Diperan (DIstributed PERformance ANalytics)

By Sotiris Oikonomou
mtp181

Motivation

- Close to hardware developing
- Learn and implement more on distributed systems
- Creating from scratch

Technologies used

- C++14
- Flatbuffers
- Zmq
- Zyre
- PAPI
- Systemd (run as a service)

Technologies used

Why Flatbuffers?

- Access to serialized data without **parsing/unpacking** - (forwards/backwards compatibility)
- Memory efficiency and speed - **0** additional allocations (in C++)
- Tiny code footprint
- Strongly typed
- Cross platform - Cross language

Technologies used

Why ZeroMQ?

- Cross platform - Many language implementations
- Message granularity oriented
- Socket like API
- Many to many connections between endpoints
- Transport over: TCP, PGM, IPC

Technologies used

Why Zyre?

- Cross platform - Many language implementations
- Based on ZeroMQ for message transport
- Zyre needs no administration or configuration
- Peers may join and leave the network at any time
- Peers talk to each other directly or in groups without any central brokers or servers
- Fast, reliable, low latency, delivery guarantee even when the network is heavily loaded

Technologies used

Why Zyre (2)?

- Designed for WiFi networks, can work over Ethernet too
- Uses UDP for discovery and heartbeating
- Fast join time (1s)
- Use cases:
    - Local service discovery.
    - Clustering of a set of services on the same network
    - Internet of Things - Multi-user mobile applications

Use cases

- Gathering of Hardware and Software information of the nodes
- CPU and system status of the distributed nodes
- Benchmarking of the nodes with configurable benchmarks (Currently SHA2, CRC32 and Matrix Multiplication)

Evaluation

- Test run
  - 6 nodes, 3 ARM (1 ARMv7-A, 2 ARMv8-A) cpus - 3 x86 cpus
    - 2 Raspberry Pi 3
    - 1 Nitrogen6x
    - 1 Atom N280
    - 1 Intel Core 2 Duo P8400
  - WIFI networking

Deployment

- Simple unix make build environment
- Designed to be deployed and run as **Systemd** service after the initialization of the network interface of the node

Future development

- Cmake build environment
- Control nodes individually, partially or the whole network
- Gather more system status information
- Semi-autonomous bridge-relay between separate networks
- Testing framework

THE END

Future development

- Bluetooth connectivity
- ANT+ connectivity
- Bridge-relay bluetooth or ANT+ and IP based networks
- WIFI configuration over Bluetooth or ANT+