

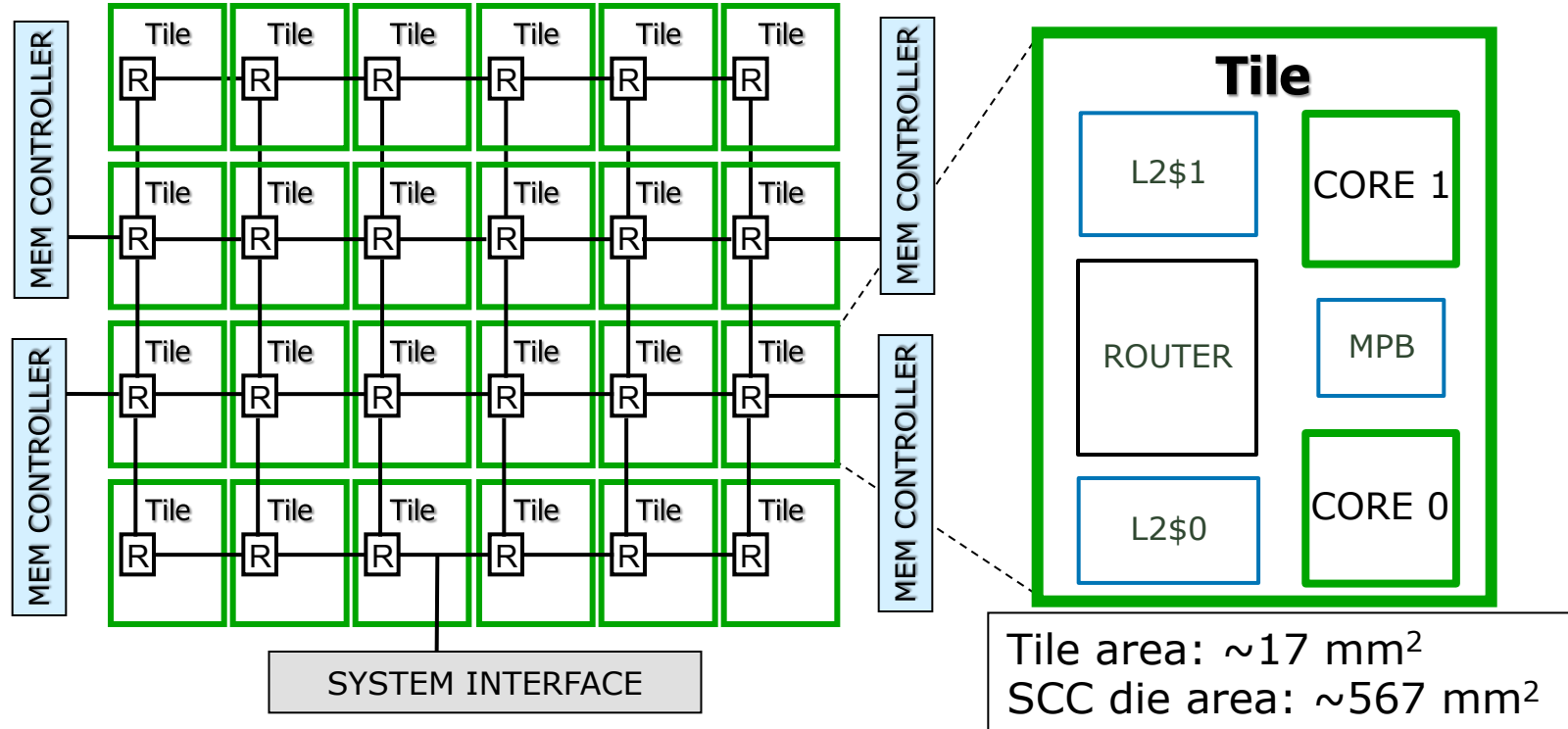
# Single-Chip Cloud Computer

**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





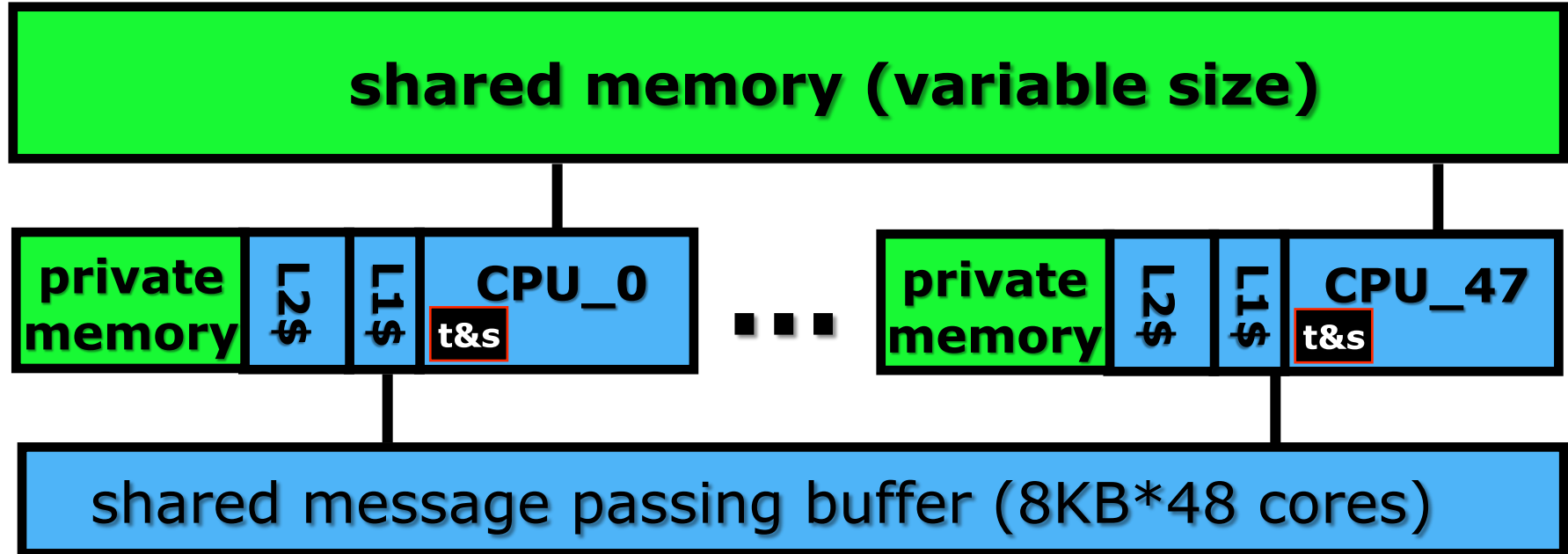
Single-Chip Cloud Computer

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)



**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)

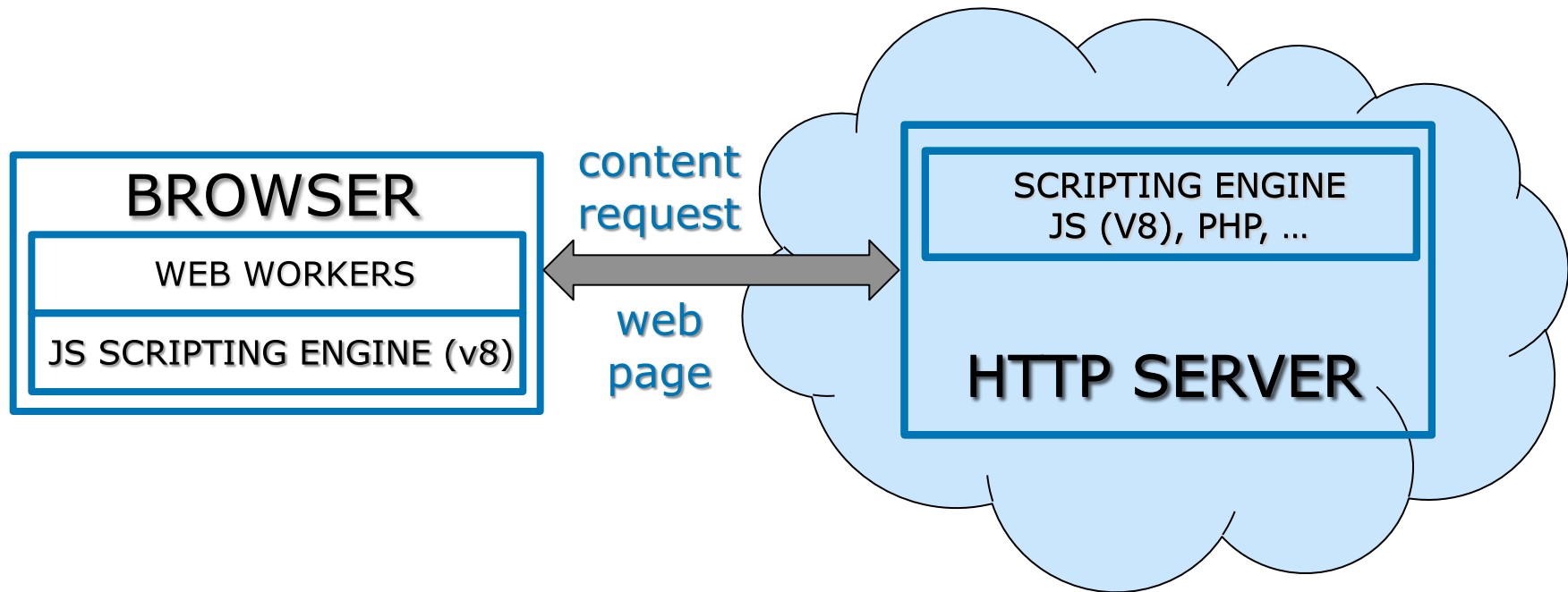
# Case Study: JavaScript

- 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

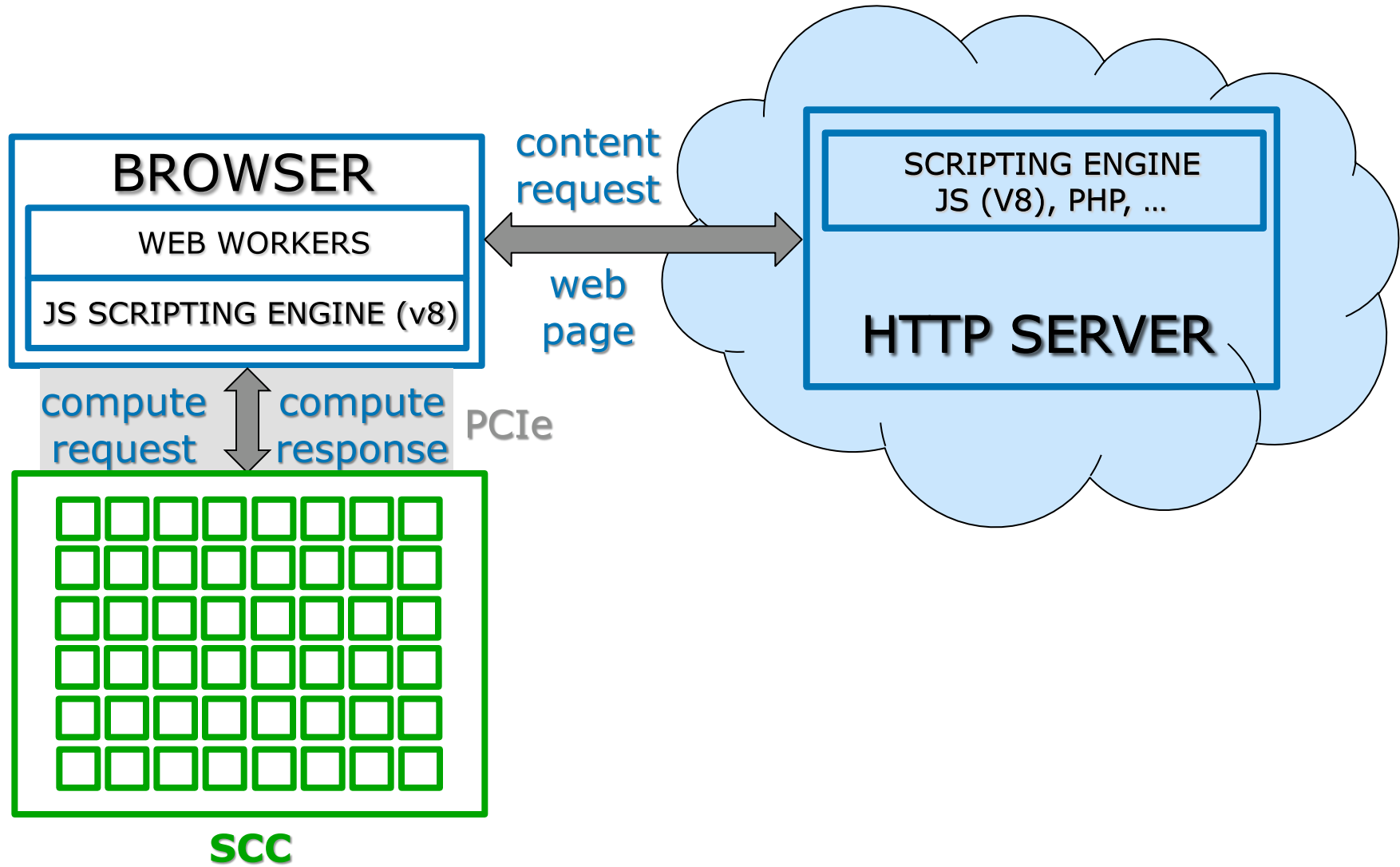
- Offload computation from the client (browser) to the server farm on SCC
- Utilize as many off-the-shelf components as possible for high productivity
  - Client and server code written in pure JavaScript
  - Unmodified client (browser) running on host
  - Largely unmodified off-the-shelf execution engine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

# Web App Architecture



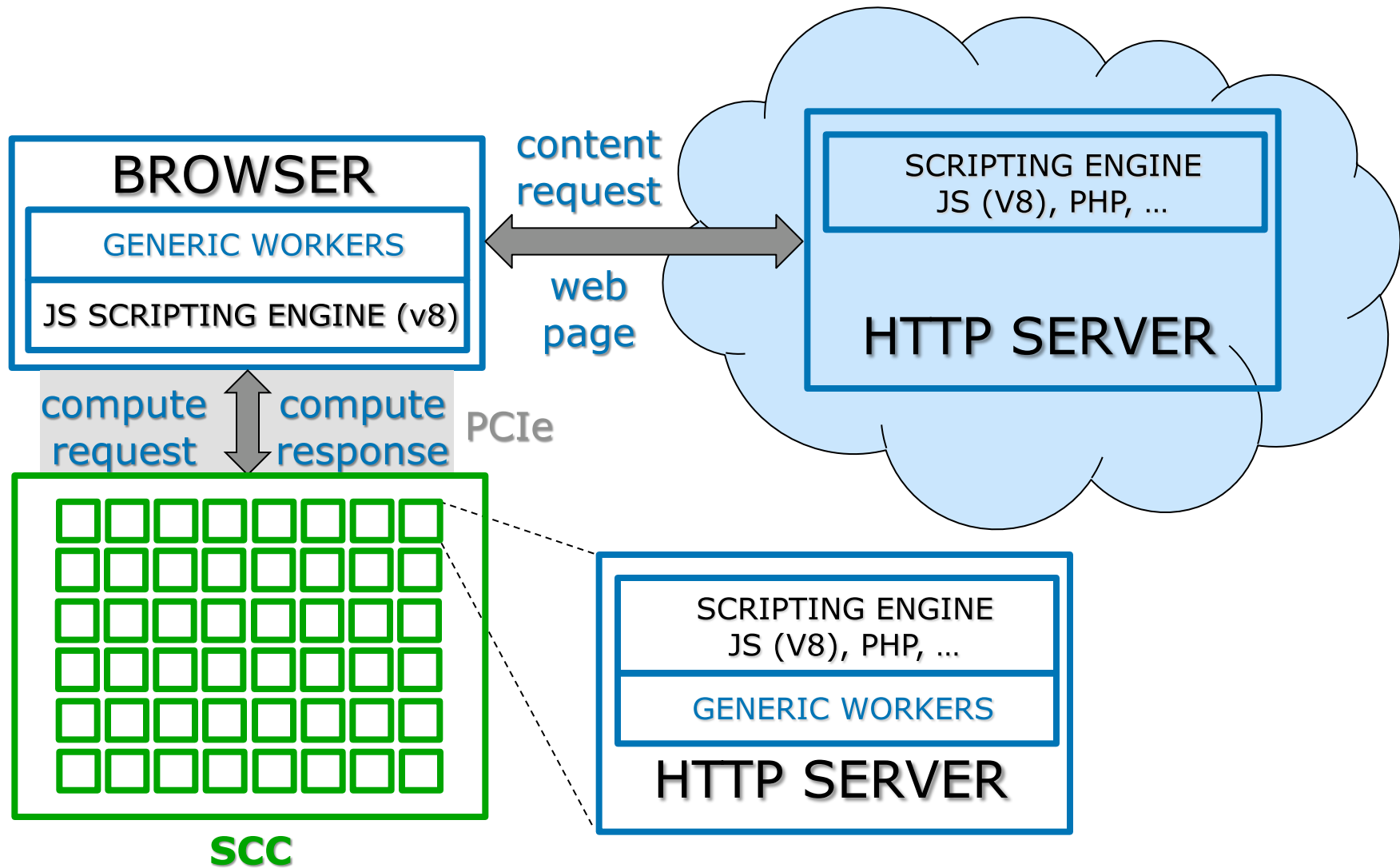
- HTTP server's scripting engine typically used for dynamic web page generation
- Can be used for general-purpose computation as well

# Enabling SCC





# Workers: Web → Generic

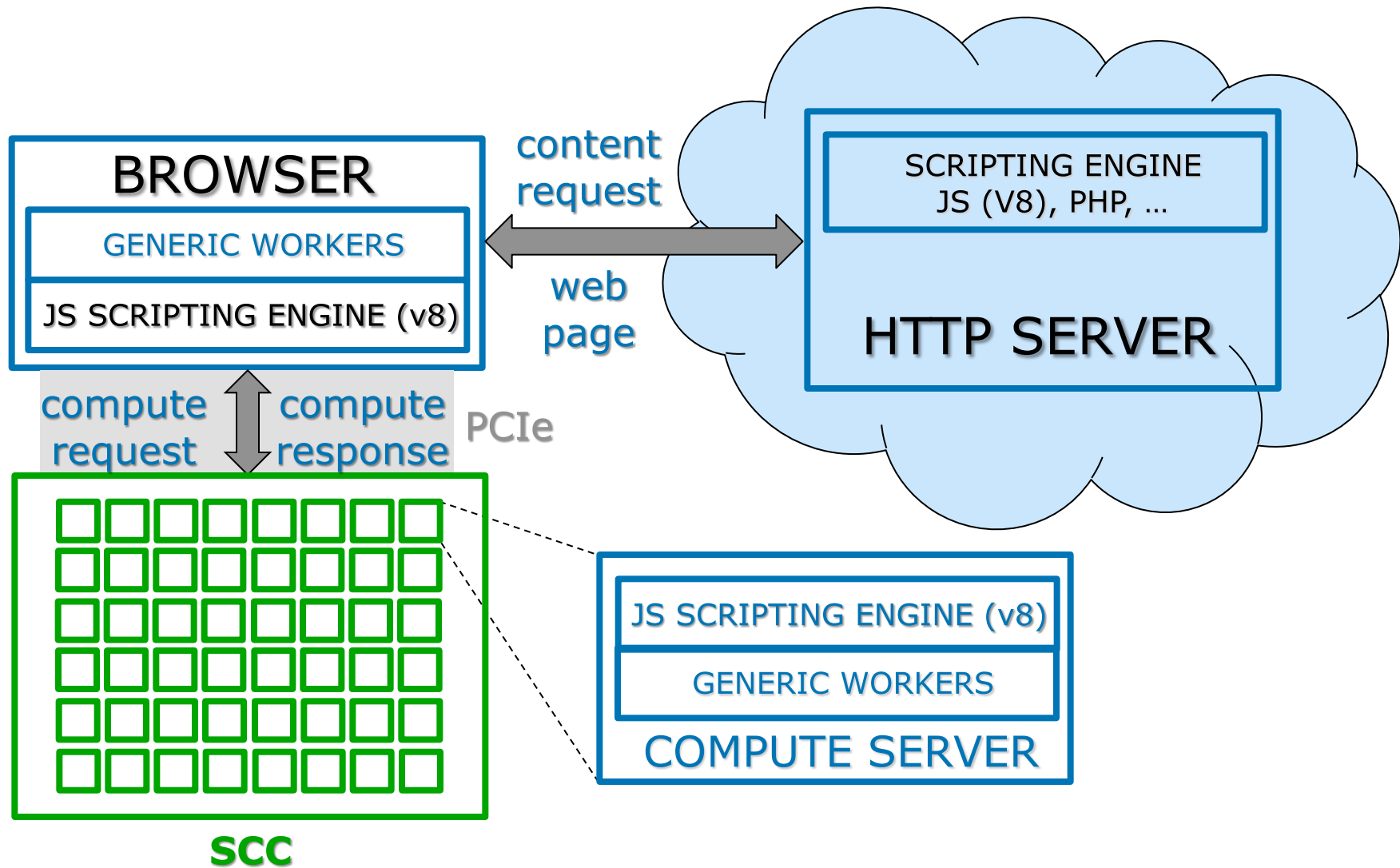


Single-Chip Cloud Computer

Adam Welc – Intel Labs, Programming Systems Lab



# Compute Servers



Single-Chip Cloud Computer

Adam Welc – Intel Labs, Programming Systems Lab

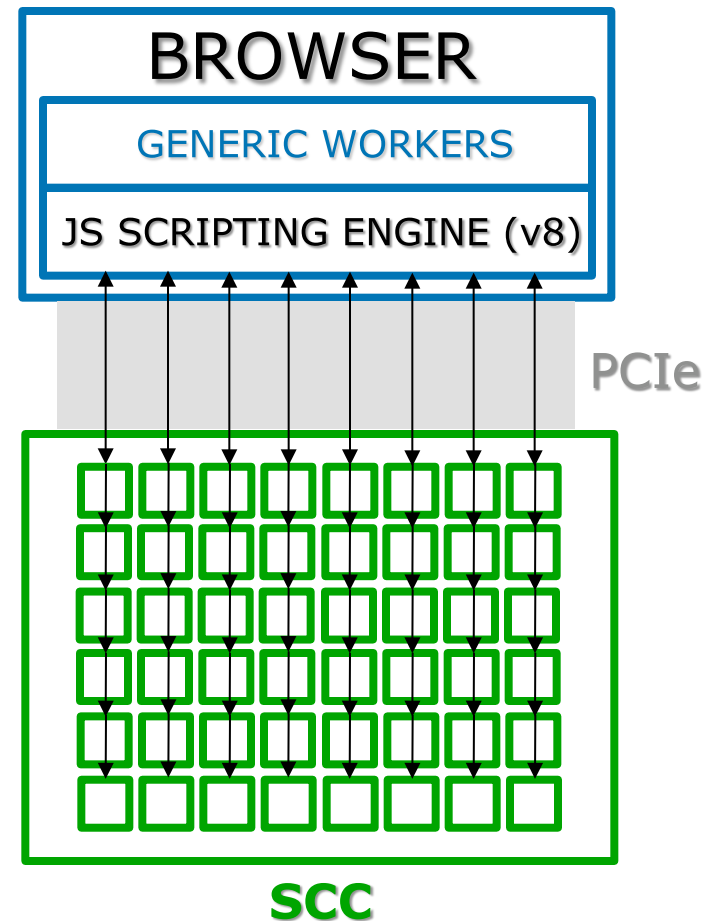


# Infrastructure

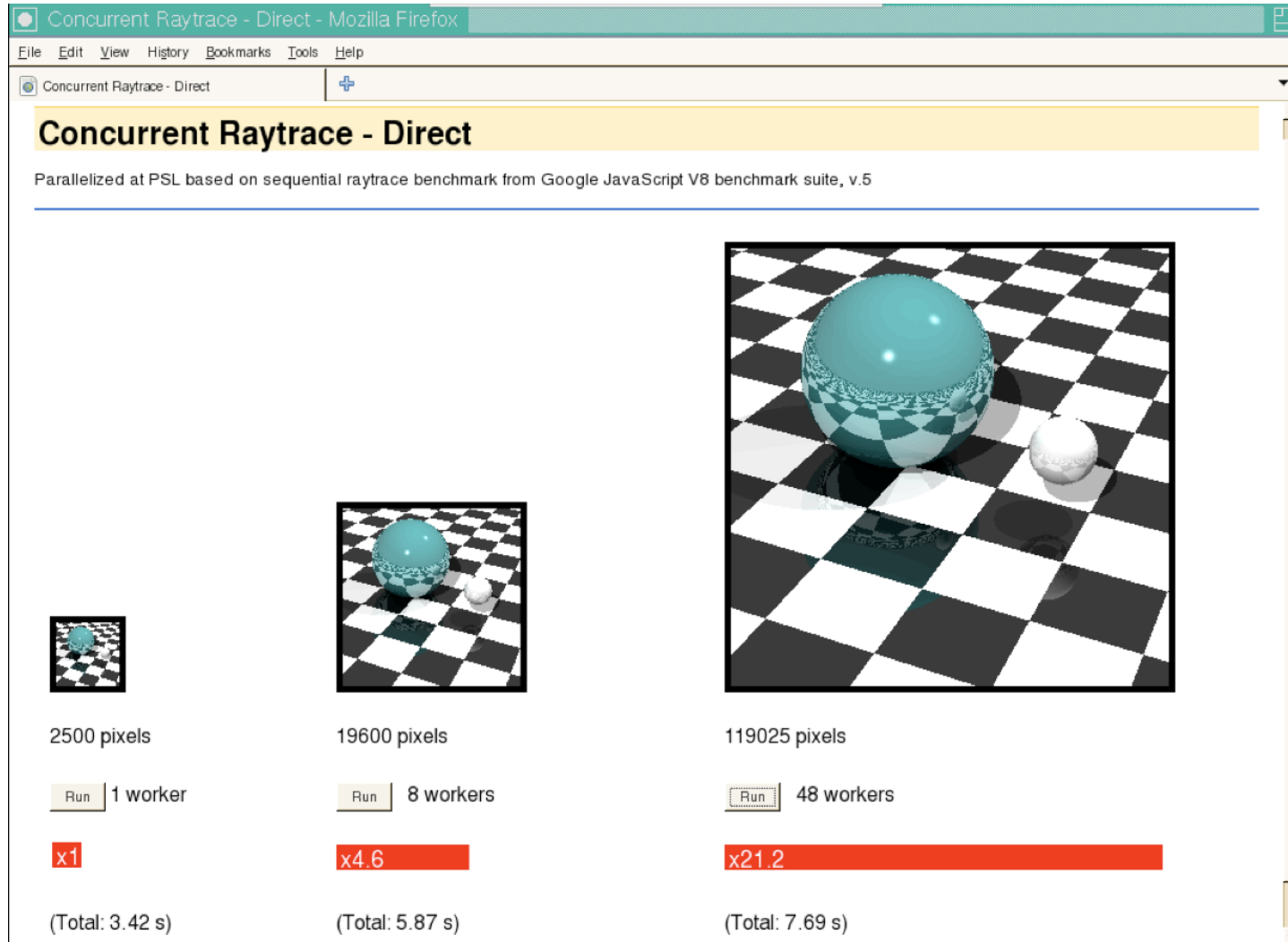
- Generic workers on both client and server side programmed in JavaScript
- Minor modifications to v8cgi wrapper around Google's v8 execution engine for support of compute servers on SCC
- Unmodified browser can only "talk" HTML
  - Compute servers "pretend" to be HTML servers (communication layer written 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



Single-Chip Cloud Computer

Adam Welc – Intel Labs, Programming Systems Lab



# 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

