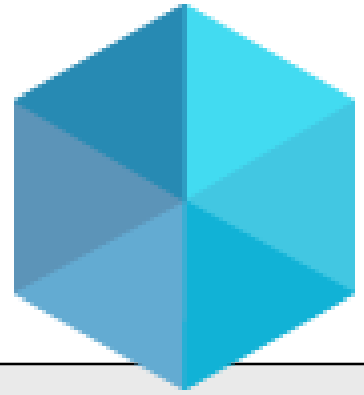
OpenPOWER™

Emerging Workload Performance Evaluation on Future Generation OpenPOWER Processors

Saritha Vinod
Power Systems Performance Analyst
IBM Systems
sarithavn@in.bm.com



Agenda

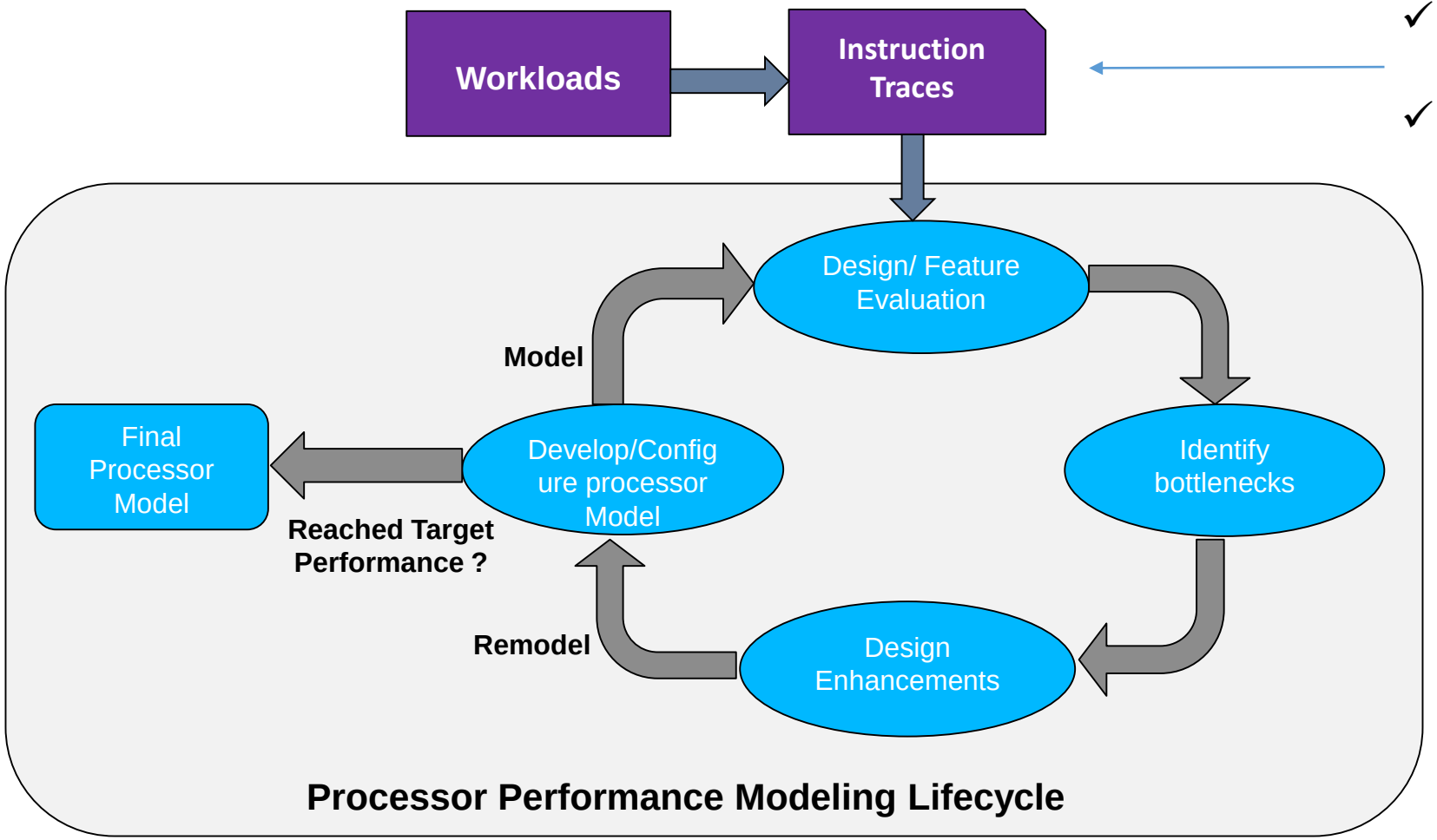
- Emerging Workloads Characteristics and Performance
- Performance Modelling Lifecycle for Future Generation Processors
- Workload Tracing Process
- Workload Tracing Methods & Tools
- Key Challenges in Workload Tracing
- Performance Evaluations using Traces
  - Microarchitecture Design Analysis
  - Software Performance Optimizations
  - Performance Verification
- Summary

Emerging Workloads Characteristics and Performance

- New industry trends leading to emerging workloads in domains such as Cognitive computing, Deep Learning, Analytics, Cloud etc.

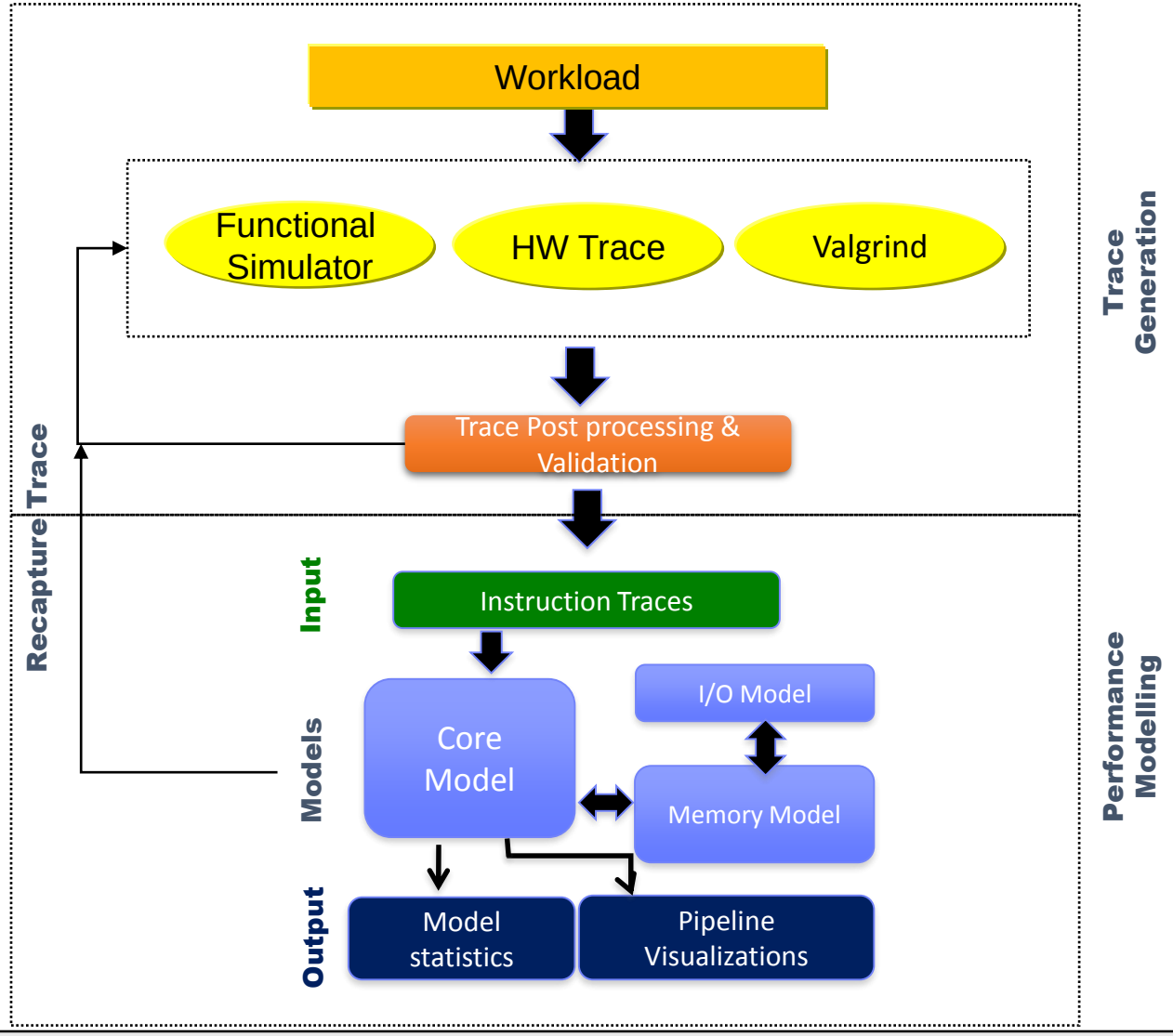
- To achieve best performance it is important for the next generation processor design to address some of the following emerging workload characteristics
  - Instruction mixes & compute needs
  - Cache access patterns & prefetch
  - Data access patterns
  - Sharing of data
  - Data affinities
  - Branch prediction
  - OS and Hypervisor calls

Performance Modelling Lifecycle for Future Generation Processors



- ✓ Traces provide key workload characteristics
- ✓ Enable performance evaluation of future generation processors

Workload Tracing Process



Workload Tracing Methods & Tools

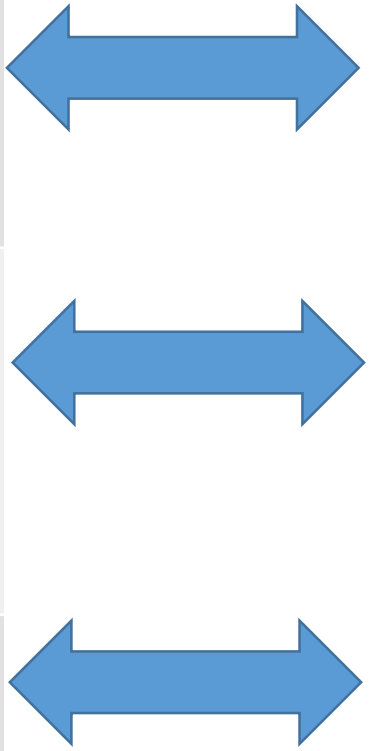
Functional Simulator		Hardware Traces		Valgrind Framework	
• Highly Controlled simulation environment • Supports sampling of multi-phase workloads • System level tracing	• Not well-suited for workloads with complex stack, large memory and highly threaded workloads	• Used for commercial workloads with high core counts and memory requirements • Instruction and bus traces • System level tracing	• Complex setup process • Lacks support for generating sampled traces	• Useful for tracing hot functions or problem areas in the application • Supports sampling	• Provides only application tracing, no system level

Reference : IBM POWER8 Functional Simulator (systemsim)
http://www-304.ibm.com/webapp/set2/sas/f/pwrfs/pwrfsinstall.html

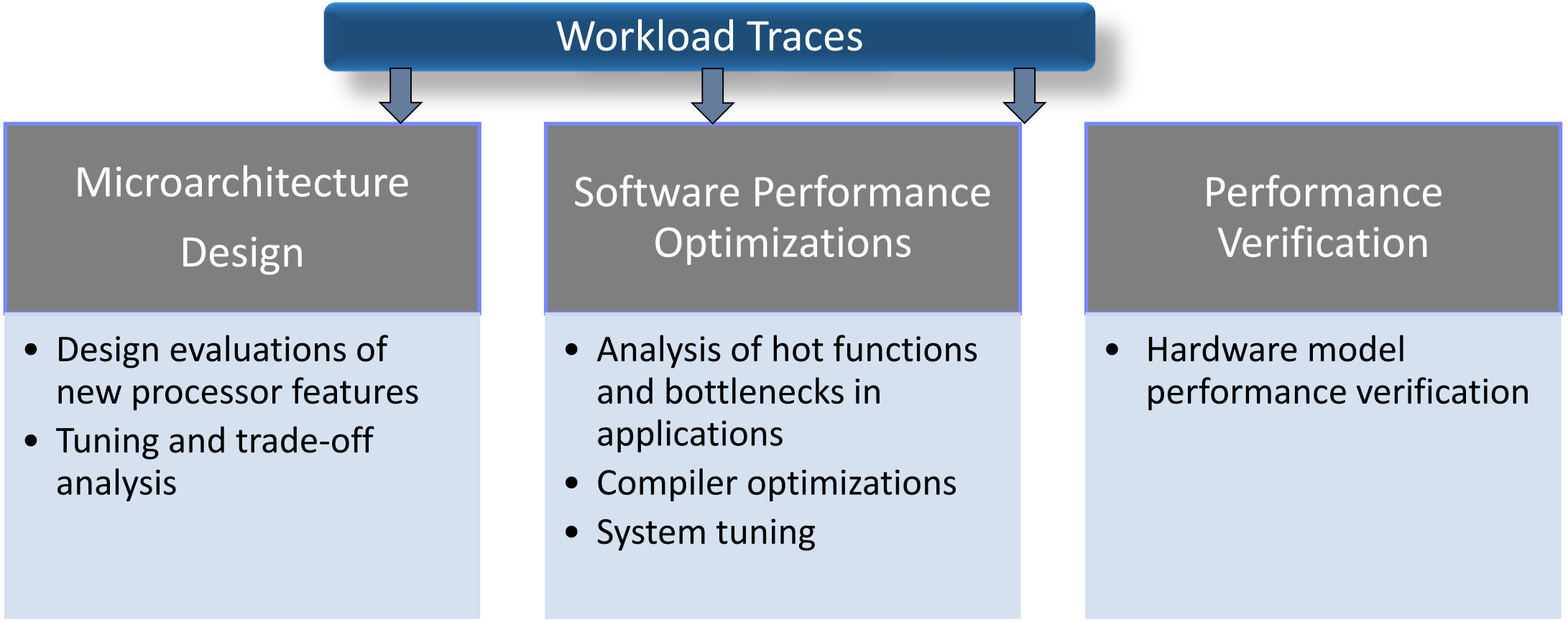
Reference : IBM SDK for Linux on Power
https://www-304.ibm.com/webapp/set2/sas/f/lopdiags/sdklop.html

Key Challenges in Workload Tracing

Challenges	Resolutions
• Hardware models execute only a subset of instructions; most workloads run into billions of instructions. • Overall runtime of emerging workloads increasing	• Identify workload interval to trace – workload steady state, phases based on performance counter data
• A smaller subset of runtime with representative workload behavior required for design studies. • Selection depends on the design needs and the workload characteristics	• Representative trace segment selection – sampled, contiguous, filtered or at unit level
• The selected segment need to retain the original workload characteristics	• Trace profile validation – capturing the right application runtime, maintaining the CPI characteristics



Performance Evaluations using Traces



Microarchitecture Design Analysis and Optimization

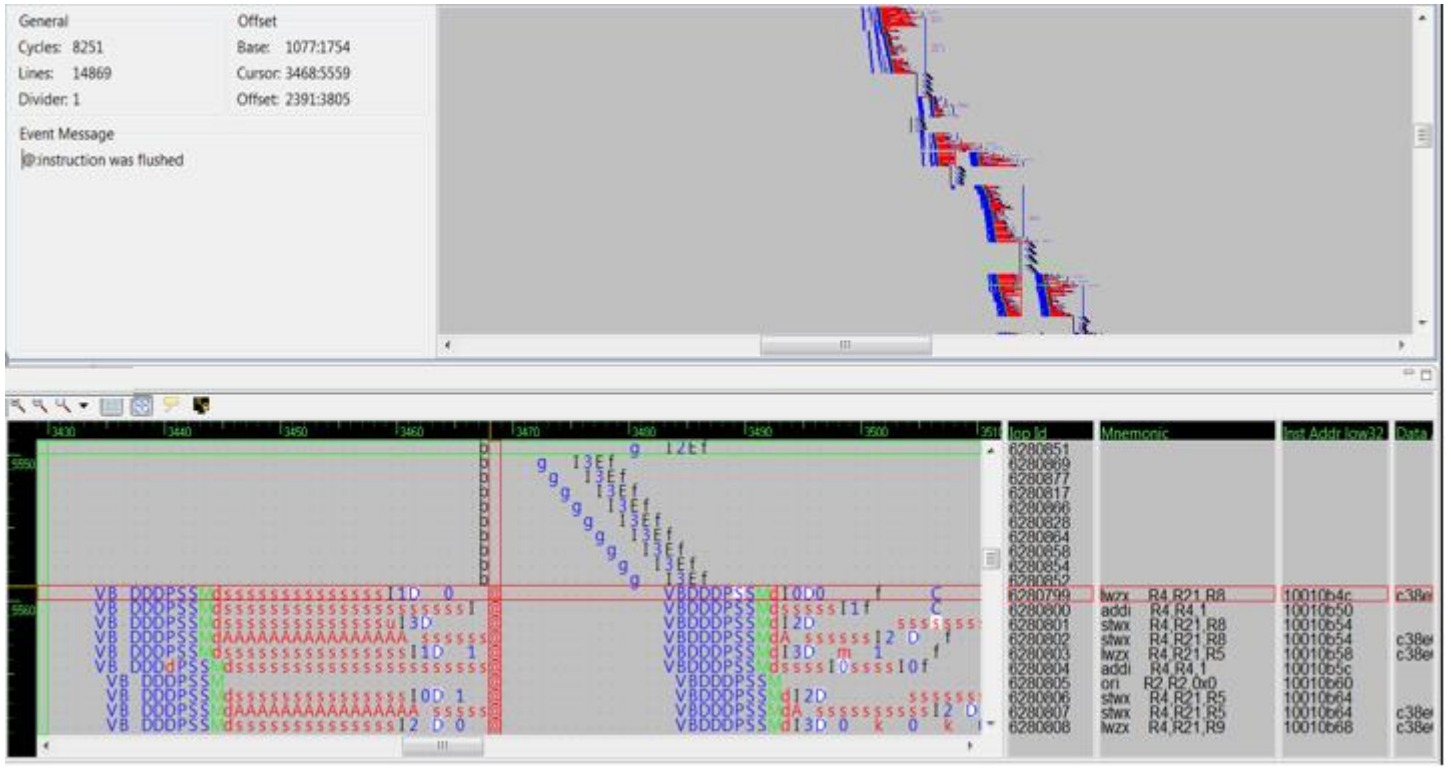
- Tuning and trade-off analysis
  - Determine capacity – Cache size , queue size
  - Sensitivity analysis using various categories of workload traces

- New Design evaluations
  - New techniques for load-store handling
  - Branch prediction algorithms
  - Data prefetch design

Software Performance Optimizations

- Analyzing application performance bottlenecks
  - Back to back latency issues, LSU stalls, Branch mispredictions etc.
- Compiler optimizations
  - Microarchitecture dependent
    - Scheduling, ISA exploitation
  - Microarchitecture independent
    - Inlining, unrolling etc.
  - Flag tuning
- System tuning
  - SMT levels
  - Prefetch settings
  - Large pages

Micro-architecture Pipeline View for Optimizations



Cycle accurate simulator

- Micro-architecture statistics
- Pipeline view for the instruction mix

References:
IBM Power 8 Performance Simulator
https://www-304.ibm.com/webapp/set2/sas/f/lopdiags/sdkdownload.html

Performance Verification

- Workload traces used for performance verification of hardware model
- Broader performance comparison of final hardware model and the performance model
- To identify delta gaps in performance

Summary

- OpenPOWER processors designed to deliver superior performance
- Performance evaluation and micro-architecture analysis tools and methods available for open innovation
- Key insights derived from emerging workloads through traces
  - Enables micro-architecture design evaluations, trade-off analysis, software/compiler optimizations and verification

Thank you