Embedded Distributed Systems: A Case of Study

Victor Rodriguez
Intel OTC Guadalajara Mexico

Thesis Professor
Marcos de Alba
Agenda

● The rise of IoT .. the rise of data .. the rise of problems
● Do we really need a server ?
● MPI meet the new embedded systems
● Conclusion
The rise of embedded & IoT ....

Independent failures
No global clock
Concurrency

Distributed Systems Concepts and design George Coulouris
The rise of data

Data just from embedded systems already accounts for 2% of the digital universe. By 2020 that will rise to 10%.

EMC Digital Universe with Research & Analysis by IDC
The rise of real problems

“The total amount of user data (data payload) to be stored or processed doubles every two years”: Chemnitz, Germany

“Boeing 787s to create half a terabyte of data per flight”: Virgin Atlantic
Distributed System

Advantages
- Partitioned Workload
- Heterogeneous HW

Disadvantages
- Network
- Lack of optimized OS
Do we really need a server ($$$) ?

Prove that some distributed embedded systems can process their own big data
Previous Work: LMPI

An example of a traditional MPI.

An example of the LMPI system.
Previous Work: LMPI

The CPU time with different number of nodes in Jacobi with 4 processes
Results: MPI meet the new embedded systems

- All Reduce
- Bidirectional Bandwidth
- All to All
- Latency and round trip
MPI_Reduce

0 5 1 1 2 3 2 7 8 3 4 2

0 18 14

MPI_Allreduce

0 5 1 1 2 3 2 7 8 3 4 2

0 18 14 1 18 14 2 18 14 3 18 14
if (am_i_the_master()) {
    TIMER_START;
    for (i=0; i<cnt; i++) {
        mp_irecv(dest_rank, 2, destbuf, bytes, &requestarray[1]);
        mp_isend(dest_rank, 1, sendbuf, bytes, &requestarray[0]);
        MPI_Waitall(2, requestarray, statusarray);
    }
}

else if (am_i_the_slave()) {
    for (i=0; i<cnt; i++) {
        mp_irecv(source_rank, 1, destbuf, bytes, &requestarray[0]);
        mp_isend(source_rank, 2, sendbuf, bytes, &requestarray[1]);
        MPI_Waitall(2, requestarray, statusarray);
    }
}
MPI_Comm_size(comm, &n);
for (i = 0, i < n; i++)
    MPI_Send(sendbuf + i * sendcount * extent(sendtype),
             sendcount, sendtype, i, 0, comm);
for (i = 0, i < n; i++)
    MPI_Recv(recvbuf + i * recvcount * extent(recvtype),
             recvcount, recvtype, i, 0, comm);
Conclusion: Everybody wants the control

- Not all the systems apply for this
- Energy must be characterized
- We need custom OSs and Middleware
  - Yocto patches for MPI on RFC in meta-oe
  - OS for specific architecture
  - Implement LMPI

\[ E = \frac{P}{W} \]

number of systems
Resources

- Distributed Systems Concepts and design George Coulouris
- EMC Digital Universe with Research & Analysis by IDC
- Energy efficient handling of big data in embedded, wireless sensor networks Bergelt, R. Tech. Univ. Chemnitz, Chemnitz, Germany
- LMPI: MPI for heterogeneous embedded distributed systems Agbaria, A.; Inf. Sci. Inst., Southern California Univ., Los Angeles, CA; Dong-In Kang; Singh, K.
- Clear Linux Project for Intel® Architecture