

DARPA AACE Blackjack



Presented by

Anthony Danalis and Jack Dongarra

Oak Ridge National Laboratory

University of Tennessee–Knoxville Innovative Computing Laboratory

Piotr Luszczek

University of Tennessee–Knoxville Innovative Computing Laboratory

Gabriel Marin and Jeff Vetter

Oak Ridge National Laboratory

Eric Stoltz and Michael Wolfe

The Portland Group, Inc.

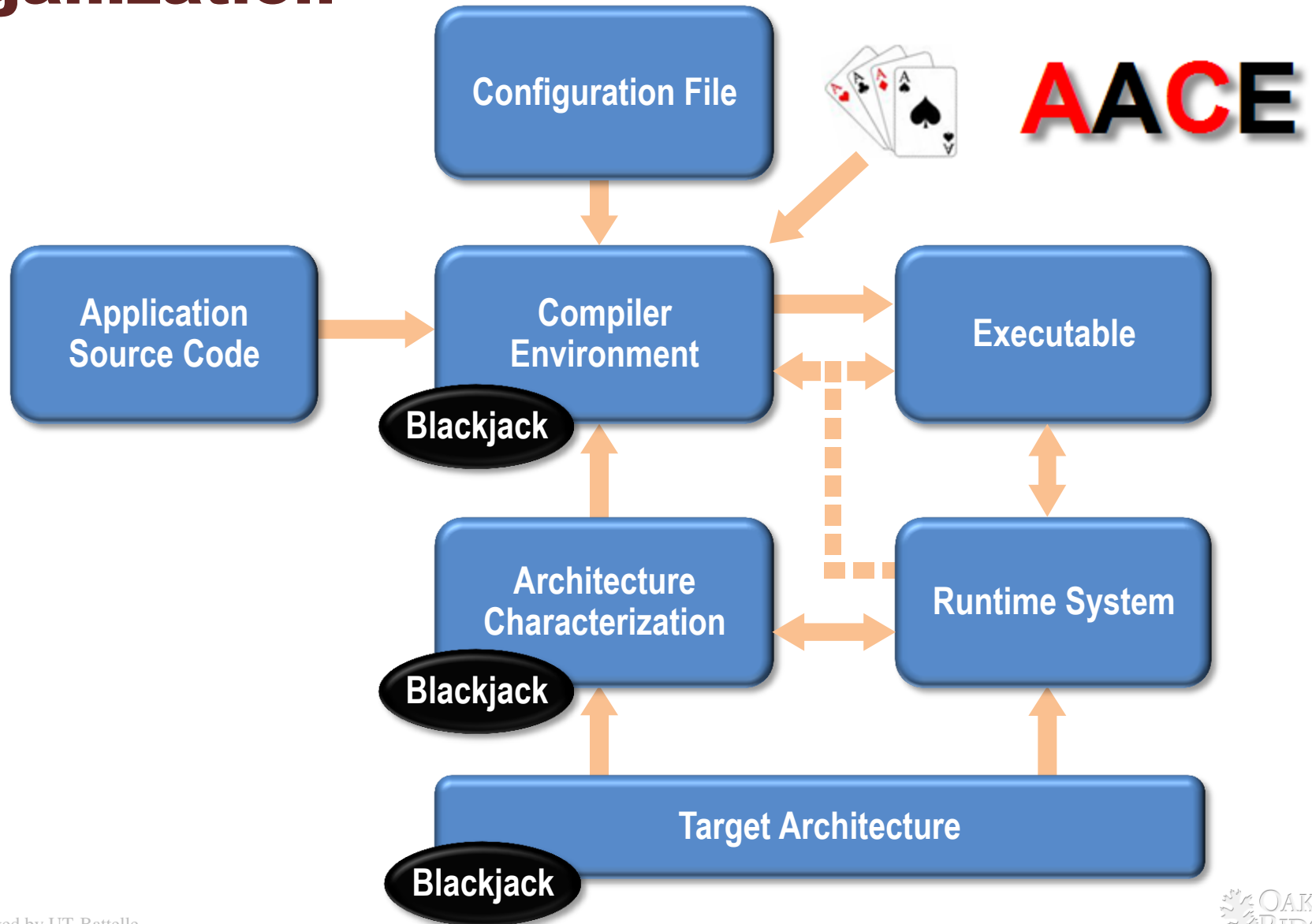


High-level Blackjack project outline

Blackjack develops systematic ways to

- **Characterize the architecture of various computing systems**
- **Benchmark complex compiler systems**
- **Evaluate the effect that compiler systems have on productivity and performance**

Blackjack in typical AACE system organization



Blackjack is organized across six technical thrusts

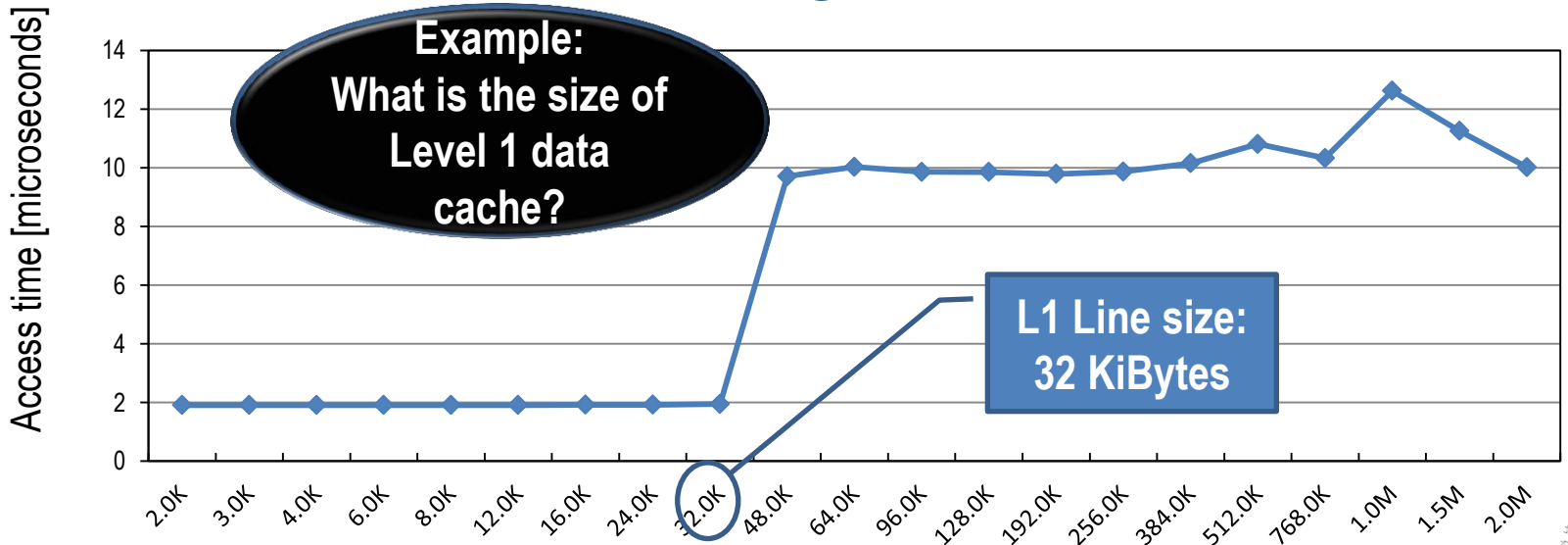
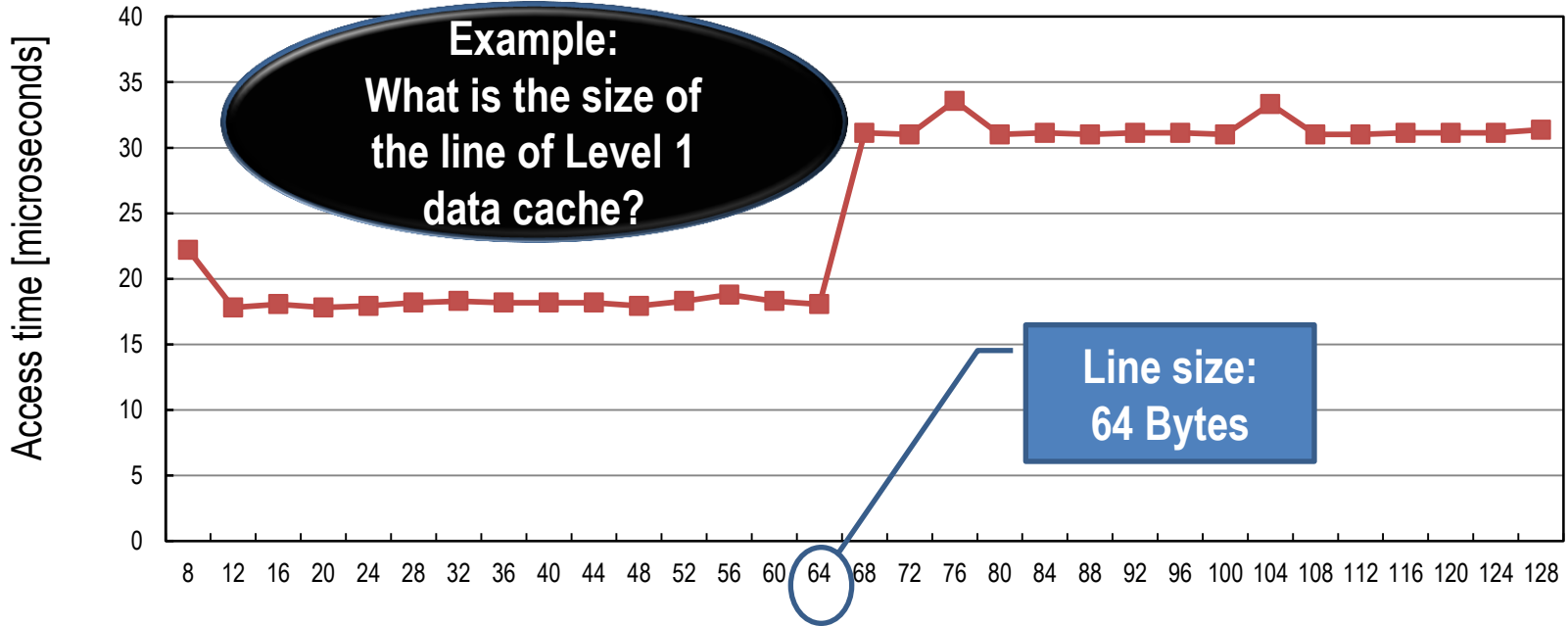
Metrics	<ul style="list-style-type: none">• Correctness, performance, productivity• In collaboration with DARPA and compiler teams
Metadata for Configuration and Characterization	<ul style="list-style-type: none">• Work with compiler teams to define and refine configuration and characterization metadata, process
Benchmark and Application Evaluation Suite	<ul style="list-style-type: none">• Select and provide benchmark and application suites• Adapt select applications/benchmarks to compiler team features
Evaluation Harness and Database	<ul style="list-style-type: none">• Develop and deploy evaluation harness and database for testing AACE compiler team products
Evaluation System Selection and Characterization	<ul style="list-style-type: none">• Work with DARPA to select AACE target systems• Develop configuration files for systems• Manually characterize systems
Prototype Evaluation	<ul style="list-style-type: none">• Independently evaluate compiler products• Program phases I, II

Blackjack system characterization

Main Challenges

- **Discover hardware system characteristics**
 - With micro-benchmarks and statistical data analysis
 - With information provided by the vendor (configuration)
- **Define/format/store the information**
 - XML schema
- **Evaluate automatically generated characterizations**
 - Grade individual characteristics based on type
 - Weigh and average grades based on importance to compiler

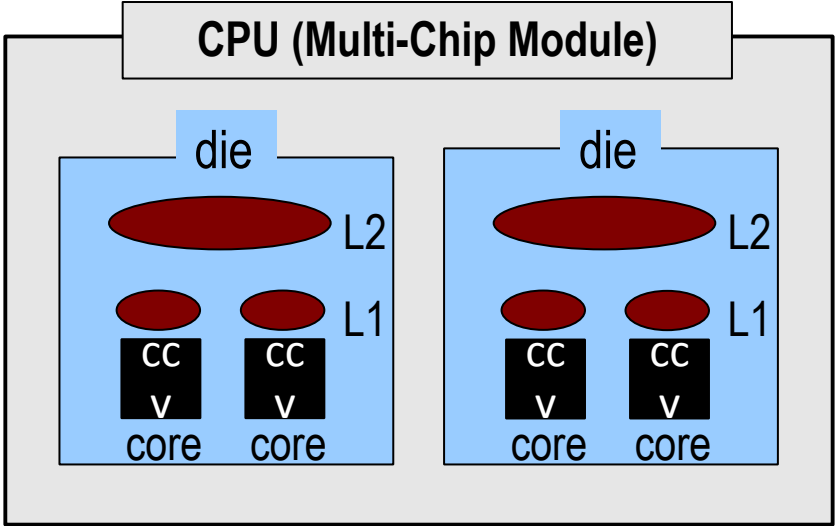
Discovering system characteristics



Sample of characterization schema

Intel Yorkfield™ Q9450 (core 2 quad)

2 cores per die
2 dies per CPU
N CPUs per node



```
<node>
  <hierarchy>N-2-2</hierarchy>
  <pu>
    <type>Intel Q9450</type>
    <frequency>2660</frequency>
    <SIMD>SSE4.1</SIMD>
```

```
  ...
  <cache>
    <id>1</id>
    <level>1</level>
  ...
</cache>
```

```
<cache>
  <id>2</id>
  <level>2</level>
  <count>2</count>
  ...
  <shared>true</shared>
  <sharedatlevel>1</sharedatlevel>
  <upstream>
    <upstreamID>3</upstreamID>
    <minlatency>12345</minlatency>
    <bandwidth>54321</bandwidth>
  </upstream>
</cache>
</pu>
```

2 L2 caches/CPU


2 L2 caches/CPU

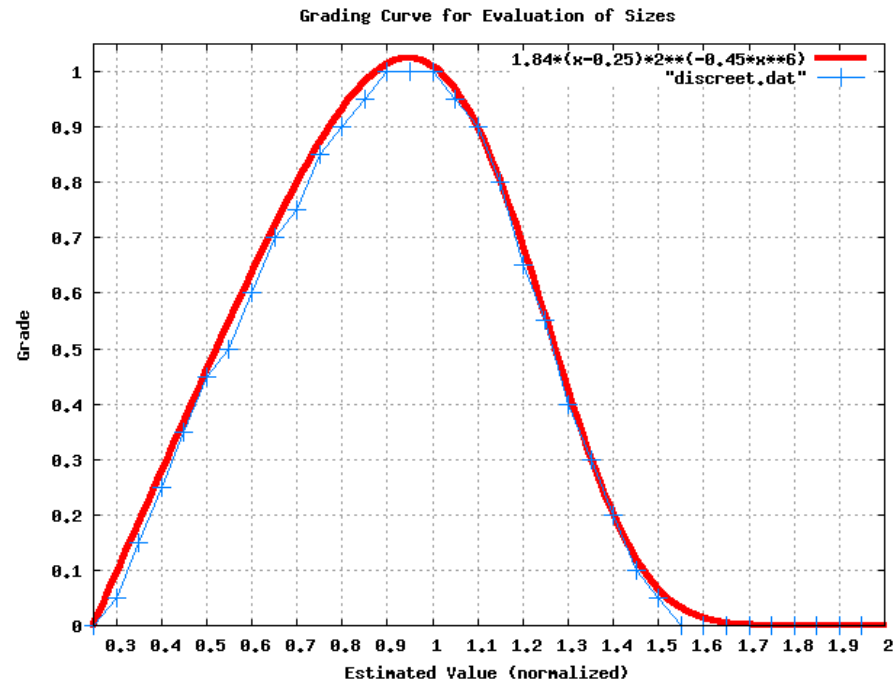
2 L2 caches/CPU



Evaluating a characterization

Grading approaches

- **Binary (pass/fail)**
 - Critical characteristics
- **Normal distribution**
 - Latencies/bandwidths
- **Skewed curve** 
 - Values that are better to underestimate than overestimate



Contacts

Piotr Luszczyk

The University of Tennessee, Knoxville
Innovative Computing Laboratory
(865) 974-3126
luszczyk@eecs.utk.edu

Anthony Danalis

The University of Tennessee, Knoxville
Innovative Computing Laboratory
(865) 974-3052
adanalis@eecs.utk.edu