Understanding Networks-on-Chip: A Decade and Beyond

Radu Marculescu
Carnegie Mellon University
Pittsburgh, PA 15213, USA
NOCS, Tempe, AZ
April 23, 2013

Multicores are critical to IT revolution

Embedded computing (cell phones, automotive, gaming)  Cyber-physical systems (healthcare, transportation, smart grid)

High performance computing (scientific applications, forecasting, data centers)
Multicore platforms are large scale distributed systems at nanoscale; they are dominated by communication costs

Last level cache (LLC)
Memory controller (MC) & channels
I/O controller(s)
QPI controller,
Power control unit (PCU), etc.

Need to understand the behavior of thousand core systems.

Network (routers+links) is the missing link in understanding.

"Knowing in part may make a fine tale, but wisdom comes from seeing the whole."
The rest of this presentation focuses on this “whole”, i.e., a few emerging ideas from a decade of NoC research.

**Structure**
- Architecture and small world effects

**Dynamics**
- Workloads and multiscale behavior

**Control**
- Power and resource management

Our first insight into communication-based design came through architecture (topology, buffer, etc.) optimization.
Can induce small-world effects in regular NoCs. This brings huge performance improvements.

- Completely structured
  \[ b(n) = D(m,n) = m + n - 2 \]
- Customized via LRL
  \[ b(n) = D'(m,n) < m + n - 2 \]
- Complete graph
  \[ b(n) = \log_2 N \]

This way, the fundamental idea of Small World networks (aka “six degrees of separation”) enters the multicore world.

Small world effects can also be exploited to reduce hop count in 3D wireless NoCs and improve performance.

Wired and wireless NoCs can be used intra-chip, while inter-chip communication is based on wireless inductive-coupling.
Flow-control mechanisms, reconfigurability, adaptive routing have all been used to improve performance

One way or another, they all exploit small world effects...

Optimum path-seeking behavior may be goal-oriented. Need to collect fitness information locally, at routing nodes

\[ \text{fitness}_i = \alpha \cdot m_{in} \cdot w_{in}^{-1} \]

\( m = \text{avg. number of free slots} \)
\( w = \text{avg. waiting time in channel} \)
The rest of this presentation focuses on this “whole”, i.e., a few emerging ideas from a decade of NoC research.

Structure
Architecture and small world effects

Dynamics
Workloads and multiscale behavior

Control
Power and resource management

Packet inter-arrival times at interface queues play a fundamental part in network behavior.

Fractals are geometrical objects or stochastic processes displaying self-similar behavior over multiple scales.
Real world processes are not always smooth. Fractional dynamics needs a formalism stronger than integer calculus.

\[ \langle \Delta x \rangle \propto \Delta t^H, \quad 0 < H < 1 \]

\[ \frac{d^\alpha x(t)}{dt^\alpha} = \lim_{\Delta t \to 0} \frac{1}{\Delta t^\alpha} \sum_{j=0}^{[t/\Delta t]} (-1)^j \binom{\alpha}{j} x(t - j\Delta t) \]

Main idea is to exploit workload variations and optimize system behavior (minimize power, latency, etc.)

\[ x(k) = x(k-1) + T\lambda_1 (k-1) - T\mu_1 (k-1) \]

Inter-arrival times dictate the fractal exponent \( \alpha_k \) of state equations; this is used to characterize the system dynamics.
Performance analysis at high injection rates needs a non-equilibrium approach that accounts for fractal behavior.

High injection rates cause inter-arrival times deviate from exponential distribution and exhibit power law correlations.

Network understanding needs to evolve from deterministic and stochastic, to statistical physics type of approaches.

System optimization and resource management becomes all about time-varying dynamic processes running over networks.
The rest of this presentation focuses on this “whole”, i.e., a few emerging ideas from a decade of NoC research

Structure
Architecture and small world effects

Dynamics
Workloads and multiscale behavior

Control
Power and resource management

8-Core Xeon® Processor has three clock domains and three voltage domains that help minimizing power.

Largest device count reported for a microprocessor 2.3B transistors.

[Rusu, ISSCC 2009]
Fine-grain power management becomes possible by exploiting workload variations.

Globally asynchronous, locally synchronous (GALS)

Clock domain 1
Volt./Freq. Island VFI 1
$(V_1, f_1)$

Clock domain 2
VFI 2 $(V_2, f_2)$

Fine-grain power management can be implemented via control-theoretic approaches.

Voltage-Frequency Controller

$X^{ref} = [x_1^{ref}, x_2^{ref}, ..., x_{N_f}^{ref}]^T$

Physical constraints
$V_1, f_1 \in [f_{1\min}, f_{1\max}]$

$V_{N_r}, f_{N_r} \in [f_{N_r\min}, f_{N_r\max}]$

V/F interface FIFO

Actual utilization of interface FIFOs
$X = [x_1, x_2, ..., x_{N_f}]^T$

State feedback
$x_k^{min} \leq x_k \leq x_k^{max}$

Applications (workload)

Workload characteristics and application deadlines are critical.
Also, control signals (voltage, frequency) need to be constrained.
V/F controller selects the minimum operating frequencies s.t. the queues reference values are satisfied

\[ \frac{d}{dt} y_i(t) = a_i(t) y_i(t) + b_i(t) f_i(t) - c_i(t) f_i(t), \]

\[ 0 \leq y_i^{\min} \leq y_i(t) \leq y_i^{\max} \leq 1, \quad i = 1 + N_{PE} \]

\[ \frac{d}{dt} x_k(t) = a_k(t) x_k(t) + b_k(t) f_j(t) - c_k(t) f_j(t), \]

\[ 0 \leq x_k^{\min} \leq x_k(t) \leq x_k^{\max} \leq 1, \quad k = 1 + 4, \quad j, l = 1 + N_r \]

\[
\min \left\{ \sum_{i=1}^{N_{PE}} \left[ \frac{1}{2} w_i \left( y_i(t) - y_i^{\text{ref}} \right)^2 + \frac{1}{2} z_i f_i(t)^2 \right] \right\} \int dt
\]

\[
\sum_{j=1}^{N_r} \left[ \frac{1}{2} r_j f_j(t)^2 + \sum_{k=1}^{N_{PE}} \left[ \frac{1}{2} q_{jk} \left( x_k(t) - x_k^{\text{ref}} \right)^2 \right] \right] \int dt
\]

\[ f_i^{\min} \leq f_i(t) \leq f_i^{\max}, \quad i = 1 + N_{PE}, \quad \text{and} \quad f_j^{\min} \leq f_j(t) \leq f_j^{\max}, \quad j = 1 + N_r \]

Accurate mathematical modeling and rigorous optimization can enable cross-layer power management

Queues utilization at tiles (0,0), (1,1) and (1,2) for a 4×4 mesh NoC running Apache HTTP webserver application

FOC keeps the utilization of all queues below 0.1 by adjusting the operating frequencies of all PEs and routers. FOC consumes less power compared to LQR controller
Workload analysis should not be an afterthought. In real applications, traffic is rarely Poisson or stationary

Classical dynamics: Linear Dependence & Exponential Inter-Event Distribution

\[
\frac{dP(a,t)}{dt} \propto P(a,t)
\]
\[
\frac{dM_1(t)}{dt} \propto M_1(t)
\]

Fractal dynamics: Linear Dependence & Power-Law Inter-Event Distribution

\[
\frac{d^{\alpha} [tP(a,t)]}{dt^{\alpha}} \propto P(a,t)
\]
\[
\frac{d^{\alpha} [tM_1(t)]}{dt^{\alpha}} \propto M_1(t)
\]

Statistical properties of the workload have deep implications in resource allocation, architectural design, scheduling, etc.

For thousand core systems distributed approaches for power management are of crucial importance

15-router synch NoC that connects 22 processing units

[F. Clermidy et al., ISSCC’10]
Both centralized and distributed approaches have their own limitations

- Scalability issues due to cost and latency of long wires. Long synchronization times
- Highly scalable but potential problems in control performance

Need ‘best-of-both-worlds’ between fully-centralized and fully-distributed solutions. This is true for thermal management too.

Custom feedback control offers a good compromise between fully centralized and fully distributed solutions

Unexplored Design Space

Implementation cost

[S. Garg et al., ISLPED 2010]
An hierarchy of globally distributed locally centralized control may help the system self-organize

Local control w/ full state information, global control w/ partial information. Small world effects help convergence

<table>
<thead>
<tr>
<th>System Size</th>
<th>Flat Mesh (nJ)</th>
<th>WiNoC (nJ)</th>
<th>Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>1319</td>
<td>22.57</td>
<td>58x</td>
</tr>
<tr>
<td>256</td>
<td>2936</td>
<td>24.02</td>
<td>122x</td>
</tr>
<tr>
<td>512</td>
<td>4992</td>
<td>37.48</td>
<td>133x</td>
</tr>
</tbody>
</table>

[Orders of magnitude! Gets better with size]

[Current: Monolithic (single kernel) OS (e.g., Windows)
User + Applications + Services
Single shared state & lock OS
Core 1 Core 2 Core 2

Future: Distributed (multi-kernel) OS (e.g., Barrelfish)
User 1 + Applications 1 \ldots User p + Applications p
Services
OS 1 state replica \ldots OS m state replica
Core 1 Core 2 \ldots Core n

On-chip interconnect

OS execution as dynamic graphs. Fundamental design constraints (power, performance) depend on HW-OS-application interaction.

On-chip communication is essential for multi-kernel OS

OS inter-event times exhibit a self-similar structure over time. Hurst exponent for some real OS traces is 0.9.

In summary, thousand core systems offer ample opportunities to bring science and engineering even closer.

Workload analysis should not be an afterthought, from chip all the way to the cloud.

Application/OS optimization and resources management represent the next frontier to conquer.

Optimization evolves towards self-organization; new theories and design paradigms are needed.

Understanding networks (i.e. “whole”) is crucial for sci&eng. Lessons learned so far are valuable as we move forward.
Finally...

Collaborators - Paul Bogdan (USC) and Umit Y. Ogras (Intel)

Siddharth Garg (Univ. of Waterloo), Mike Kishinevsky (Intel), Edward Ma (CMU), Diana Marculescu (CMU), Hiroki Matsutani (Keio Univ.), Chi-Ying Tsui (HK Univ. Sci. Tech), Qian Zhiliang (HK Univ. Sci. Tech), Simon Hollis (Univ. of Bristol),

Relevant papers  - www.ece.cmu.edu/~sld

Sponsors

National Science Foundation

Intel Corporation

Questions?

Thank you!