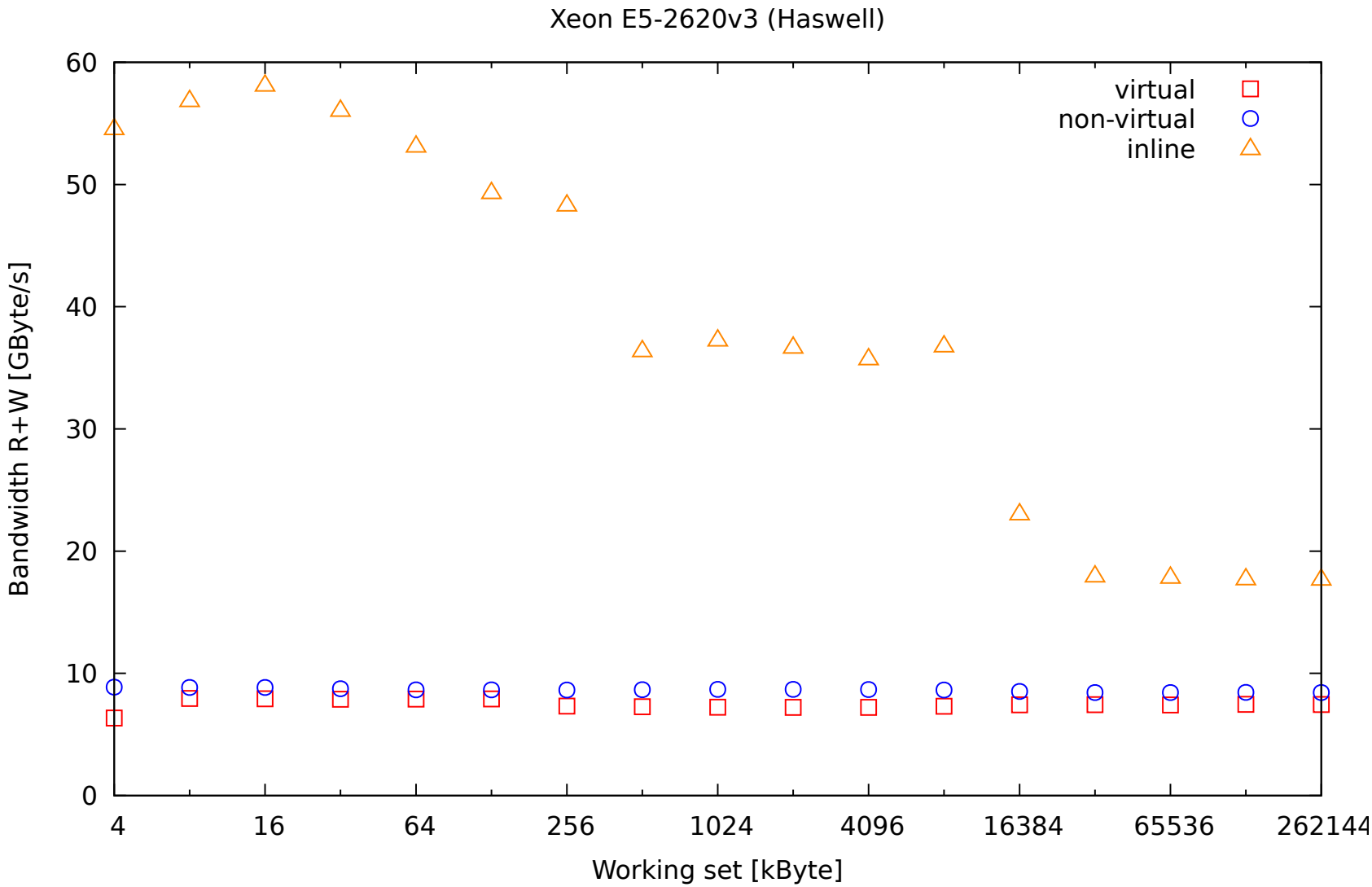
# Micro benchmarks: Virtual function call in inner loop

- Apply cheap conversion function to each element in a vector.



# Micro benchmarks: Virtual function call in inner loop

- Apply cheap conversion function to each element in a vector.



# Conclusion

## Virtual function calls. . .

- . . . are not slow.
- Not inlining is slow.

## In practice

- Can speed up `ConvertUnits` by doing virtual call for `toTOF`/`fromTOF` instead of `singleToTOF`/`singleToTOF`.
- Observed nearly 2x speed up when converting (sufficiently large) `Workspace2D`.
- Observed nearly 2x speed up when converting (sufficiently large) `EventWorkspace`.

# Micro benchmarks: Old desktop processor

- Apply cheap conversion function to each element in a vector.