WInnComm 2018

RPCexpress: a try to implement an efficient middleware from the ground up based on requirements of embedded software defined systems

Qi Tang[1,3], Jin Lian[2,1,3], Li Zhou[1,3], Shan Wang[1,3], Haitao Zhao[1,3], Jun Xiong[1,3], Shengchun Huang[1,3], Jibo Wei[1,3]

1 National University of Defense Technology, Changsha, Hunan, P.R. China

2 Hunan university, Changsha 410082, China

3 Hunan Engineering Research Center of Software Radio, Changsha, China

Content

- Introduction
- Features of RPCexpress
- Experiment method
- Experimental results
- Future work

Content

u Introduction

u Features of RPCexpress

u Experiment setting

u Experimental results

u Future work

Software defined systems

- The future of the world is software-defined, e.g.,
  - Software defined radio (GNUradio, SORA, SCA, STRS, ALOE)
  - Cognitive radio
  - Software radar

  - Distributed computing
  - Embedded programmable System-on-Chip
  - Multiprocessor system
  - Cellphone

Model driven, component based

- System complexity is increasing
- Component-based software developing is appealing
  - Development simplicity
  - Test
  - Integration
  - Maintaining
- The SDS is composed of a set of software components
  - How to interconnect different components
  - How to enable system reconfiguration
    - OS support
    - Architecture support
    - Middleware support

Middleware

- "Middleware can be viewed as a reusable, expandable set of services and functions that are commonly needed by many applications to function well in a networked environment".
- Classification
  - message oriented middleware (MOM): DDS, ZeroMQ
  - remote process all (RPC): ICE, RMI, gRPC, DDS-RPC and COM+
  - object request broker (ORB): ACE TAO, omniORB, e*ORB, ORBexpress,ORBit
  - Database
  - ……

Content

u Introduction

u Features of RPCexpress

u Experiment setting

u Experimental results

u Future work

RPCexpress

- Features
  - Objecte-oritented
  - Language support: C++
  - OS support: Windows, linux
  - Basic datatypes
    - (unsigned) char, short, int, long, float, double, string, etc,.
  - Complex datatypes
    - sequence, array, struct, interface, exception, Any , etc,.
  - C++ based IDL compiler

RPCexpress

- Features
    - Support component-based development
    - Using shared memory, the fastest IPC mechanism
    - Support interface inheritance
    - Support clt-svc service
        - Clt, svc can locate in the same or different threads/processes

Content

u Introduction

u Features of RPCexpress

u Experiment setting

u Experimental results

u Future work

Experiment environment

- Host:
  - Intel Core i7-6700 processor
    - 3.4 GHz and 8 GB RAM
  - 64 bits Windows 7
- Virtual machine
  - Vmware
  - OS
    - Ubuntu 14

Compared middlewares

- ACE TAO
- omniORB
- e*ORB
- ORBit
- PRCexpress

Experiment method

- For each middleware
    - Create the same IDL interface
    - Generate the middleware-bound SVC/CLT related files
    - Implement the CLT and SVC
        - provide the same function, i.e., sending and receiving data sequence, according to the IDL interface
        - The SVC record the transfer delay

Comparison metrics

- Different data types
  - Delay
  - throughput
  - Predictability
- Static footprint occupation
  - Shared/executable file size
  - IDL compiler-generated stubs/skeletons file size
- Dynamic memory occupation
  - Physical memory size of the process

Content

u Introduction

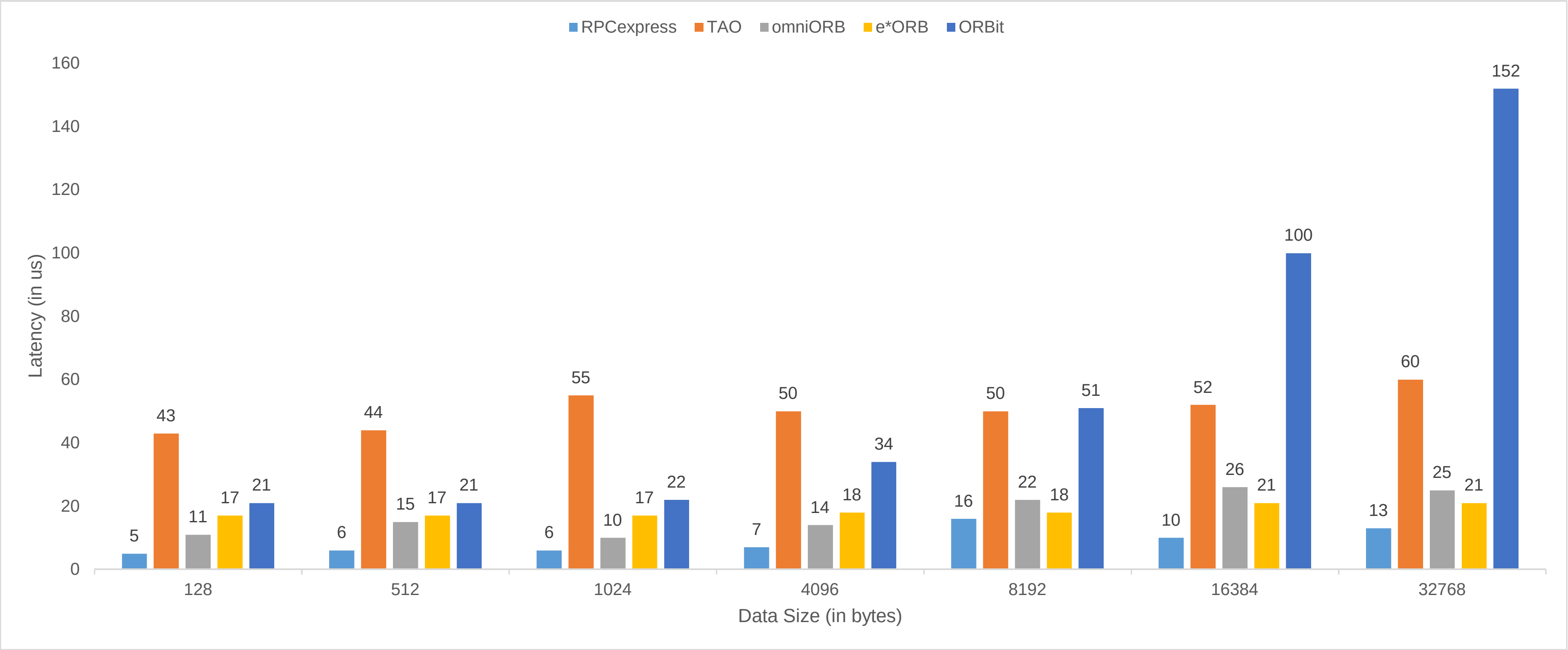
u Features of RPCexpress

u Experiment setting

u Experimental results

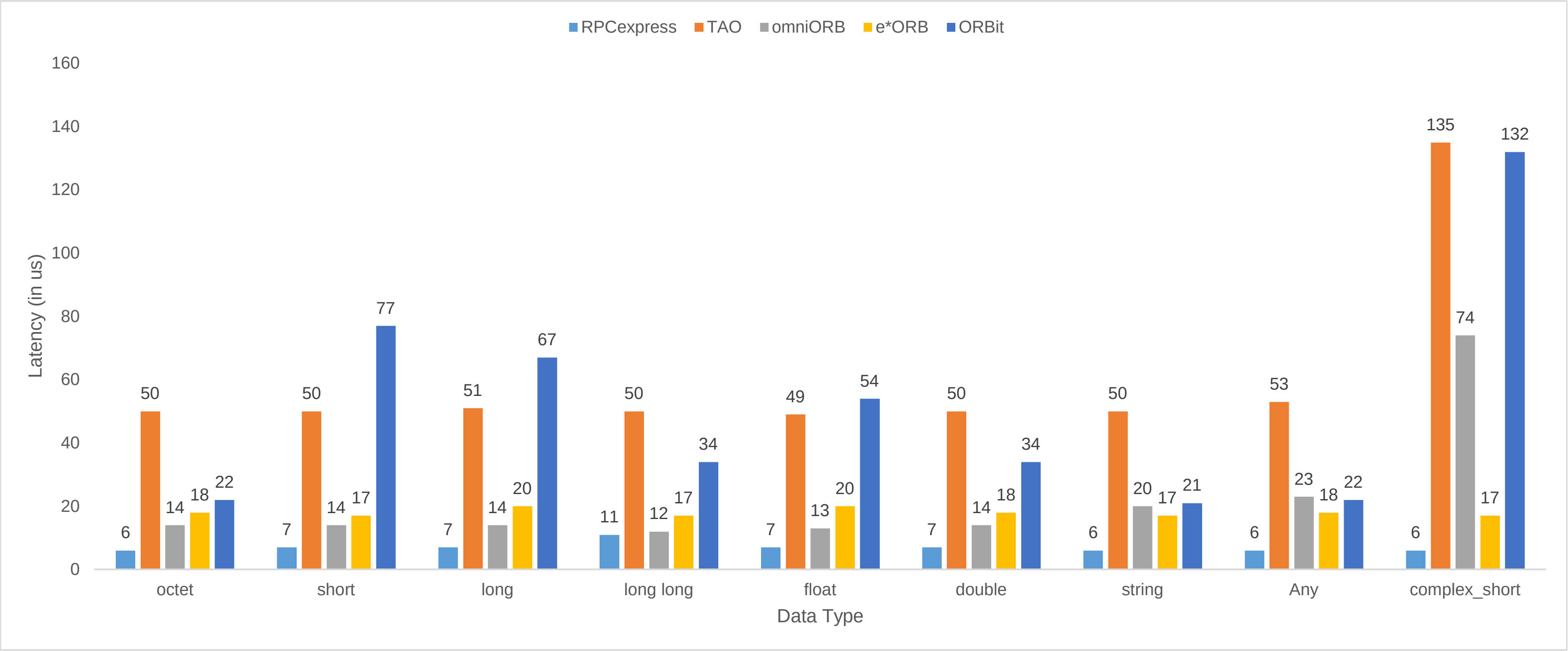
u Future work

Latency



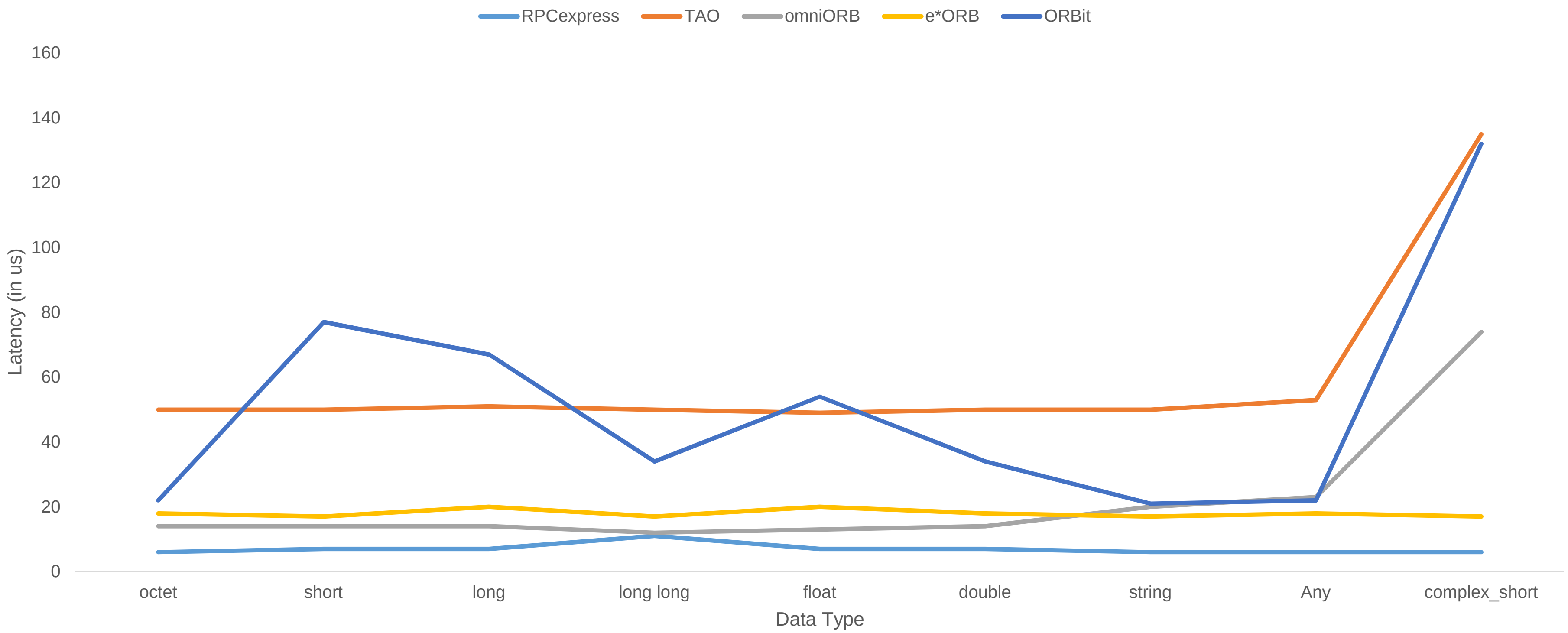
the calling latency of each middleware for the
double type data sequence with different data lengths

Latency



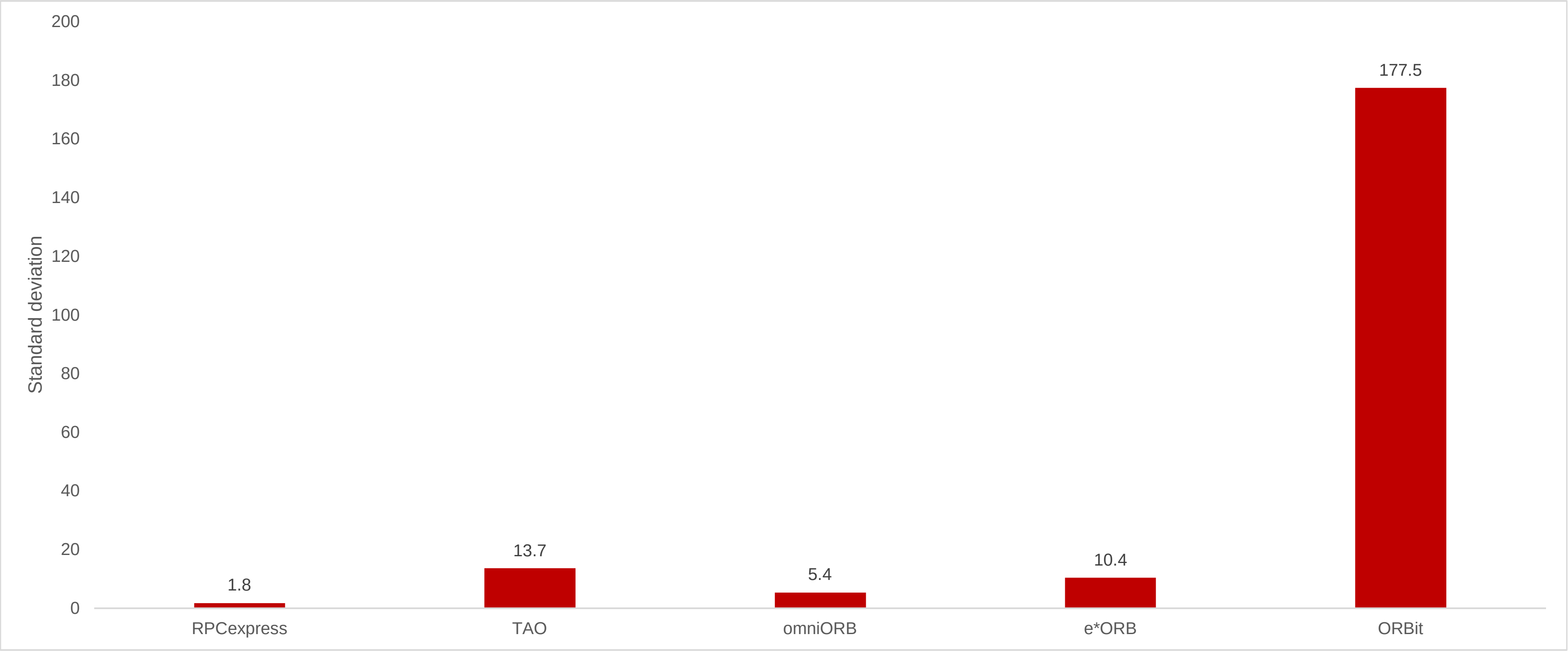
the calling latency of each middleware for the
4k bytes data sequence with different data types

Latency

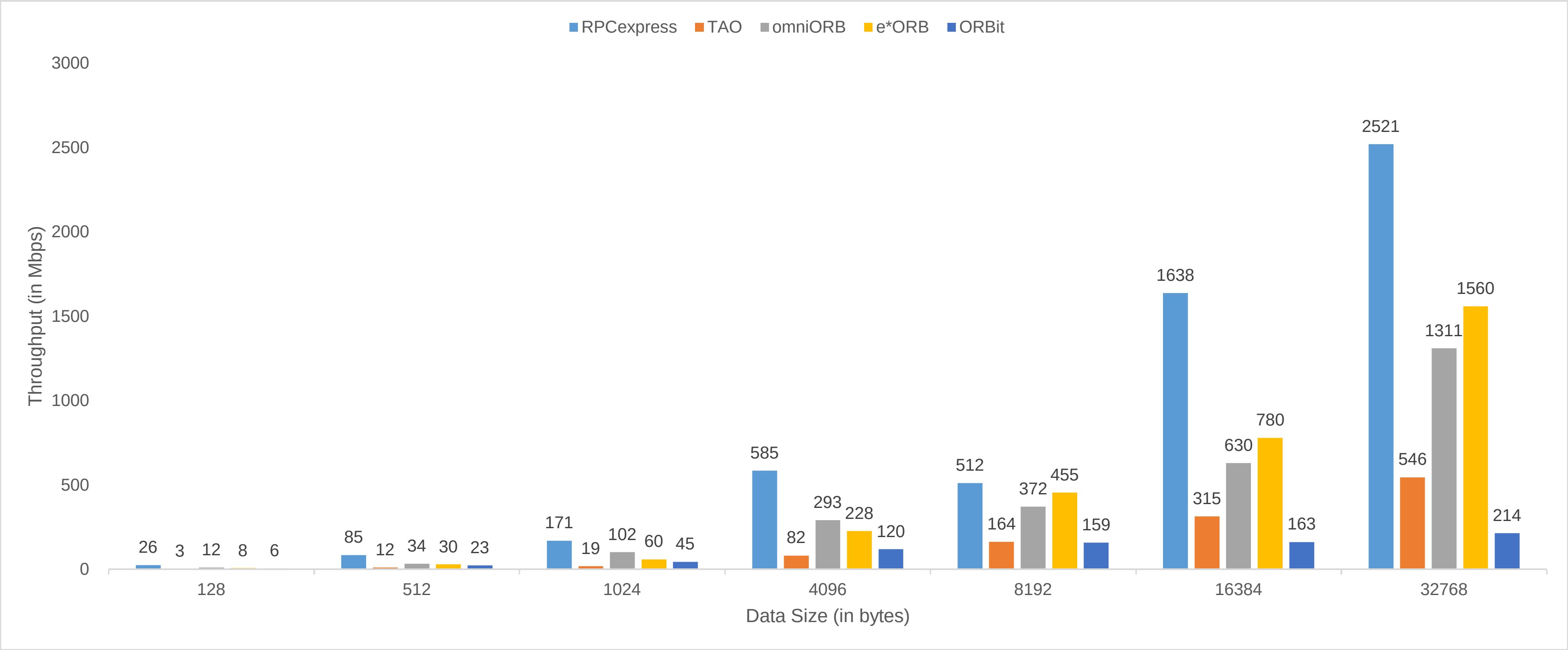


the calling latency of each middleware for the
4k bytes data sequence with different data types

Predictability

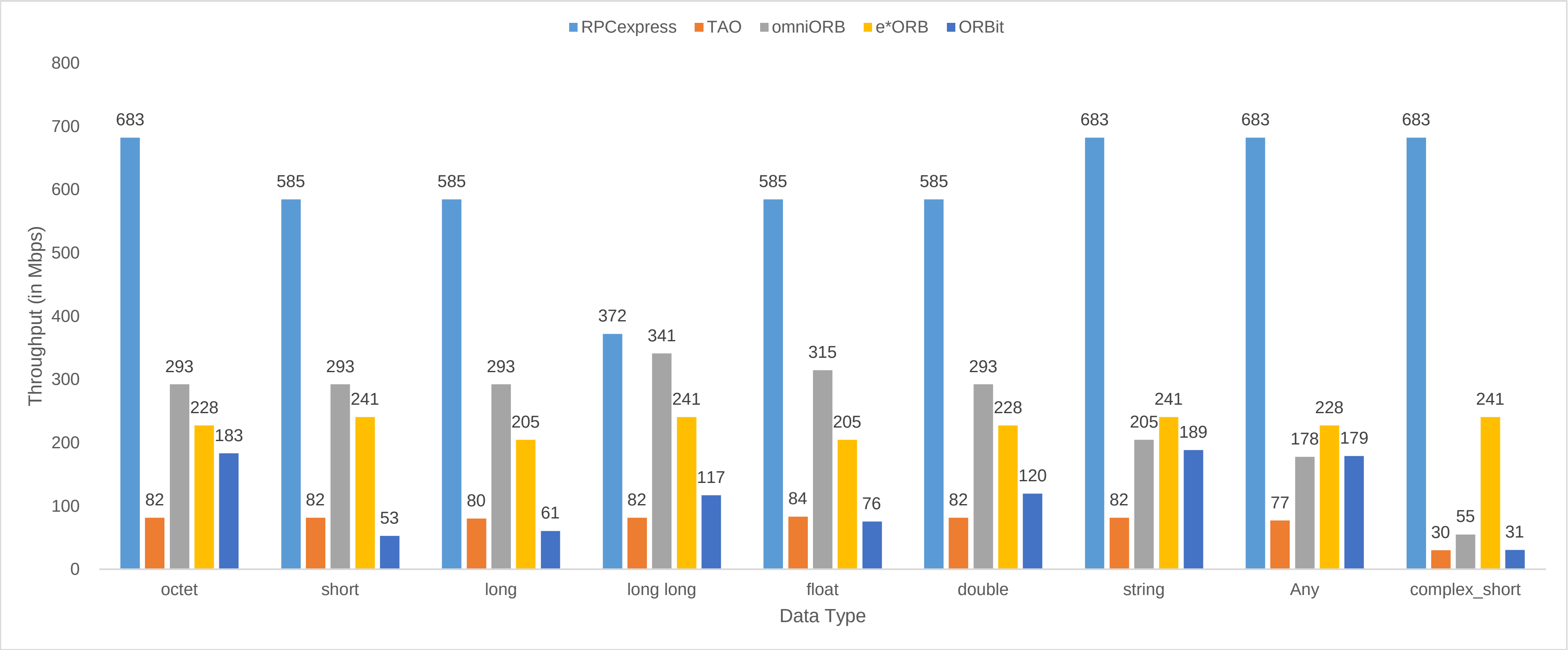


Throughput



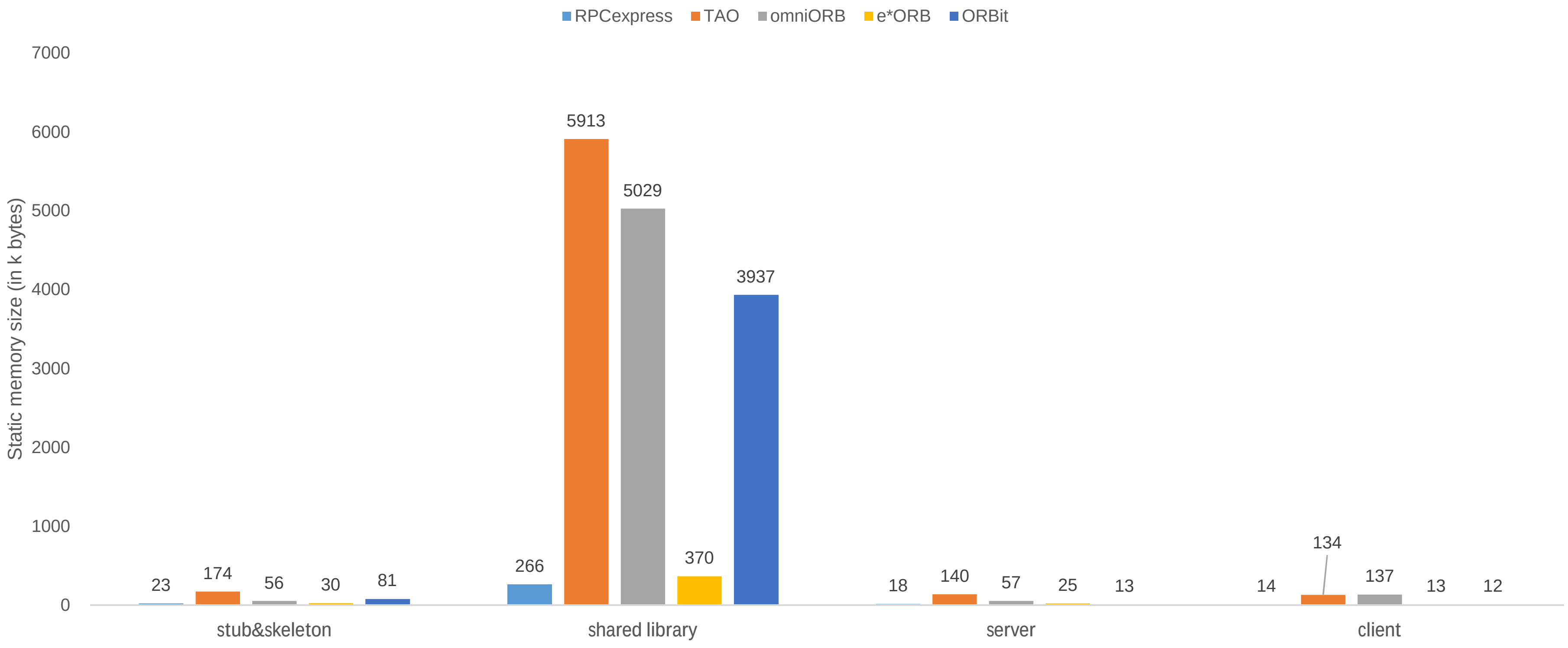
the throughput of each middleware for the
double type data sequence with different data lengths

Throughput

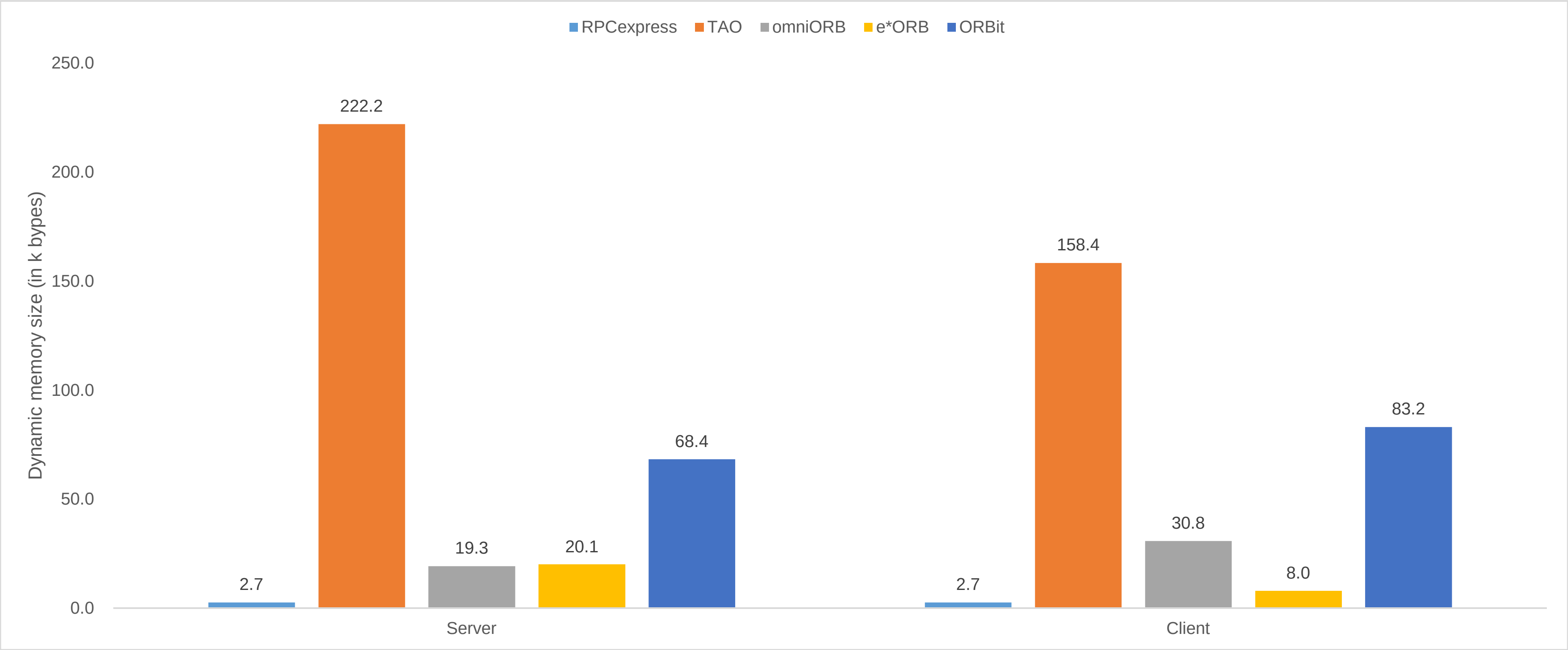


the throughput of each middleware for the
4k bytes data sequence with different data types

Static Memory Size



Dynamic Memory Size



Content

- u Introduction
- u Features of RPCexpress
- u Experiment setting
- u Experimental results
- u Future work

Future work

- Integrate with SCA-based SDR system
- Test with SDR waveform, e.g., FM3TR
- Extend to distributed systems
- Further performance optimization
- Extend to specific processors, e.g., DSP, FPGA

Thanks