IBM POWER8 Processor, NVIDIA P100 GPU and IBM Minsky Node Hardware Architecture

SC17 Tutorial

13.11.2017 | Dirk Pleiter | Forschungszentrum Jülich/JSC & University of Regensburg

Goals for Today

- Obtain knowledge about Minsky heterogeneous server platform including it's
  - IBM POWER8 processor
  - NVIDIA P100 processor
- Obtain skills to analyse application for attainable performance
- Learn and practice measuring performance
- Learn and exercise how to optimise a simple application on IBM POWER8 processors and NVIDIA P100 GPUs
- Learn and practice parallelisation on multiple GPUs
- Study best practices for porting scientific applications

Overview of the Tutorial

Lecture	IBM POWER8 processor, NVIDIA P100 GPU and IBM Minsky node hardware architecture
Lecture + hands-on	Performance counters and tools
Lecture + hands-on	Strategies to improve application performance on POWER8
Lecture + hands-on	P100 GPU programming and optimization Multi-GPU programming
Lecture	Best practices for porting scientific applications


D. Pleiter


A. Herten


B. Messer


A. Ravindar


J. Kraus


C. Hagleitner

Overview of this Lecture

- Introduction
- IBM POWER8 Processor
- NVIDIA P100 GPU
- IBM Minsky Platform
- Use Case
- Summary

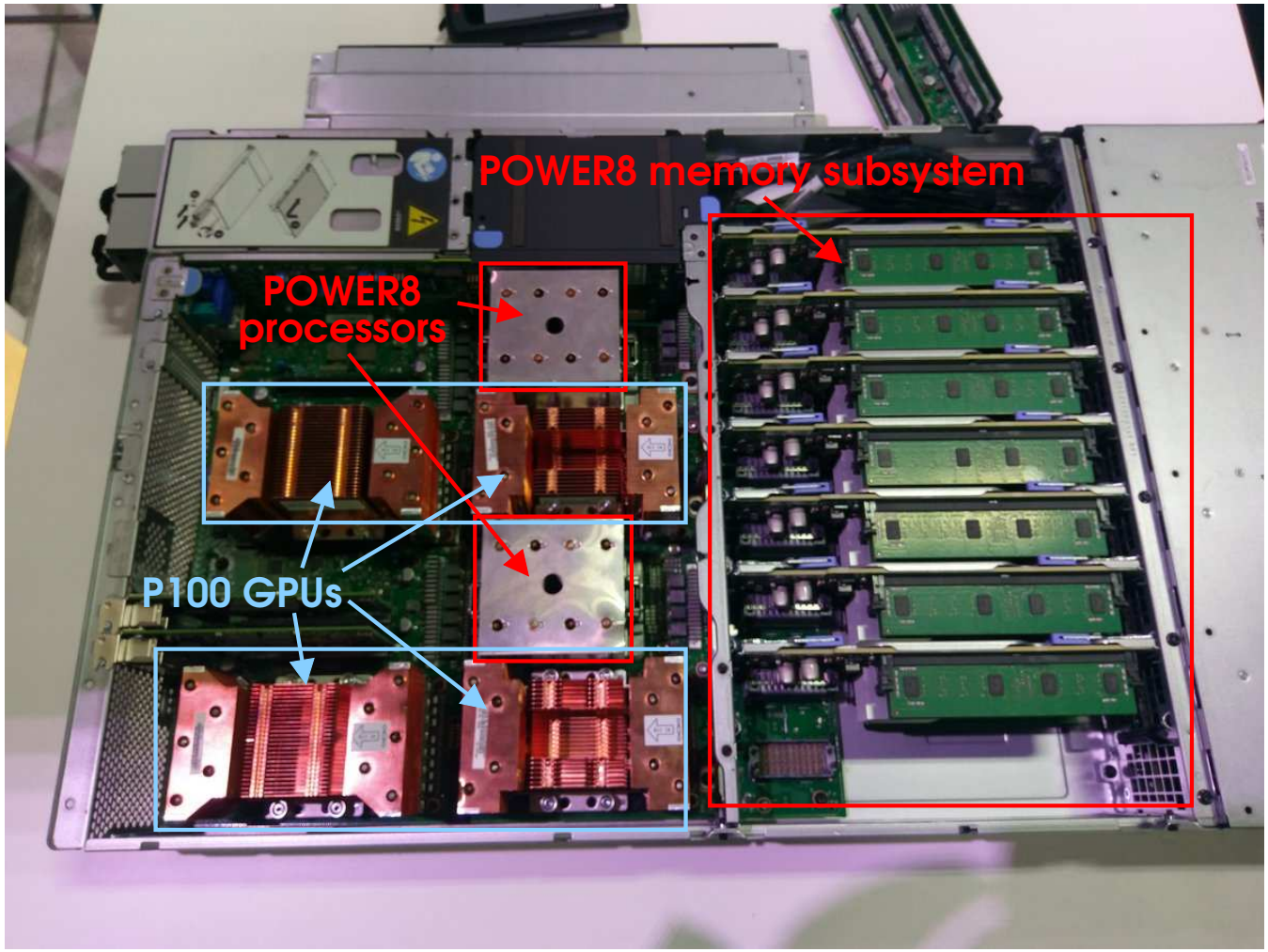
JÜLICH
FORSCHUNGSZENTRUM

IBM POWER8 Processor, NVIDIA P100 GPU and IBM Minsky Node Hardware Architecture

Part I: Introduction

13.11.2017 | Dirk Pleiter | Forschungszentrum Jülich/JSC & University of Regensburg

Minsky Platform: Overview



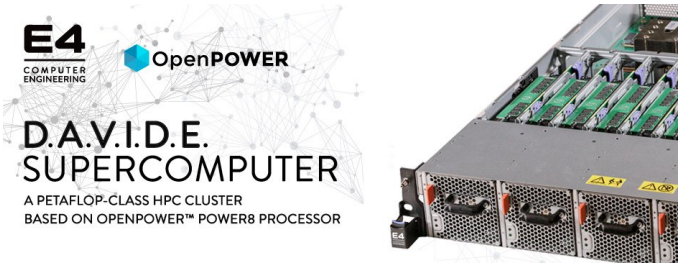
[Gábor Samu, 2017]

OpenPOWER for HPC

JURON @ JSC
18 Minsky nodes
0.35 PFlop/s

⇓

D.A.V.I.D.E @ CINECA
45 Minsky nodes
0.9 PFlop/s



⇓

SUMMIT @ ORNL
4600 Witherspoon nodes
>180 PFlop/s

Information Exchange Function

- A computation implies that information is transferred from a storage device x to a storage device y.
- **Information Exchange Function**:

$$I^k_{x,y}(W) \quad = \quad \text{data transferred between computer sub-systems for specific computation } k$$

x ... source storage device (e.g. memory)
y ... destination storage device (e.g. register file)
W ... **problem size/work-load**

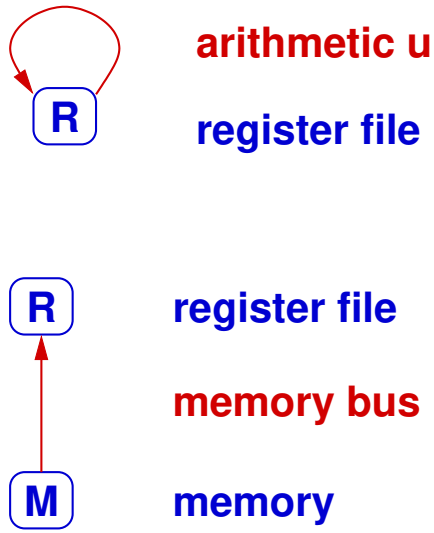
Bandwidth-Latency Performance Model

Ansatz to predict latency:

$$\Delta t^k_{x,y} \simeq \lambda_{x,y} + I^k_{x,y}/\beta_{x,y}$$

Examples:

- Throughput of arithmetic operations
  - x, y = R
  - $\beta_{\mathrm{R,R}}$ = throughput arithmetic unit
  - $I^k_{\mathrm{R,R}}$ = number of operations
- Load of data
  - x = M, y = R
  - $\beta_{\mathrm{M,R}}$ = memory bandwidth
  - $I^k_{\mathrm{M,R}}$ = amount of data



Balanced Architecture

- Assume perfect overlap of computation and data transport
- Condition for balanced architecture

$$
\begin{aligned}
\Delta t_{\mathrm{fp}} &\simeq \Delta t_{\mathrm{mem}} \\
&\Downarrow \\
\frac{I_{\mathrm{fp}}}{\beta_{\mathrm{fp}}} &\simeq \frac{I_{\mathrm{mem}}}{\beta_{\mathrm{mem}}} \\
&\Downarrow \\
\frac{\beta_{\mathrm{fp}}}{\beta_{\mathrm{mem}}} &\simeq \frac{I_{\mathrm{fp}}}{I_{\mathrm{mem}}} = \mathrm{AI}
\end{aligned}
$$

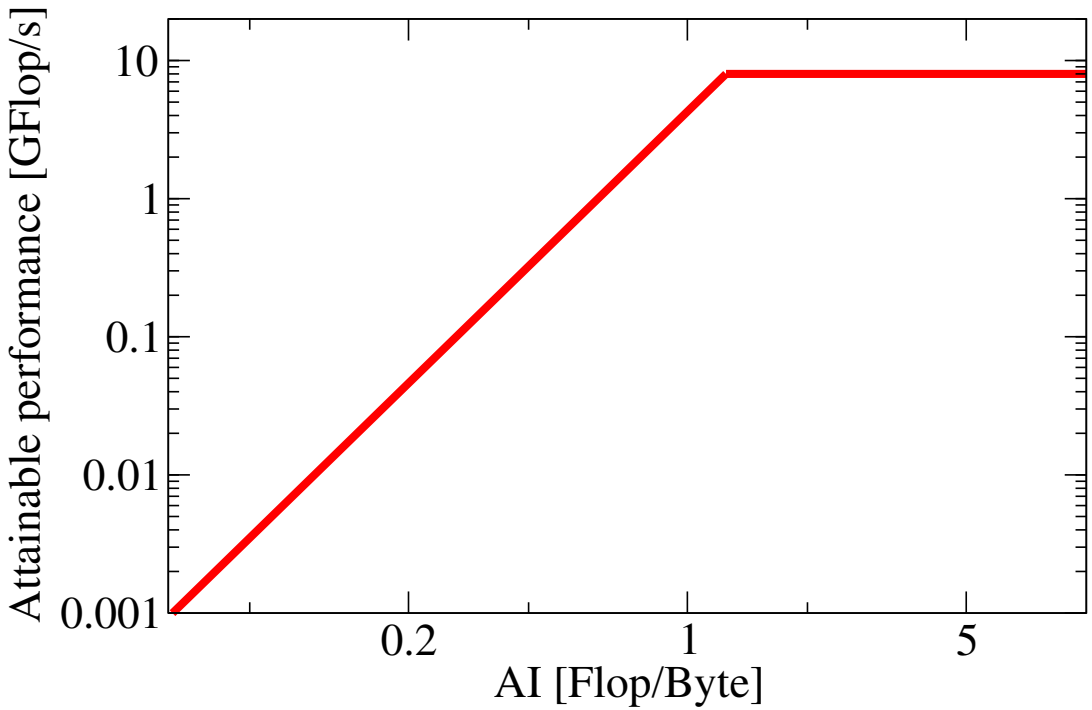
Arithmetic Intensity

- Unbalanced cases
  - $\mathrm{AI} < \beta_{\mathrm{fp}}/\beta_{\mathrm{mem}} \Rightarrow$ **memory bandwidth limited**
  - $\mathrm{AI} > \beta_{\mathrm{fp}}/\beta_{\mathrm{mem}} \Rightarrow$ **compute performance limited**

Roofline Model

Attainable performance

$$b_{\mathrm{fp}} < \frac{I_{\mathrm{fp}}}{\max(\Delta t_{\mathrm{fp}}, \Delta t_{\mathrm{mem}})} \simeq \min(\beta_{\mathrm{fp}}, \mathrm{AI} \cdot \beta_{\mathrm{mem}})$$



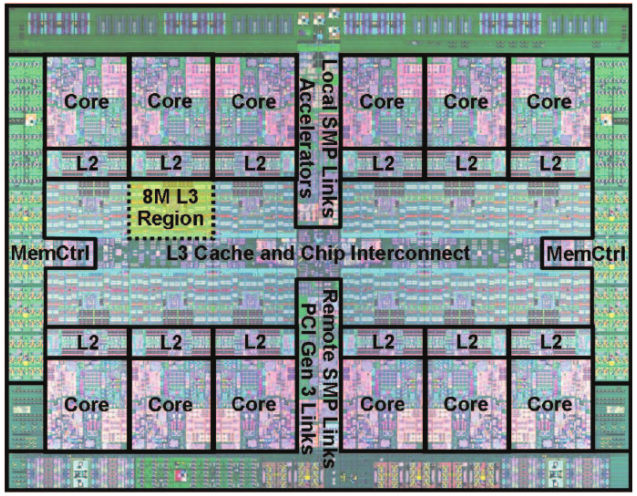
IBM POWER8 Processor, NVIDIA P100 GPU and IBM Minsky Node Hardware Architecture

Part II: IBM POWER8 Processor

13.11.2017 | Dirk Pleiter | Forschungszentrum Jülich/JSC & University of Regensburg

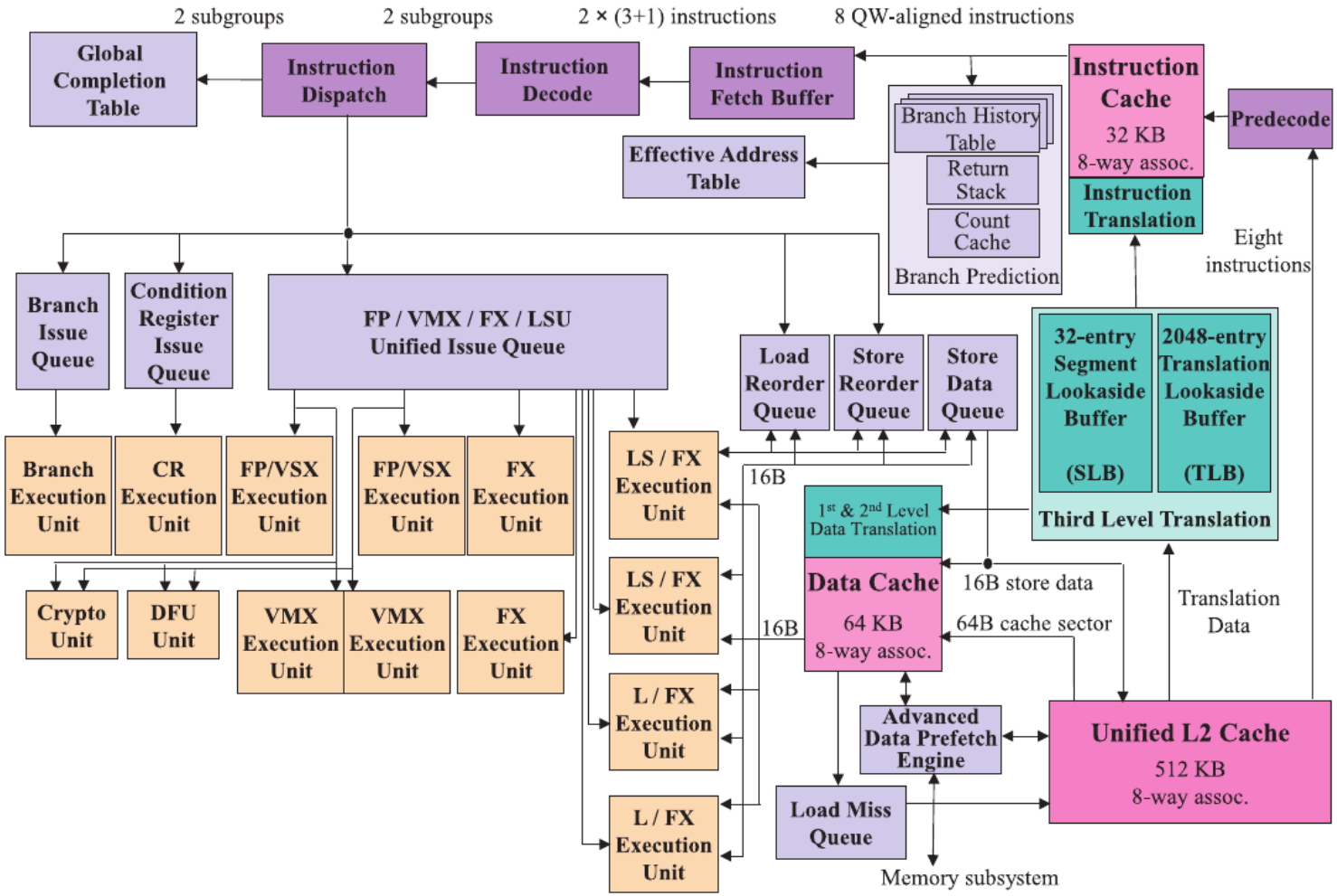
POWER8 Overview

- Introduced in 2014 using 22nm technology
- POWER v2.07 ISA
- Up to 12 cores, frequency 2.5-5.0 GHz
- Core performance figures
  - Double precision floating-point arithmetics:
    $2 \cdot 2$ FMA/cycle = 8 Flop/cycle
  - Load from L1 or L2:
    $4 \cdot 16$ Byte/cycle
  - Store to L1 or L2:
    16 Byte/cycle



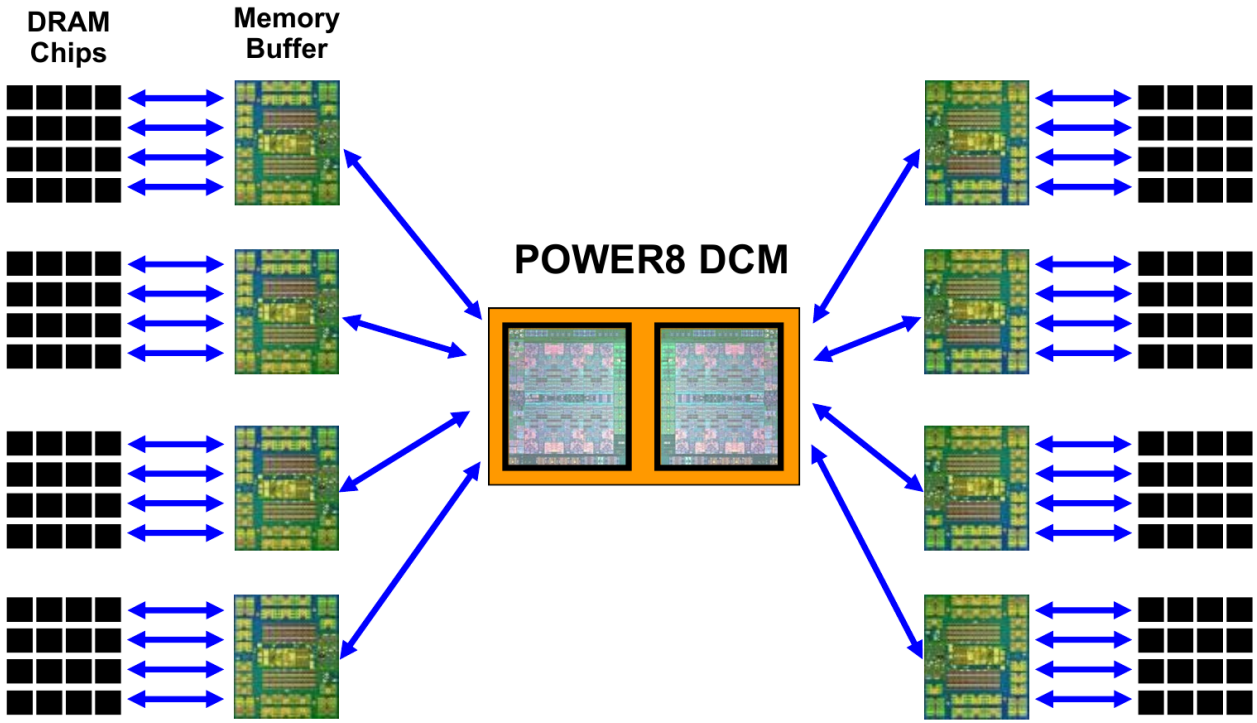
POWER8 Micro-Architecture

[IBM, 2015]



POWER8 Memory Subsystem

[IBM, 2014]



Memory buffer features:

- 4 DRAM interfaces, 16 MiByte L4 buffer cache
- 2+1 Byte link to+from processor, 9.6 Gbit/s data rate

POWER8 Memory Hierarchy

- Cache line size = 128 Byte
- L1 data cache
  - 64 kByte, 8-way set associative
  - Write update policy: store-through
  - Core private
- L2 data cache
  - 512 kByte, 8-way set associative
  - Fully inclusive of the L1 cache
  - Core private
- L3 data cache
  - 8 MByte region per core, 8-way set associative
  - Victim cache, not inclusive of the L2 cache
  - Accessible from other cores

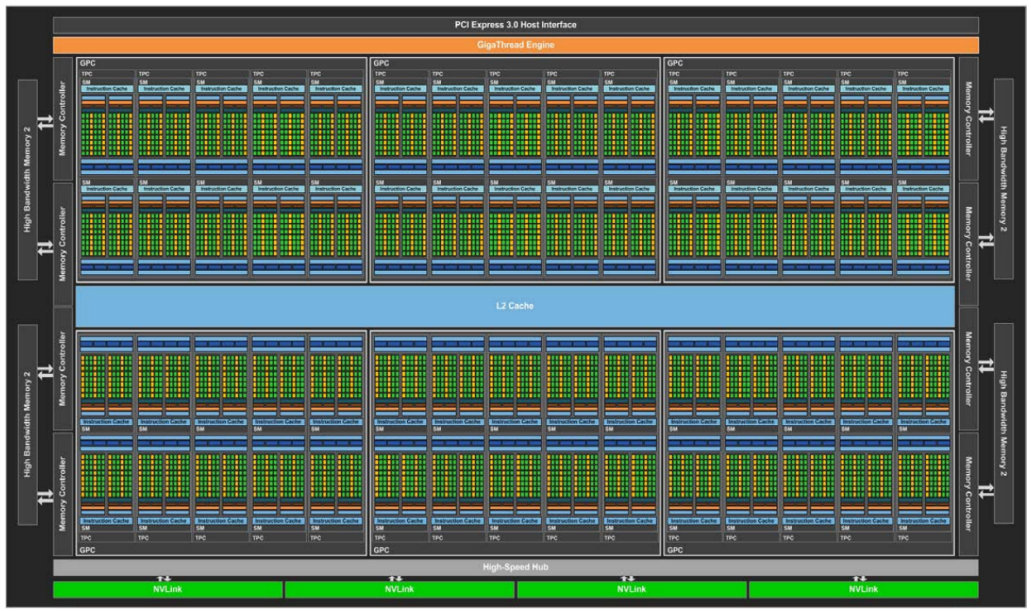
IBM POWER8 Processor, NVIDIA P100 GPU and IBM Minsky Node Hardware Architecture

Part III: NVIDIA P100 GPU

13.11.2017 | Dirk Pleiter | Forschungszentrum Jülich/JSC & University of Regensburg

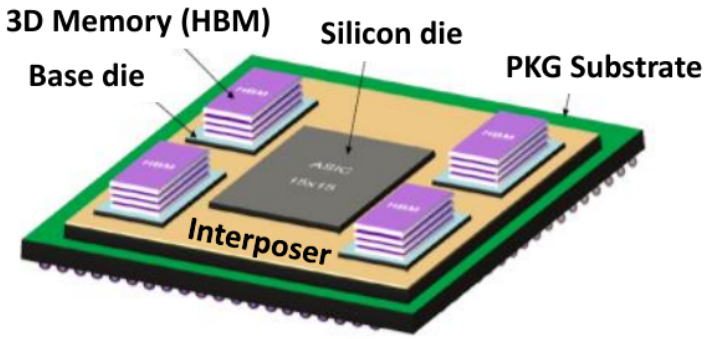
Pascal Architecture Overview

- Introduced in 2016 using 16 nm and 14 nm
- Up to 56 Streaming Multiprocessors (SM)
- SM performance figures
  - 32 FP64 CUDA “cores”
  - 64 Flop/cycle

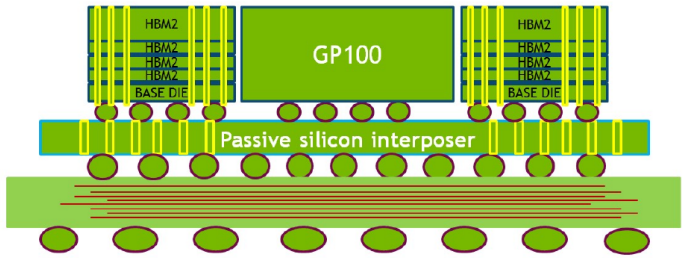


P100 Memory Subsystem

- In-package HBM2 memory
  - 4 memory stacks, 4 GByte/stack
  - Very wide bus ($2 \cdot 512$ bit) and relative low data rate (1.43 Gbit/s)
  - Aggregate bandwidth: 732 GByte/s
- On-chip memory
  - Aggregate register file size: 256 kByte/SM
  - L1 cache: coalescing buffer for memory accesses
  - Shared memory: 64 kByte/SM
  - L2 cache: 4096 kByte



[H. Jun (SK Hynix), 2013]



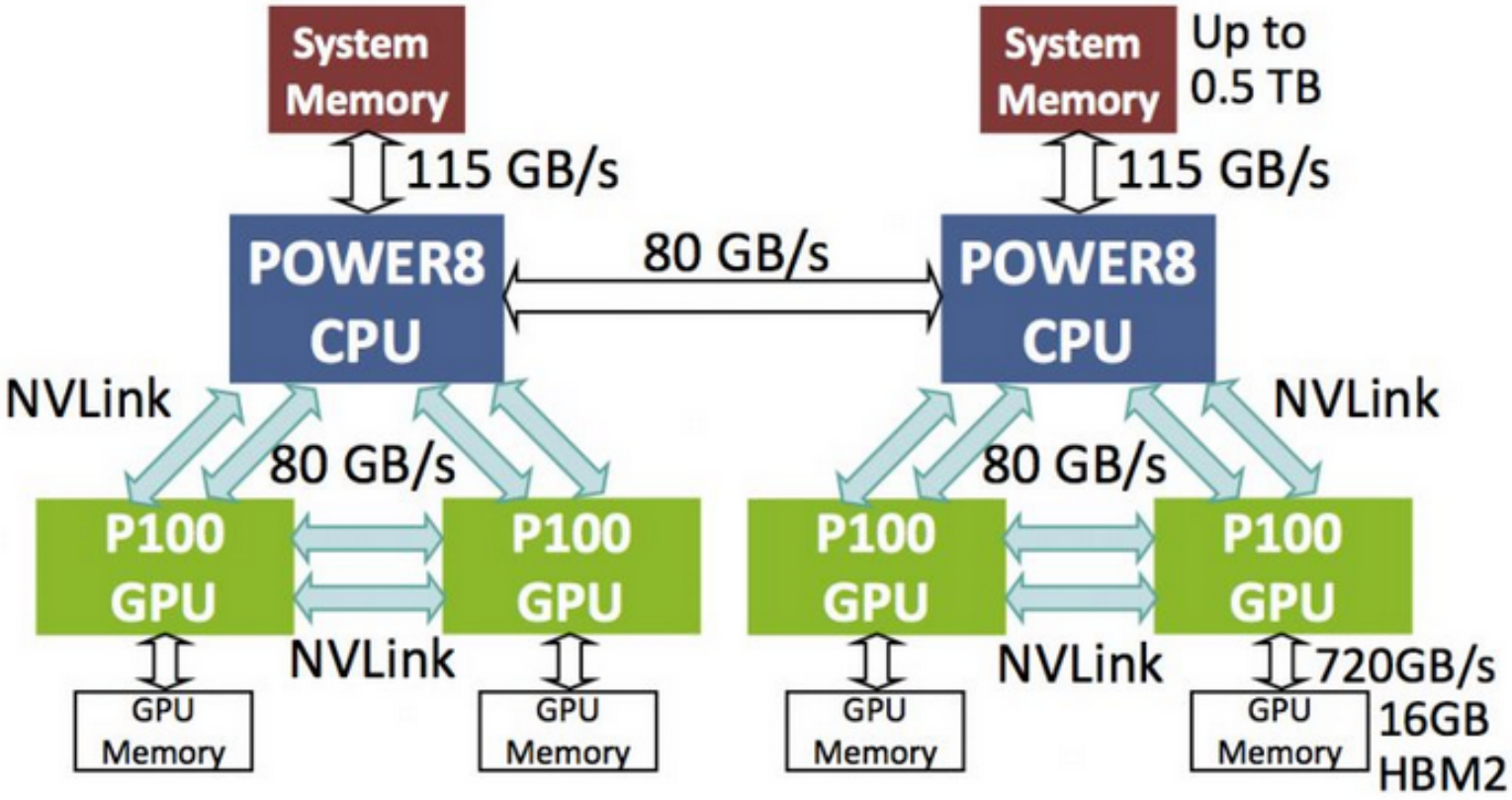
[NVIDIA, 2016]

IBM POWER8 Processor, NVIDIA P100 GPU and IBM Minsky Node Hardware Architecture

Part IV: Minsky Platform

13.11.2017 | Dirk Pleiter | Forschungszentrum Jülich/JSC & University of Regensburg

Minsky Platform: Details



Comparison POWER8 vs. P100 on JURON

	POWER8	P100
Number of cores/SMs	10	56
Double-precision Flop/cycle	80	3,584
Clock frequency [GHz]	3.49	1.33
Double-precision GFlop/s	279	4,760
Memory bandwidth read+write [GByte/s]	115.2	720
Balance [Flop/Byte]	2.4	6.6
Number of CPU or GPU	2	4
Aggregate double-precision TFlop/s	0.56	19
Aggregate memory bandwidth [GByte/s]	230	2880
Aggregate memory capacity [GByte]	256	64

IBM POWER8 Processor, NVIDIA P100 GPU and IBM Minsky Node Hardware Architecture

Part V: Use Case

13.11.2017 | Dirk Pleiter | Forschungszentrum Jülich/JSC & University of Regensburg

The Poisson Equation

- Poisson equation in 2 dimensions:

$$-\frac{\partial^2 v(x,y)}{\partial x^2} - \frac{\partial^2 v(x,y)}{\partial y^2} = f(x,y)$$

- Discretisation of 2nd-order derivative

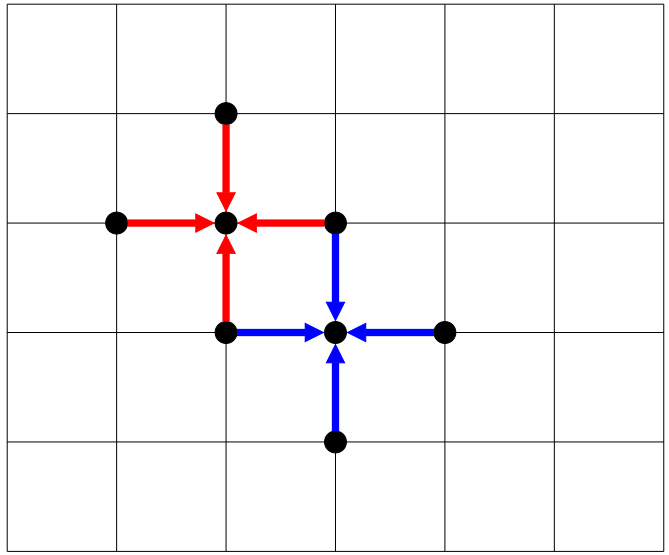
$$-\frac{\partial^2 v(x,y)}{\partial x^2} \leftarrow \frac{2v_{i,j} - v_{i-1,j} - v_{i+1,j}}{h^2}$$

- Discrete Poisson equaiton in 2 dimensions:

$$T\,v = h^2 f \quad \text{where} \quad T = \begin{pmatrix} 4 & -1 & 0 & \cdots \\ -1 & 4 & -1 & \cdots \\ \vdots & \vdots & \ddots & \end{pmatrix}$$

Discrete Poisson Equation: Data Locality

Graphical representation of $T\,v$:



Observations:

- Matrix T acts as a stencil operator
- Any element of vector v is reused 4 times

Jacobi Algorithm

- Algorithm suitable for diagonally dominant systems
- Matrix decomposition: $T = D + R$

$$D = \begin{pmatrix} t_{0,0} & 0 & 0 & \cdots \\ 0 & t_{1,1} & 0 & \cdots \\ \vdots & \vdots & \ddots & \end{pmatrix}, \quad R = \begin{pmatrix} 0 & t_{0,1} & t_{0,2} & \cdots \\ t_{1,0} & 0 & t_{1,2} & \cdots \\ \vdots & \vdots & \ddots & \end{pmatrix}$$

- Obtain solution through iterative precedure

$$v^{(k+1)} = D^{-1} \left(h^2 f - R v^{(k)} \right)$$

Jacobi Algorithm Implementation

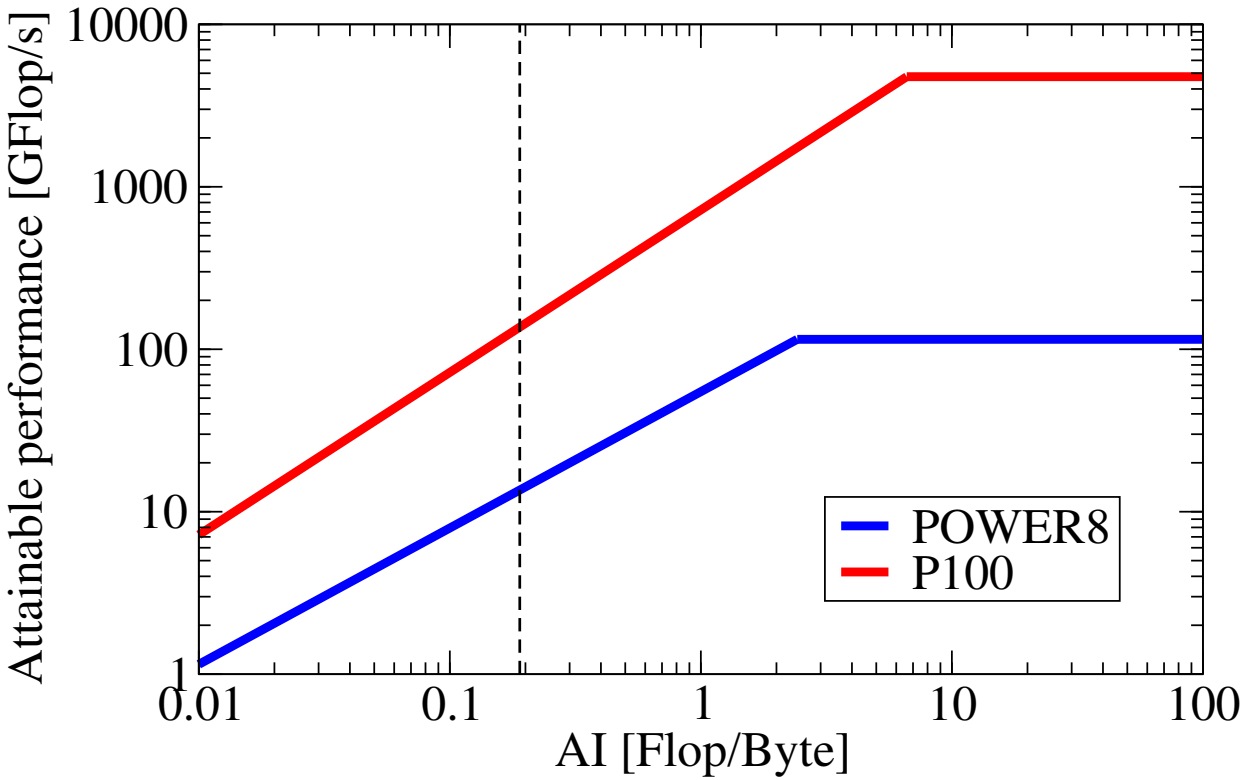
```
for( int iy = 1; iy < ny-1; iy++)
{
  for( int ix = 1; ix < nx-1; ix++ )
  {
    Anew[iy*nx+ix] = -0.25 * (rhs[iy*nx+ix]
      - (  Aref[iy    *nx + ix+1] + Aref[iy    *nx + ix-1] +
           Aref[(iy-1)*nx + ix  ] + Aref[(iy+1)*nx + ix  ]  ));
    error = fmaxr(error, fabsr(Anew[iy*nx+ix]-Aref[iy*nx+ix]));
  }
}
```

Information exchange analysis assuming small cache:

- Load of `Anew`, `rhs`, `Aref`: $I_{\mathrm{ld}} = n_x \cdot n_y \cdot 3 \cdot 8\,\mathrm{Byte}$
- Store `Anew`: $I_{\mathrm{st}} = n_x \cdot n_y \cdot 1 \cdot 8\,\mathrm{Byte}$
- Floating-point arithmetics: $I_{\mathrm{fp}} = n_x \cdot n_y \cdot 6\,\mathrm{Flop}$

Jacobi on POWER8/P100: Roofline Analysis

$$\mathrm{AI} = I_{\mathrm{fp}}/(I_{\mathrm{ld}} + I_{\mathrm{st}}) = 0.19\,\mathrm{Flop/Byte}$$



IBM POWER8 Processor, NVIDIA P100 GPU and IBM Minsky Node Hardware Architecture

Part VI: Summary

13.11.2017 | Dirk Pleiter | Forschungszentrum Jülich/JSC & University of Regensburg

Summary

- Approach for systematic performance analysis and modelling based on Information Exchange introduced
- Minsky as a first generation of node architectures for supercomputers based on
  - POWER8 processors
  - P100 GPUs
- Introduced use case for today: Solving 2-dimensional Poisson equation using Jacobi solver
  - Memory bandwidth limited application
  - Roofline model defines upper performance limit