Single-Chip Cloud Computer

Adam Welc, Richard L. Hudson, Tatiana Shpeisman, Ali-Reza Adl-Tabatabai, and the SSC team

Intel Labs
HW Architect’s Point of View

- 6x4 mesh 2 Pentium™ P54c cores per tile
- 256KB L2 Cache, 16KB shared MPB per tile
- 4 memory controllers, 16-64 GB total memory

Tile area: ~17 mm²
SCC die area: ~567 mm²

Single-Chip Cloud Computer
Adam Welc – Intel Labs, Programming Systems Lab
Programmer’s Point of View

• 48 x86 cores capable of running a full Linux distribution
• 3 memory spaces (☐/☐: on/off-chip)

shared memory (variable size)

private memory $\frac{1}{2}$ $\frac{1}{2}$ CPU_0 $\frac{1}{2}$ $\frac{1}{2}$
t&s

private memory $\frac{1}{2}$ $\frac{1}{2}$ CPU_47 $\frac{1}{2}$ $\frac{1}{2}$
t&s

shared message passing buffer (8KB*48 cores)

t&s = shared test and set register
SCC Features

• Memory spaces:
  – Shared – not cached or non-coherent
  – Private – x86 memory model
  – MPB – non-coherent (bypasses L2 and requires invalidation of L1 for valid reads and writes)

• Communication
  – RCCE – message passing library utilizing MPB
  – Full TCP/IP stack (on-chip and host/device)

• Four-tile power management domains
  – V change – millions of cycles (non-blocking API)
  – GhZ change – few cycles (blocking API)
Case Study: JavaScript

• Object-oriented dynamically typed scripting language

• Limited support for parallelism
  – Web workers (in HTML 5) designed to increase GUI responsiveness
  – Web workers can communicate with HTTP servers via message passing

• SCC viewed as a cluster – utilized only high-level capabilities
Parallelizing JavaScript on SCC

• Offload computation from the client (browser) to the server farm on SCC
• Utilize as many off-the-shelf components as possible for high productivity
  – Client and server code written in pure JavaScript
  – Unmodified client (browser) running on host
  – Largely unmodified off-the-shelf execution engine for servers running on SCC
  – Standard libraries and tool-chain used on SCC
Web App Architecture

- HTTP server’s scripting engine typically used for dynamic web page generation
- Can be used for general-purpose computation as well

BROWSER

WEB WORKERS

JS SCRIPTING ENGINE (v8)

content request

web page

SCRIPTING ENGINE
JS (V8), PHP, ...

HTTP SERVER
Enabling SCC

BROWSER
- WEB WORKERS
- JS SCRIPTING ENGINE (v8)

content request

compute request

compute response

scripting engine
- JS (V8), PHP, ...

http server

web page

PCIe

SCC

Single-Chip Cloud Computer
Adam Welc – Intel Labs, Programming Systems Lab
Workers: Web ➔ Generic

BROWSER

- GENERIC WORKERS
- JS SCRIPTING ENGINE (v8)

content request

compute request
compute response

PCiE

HTTP SERVER

SCRIPTING ENGINE
JS (V8), PHP, ...

web page

HTTP SERVER

SCRIPTING ENGINE
JS (V8), PHP, ...

GENERIC WORKERS

SCC

Single-Chip Cloud Computer
Adam Welc – Intel Labs, Programming Systems Lab
Compute Servers

BROWSER
- GENERIC WORKERS
- JS SCRIPTING ENGINE (v8)

content request
web page

SCRIPTING ENGINE
JS (v8), PHP, ...

HTTP SERVER

compute request
compute response

PCIe

SCC

JS SCRIPTING ENGINE (v8)
- GENERIC WORKERS
- COMPUTE SERVER

Single-Chip Cloud Computer
Adam Welc – Intel Labs, Programming Systems Lab
Infrastructure

- Generic workers on both client and server side programmed in JavaScript
- Minor modifications to v8cgi wrapper around Google’s v8 execution engine for support of compute servers on SCC
- Unmodified browser can only “talk” HTML
  - Compute servers “pretend” to be HTML servers (communication layer written in JavaScript)
  - Problem with “single-origin policy”
Parallel Raytracer

- Based on sequential JavaScript app from Google’s JavaScript V8 benchmark suite
- Workers all 48 cores
- Also tried different configurations (single dispatcher core) and applications (physics engine) using the same infrastructure
Results

Concurrent Raytrace - Direct

Parallelized at PSL based on sequential raytrace benchmark from Google JavaScript V8 benchmark suite, v.6

2500 pixels 19600 pixels 119025 pixels

1 worker 8 workers 48 workers

x1 x4 6 x21.2

(Total: 3.42 s) (Total: 5.87 s) (Total: 7.69 s)
Conclusions

- SCC features
  - 48 cores
  - Non-coherent shared memory with message passing as primary programming model
  - Extensive power management capabilities

- Low learning curve for potential SSC software developers:
  - Standard tool-chain
  - Off-the-shelf components