Hardware Accelerated Application Integration: Challenges and Opportunities

Active @ ACM/IFIP/USENIX Middleware 2017

Daniel Ritter
Application Integration

- De-coupling apps
- Solving n-square connection and variety problems (for textual data) [Lin2000]
- Message routing and transformation patterns from 2004 [HW2004]
Emerging Application Integration

since 2000 / 2004 ...

Mobile Apps

EAI System, Integration Processes

Lambda, Zeta, Micro-service Architectures...

Connected Things

Cloud Apps, Business Networks

On-premise Apps
Emerging Trends lead to Challenges

EAI Challenges:
- New variety problem, e.g., media message protocols
- Number of messages (Velocity)
- Message sizes (Volume)
- Fault-tolerance (Stability)
- ...

General Challenges:
- Data center efficiency
- Power consumption
- ...

On-premise Apps
Cloud Apps, Business Networks
Mobile Apps
Connected Things
Lambda, Zeta, Micro-service Architectures...
Classical Solution Space

Challenges:
- Media message protocols (variety problem)
- Fault-tolerance (Stability)

Solutions:
- Message sizes (Volume)
- Number of messages (Velocity)
- Scale (out, up), e.g., parallelization, batching
- Software solutions, e.g., streaming, process / algebraic simplification, data reduction (up to sampling)

Side-effects:
- Hardware, Data center scaling, build power plants
- Software latency and throughput limited

Conversion, User Interaction, ...

per operation

Software latency and throughput limited [LGMBEV2012]
Example: Message Routing

Content-based Router

1..n

Conditional

1..n-1 predicates / conditions

Default

EIPBench Pattern Benchmark
[RMSRM2016]; AC:=Apache Camel, TIP:=Vectorization

(A) Normal (B) Branching (C) Conditions

Java/AC TIP/AC
Hardware Acceleration

One step back, important EAI factors:
- Closeness to the network (e.g., connect two applications)
- Expressiveness (e.g., conditions, expressions)
- Efficiency (e.g., volume, velocity)
- Flexibility (e.g., change integration process)
Efficiency through Specialization

Why not use ASICs, GPUs (SIMD, flops, power)?
Why not use SDNs (network, expressiveness: Integration patterns)?

Use FPGAs due to good trade-off between flexibility vs efficiency; designs can become ASICs.

Current Software solutions

Why not use FPGAs?

EAI
VM
Guest OS
Hypervisor
CPU
NIC

Classic SAAS

Software

FPGA

ASIC

More flexible, less efficient...

similar to [Put2017]
What are FPGAs?

- Field Programmable Gate Arrays
- Fabric of interconnected logic blocks, on-chip memory, I/O
- Customize logic and I/O
- Reconfigurable hardware is more efficient than general-purpose hardware (CPU); reconfiguration times 100ms to 1s, partial reconfiguration

  + high degree of parallelism, streaming
  - limited to resources on the chip
Message Throughput (revisited)

Even complex routing (e.g., SP, AGG) and transformation patterns (MT) have throughput close to baseline (i.e., hardware limits).

The throughput is invariant to multiple conditions and route branches (e.g., CBR-B+C, LB and join router (not shown) perform near baseline).
Disruptiveness

... in information systems requires:
(a) Novel types of applications
(b) Novel technology and hardware

Similar to Wolfgang Lehner's Keynote VLDB 2017
Crossroads of Middleware and Hardware

Challenges and Opportunities
EAI Architecture Aspects

Message Endpoints

Operations

Processing Model

Classic SAAS

EAI
VM
Guest OS
Hypervisor
CPU
NIC

Message Endpoints

Programming Model

Message Endpoints

Operations

Processing Model
Programming Models

Integration process modeling, configuration
FPGAs flexible, reconfigurable, became affordable
FPGA development flow, lack the expertise to use the hardware-oriented FPGA, 10:1 or larger ratio of SW to HW programmers; UDFs space critical

Requires:
- Composable HDL / HW templates for building blocks (patterns) [RMRM2017])
- High-level synthesis of conditions / expressions (OpenCL, )
- Better editors and flow (PSHDL, http://pshdl.org/), building and verifying new hardware (incl. debugging)
- Education, Courses

Instance 1
Instance 2
... Complete (24 instances)

Resource usage on the FPGA chip (floorplan): with efficient HDL EAI template design + load balancing, UDFs as high-level synthesis become a dominating factor for multi-instance parallelism
Programming Models

**Memory Access / Bandwidth**
- on-Chip memory accessible in few clock cycles
- Capacity of on-chip memory not enough (flip-flops often required for program logic)

Requires:
- Fast off-chip DRAM memory access (shared with CPU) from the FPGA
- (even Non-volatile Memory)
- Study of optimization techniques (e.g., message indexing [RRM2017] vs. streaming)

Local Integration System, Network EAI

closer to the network, combined data flow architecture and CPU

Requires special HW: FPGA + x

Requires:
- FPGA+CPU multi-chip architectures with direct NIC access
- FPGA on NIC or SmartNIC

New EAI Architecture Variants

Classic SAAS

“Multi-chip” PAAS

“Smart NIC” IAAS

Not just data pipes (logic pushdown)

(e.g., Intel Broadwell, https://www.theregister.co.uk/2016/03/14/intel_xeon_fpga/)
Processing Models

**Message Exchange Patterns**

- **FPGA works well for inOnly streaming**
- **In/out, request-reply in some integration scenarios**

**Transport Protocol Support**

- **stateless transport protocols**
- **IP cores missing (even for UDP), no stateful transport protocols**

**Non-functional aspects**

- **many built-in IP cores, e.g., for network, memory access**
- **vendor specific IP cores incomplete, e.g., for security**

**Requires:**
- Streaming with request-reply (e.g., JMS asynchronous + correlation identifier)
- or Synch-Asynch Bridges (e.g., [RH2015])

**Requires:**
- Vendor IP core support for a broader coverage of protocols like TCP, HTTP, MQTT
- or efficient SW/HW co-design to leverage software protocol implementations (e.g., [YZQFR2011]).

**Requires:**
- Vendor IP core support for non-functional aspects like different types of authentication, encryption
- or efficient SW/HW co-design to leverage software implementations

- e.g., Solace's own network controller
(Cloud) Operations

**Multi-tenancy**
- Tenants separated on HW
- Limited resources on partitioned chip, cross-tenant processing

**Data center impact**
- Low energy, less space
- To be added to datacenter blueprints, good troubleshooting / debugging tools

**HA/DR setup**
- Less prone to failures
- HA requires tenant distributions across different HW, DR across HW and data centers with transactions, adhere to regulations (e.g., data protection)

**Requires:**
- FPGA HW virtualization à la “Configurable Cloud“[Cau2017]
- Solace’s cross virtual provider messaging
- Integration in current cloud platforms
- Regulation-aware, abstracted, configurable HA/DR capabilities
- Similar to Solace’s HA/DR broker

**In general:** SDNs

**Requires:**
- MS Project Catapult
- FPGA Developer AMI
Message Endpoints

**Applications**
- low latency, high-throughput
- Endpoints limited capacities, discrepancy between EAI egress and endpoint ingress rates

Requires:
- End-to-end flow control
- Application scaling
- Asynchronous message processing
- FPGA+RDMA
  - ....
Disruption Potential

Run one integration scenario used by hundreds of customers on just few FPGAs (shared), run all scenarios of one customer on one FPGA, instead of hundreds of VMs

EAI System, Integration Processes

Reduce energy consumption, space

Stable, multi-tenant, HA+DR

New York (active)  Las Vegas (passive)

... more

New application programming models (e.g., asynch, more EAI logic -> idempotent, retries)
Conclusion

• Disruption through novel applications + EAI challenges and hardware + technology
• Specialization with reconfigurable hardware leads to promising future EAI architecture variants
• FPGAs are less well known and harder to program, while problem is not software engineers being able to program FPGAs, but eco-system required
• FPGAs allow for optimizations of both compute and I/O operations, data flow architecture -> think beyond the core application
• This is just the starting point: many new opportunities + further research
References (1/2)


References (2/2)


Thank you

Contact information:
Daniel Ritter
daniel.ritter@sap.com