Reducing Measurement Overhead by Leveraging Static and Profile Information for Automated and Application-specific Instrumentation

Instrumentation with Call Path Differentiation

TECHNISCHE UNIVERSITÄT DARMSTADT

Christian Iwainsky

Institute for Scientific Computing

TU Darmstadt

Context:
Performance Analysis Instrumentation

TECHNISCHE UNIVERSITÄT DARMSTADT

Performance Analysis Requires:

- **Data**
- **Context of the Data**
  - Where in the execution of the program the data is from
  - How the program got to that point
- **Low Overhead Measurement**
  - Automated Instrumentation

Example Scenario:

- **MPI Context Instrumentation**
  - Instrument Functions on the Path to MPI Functions

Context:
Performance Analysis Instrumentation

TECHNISCHE UNIVERSITÄT DARMSTADT

Performance Analysis Requires:

- **Data**
- **Context of the Data**
  - Where in the execution of the program the data is from
  - How the program got to that point
- **Low Overhead Measurement**
  - Automated Instrumentation

Example Scenario:

- **MPI Context Instrumentation**
  - Instrument Functions on the Path to MPI Functions

Context:
Performance Analysis Instrumentation

TECHNISCHE UNIVERSITÄT DARMSTADT

Performance Analysis Requires:

- **Data**
- **Context of the Data** → Full Context
  - Where in the execution of the program the data is from
  - How the program got to that point
- **Low Overhead Measurement**
  - Automated Instrumentation

Example Scenario:

- **MPI Context Instrumentation**
  - Instrument Functions on the Path to MPI Functions

Instrumentation
- An overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Setup:

Modify the source/program to trigger measurement for constructs

Runtime Measurement:

For all measurement locations:

1) Update and store current context
2) Gather additional information

Instrumentation
- An overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Setup:

Modify the source/program to trigger measurement for constructs

Runtime Measurement:

For all measurement locations:

1) Update and store current context
2) Gather additional information

Where:	What Function?
From Where:	What is the Call Chain?

Instrumentation
- An overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Setup:

Modify the source/program to trigger measurement for constructs

Runtime Measurement:

For all measurement locations:

1) Update and store current context
2) Gather additional information

Where: What Function?
From Where: What is the Call Chain?

- All functions must be instrumented
- Context can be constructed offline

Instrumentation
- An overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Setup:

Modify the source/program to trigger measurement for constructs

Runtime Measurement:

For all measurement locations:

1) Update and store current context
2) Gather additional information

Where: What Function?
From Where: What is the Call Chain?

Overhead

$$O_{Instrumentation} = O_{Probe} * n_{invocations}$$

- All functions must be instrumented
- Context can be constructed offline

Instrumentation - An overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Setup:

Modify the source/program to trigger measurement for constructs

Runtime Measurement:

For all measurement locations:

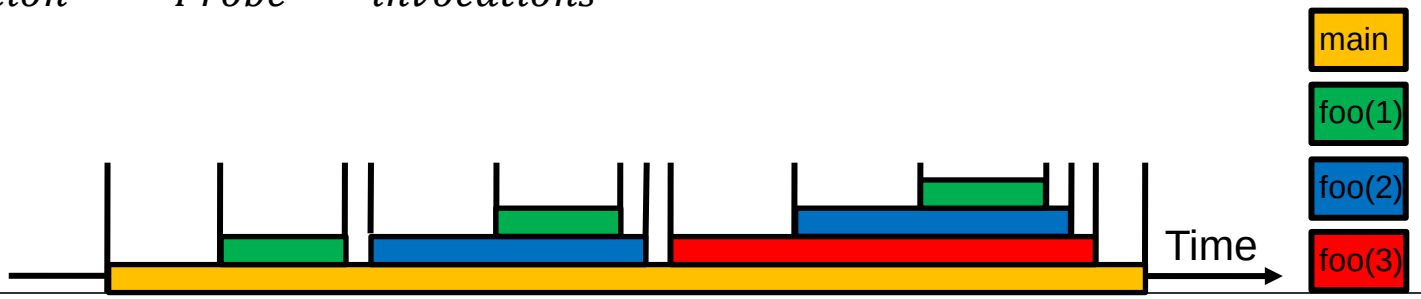
1) Update and store current context
2) Gather additional information

Where: What Function?
From Where: What is the Call Chain?

- All functions must be instrumented
- Context can be constructed offline

Overhead

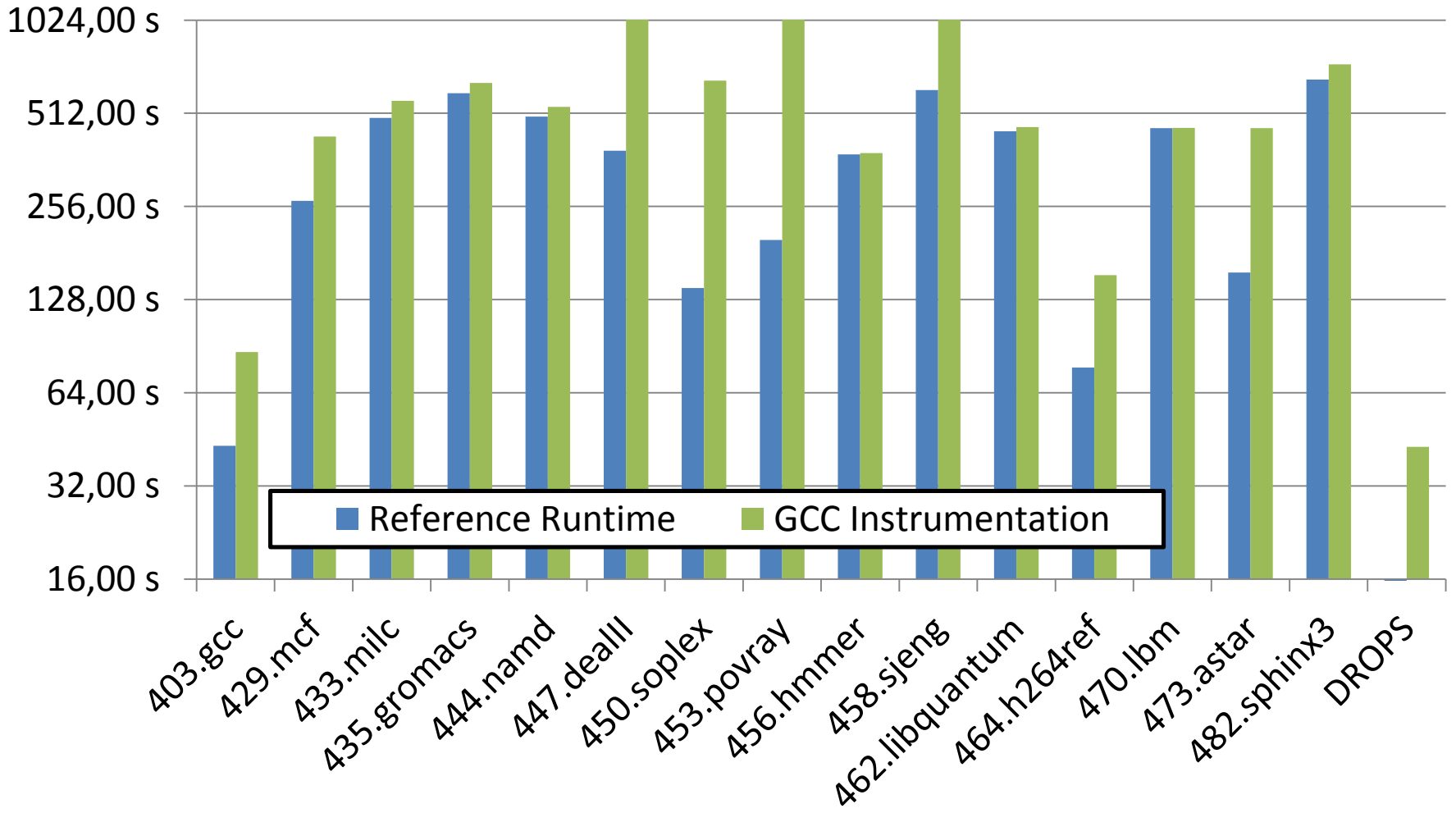
$$O_{Instrumentation} = O_{Probe} * n_{invocations}$$



SCIENTIFIC COMPUTING

SPEC CPU Measurements Part 1/3: Reference Time & Full Instrumentation

TECHNISCHE UNIVERSITÄT DARMSTADT

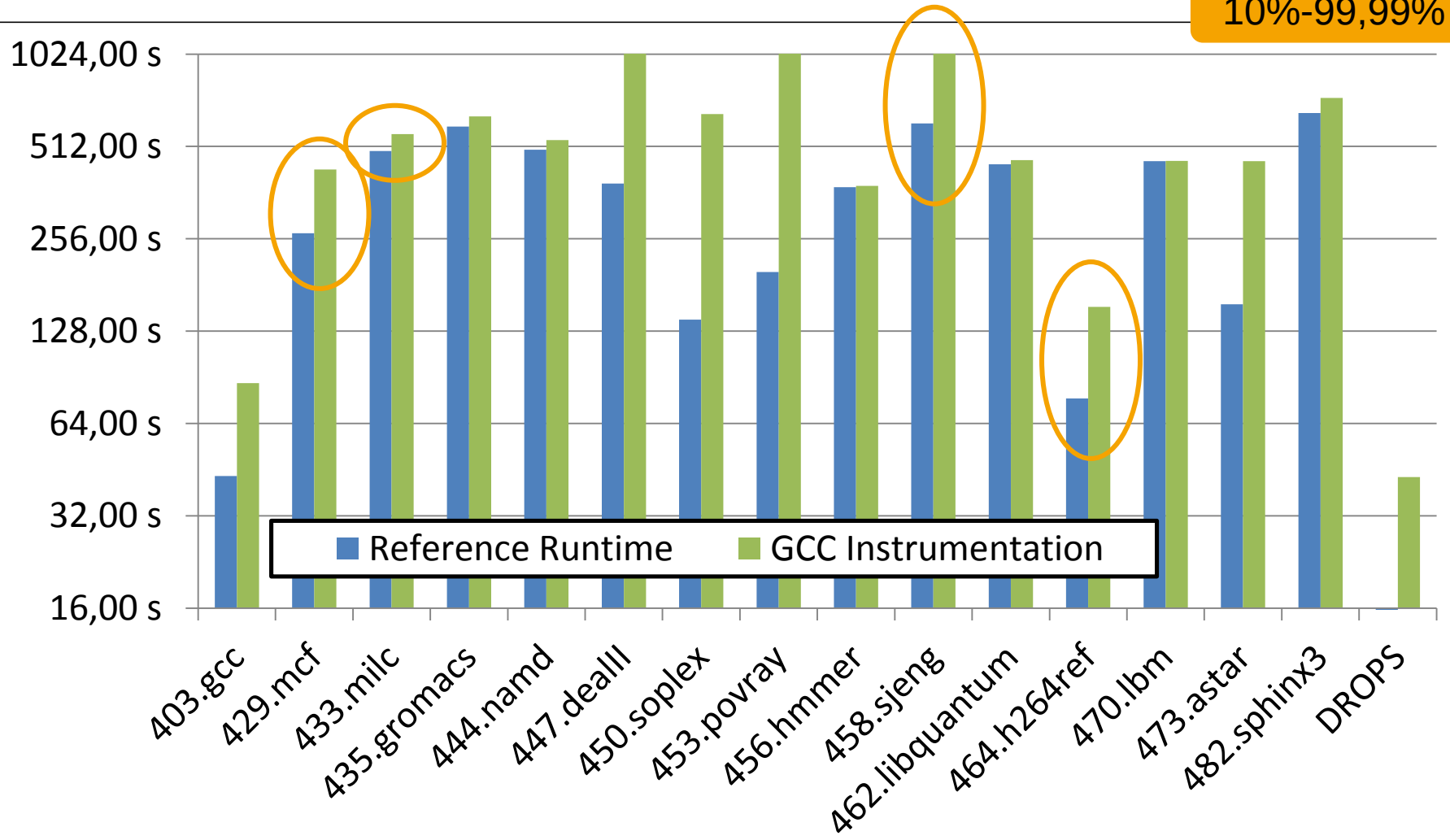


SCIENTIFIC COMPUTING

SPEC CPU Measurements Part 1/3: Reference Time & Full Instrumentation

TECHNISCHE UNIVERSITÄT DARMSTADT

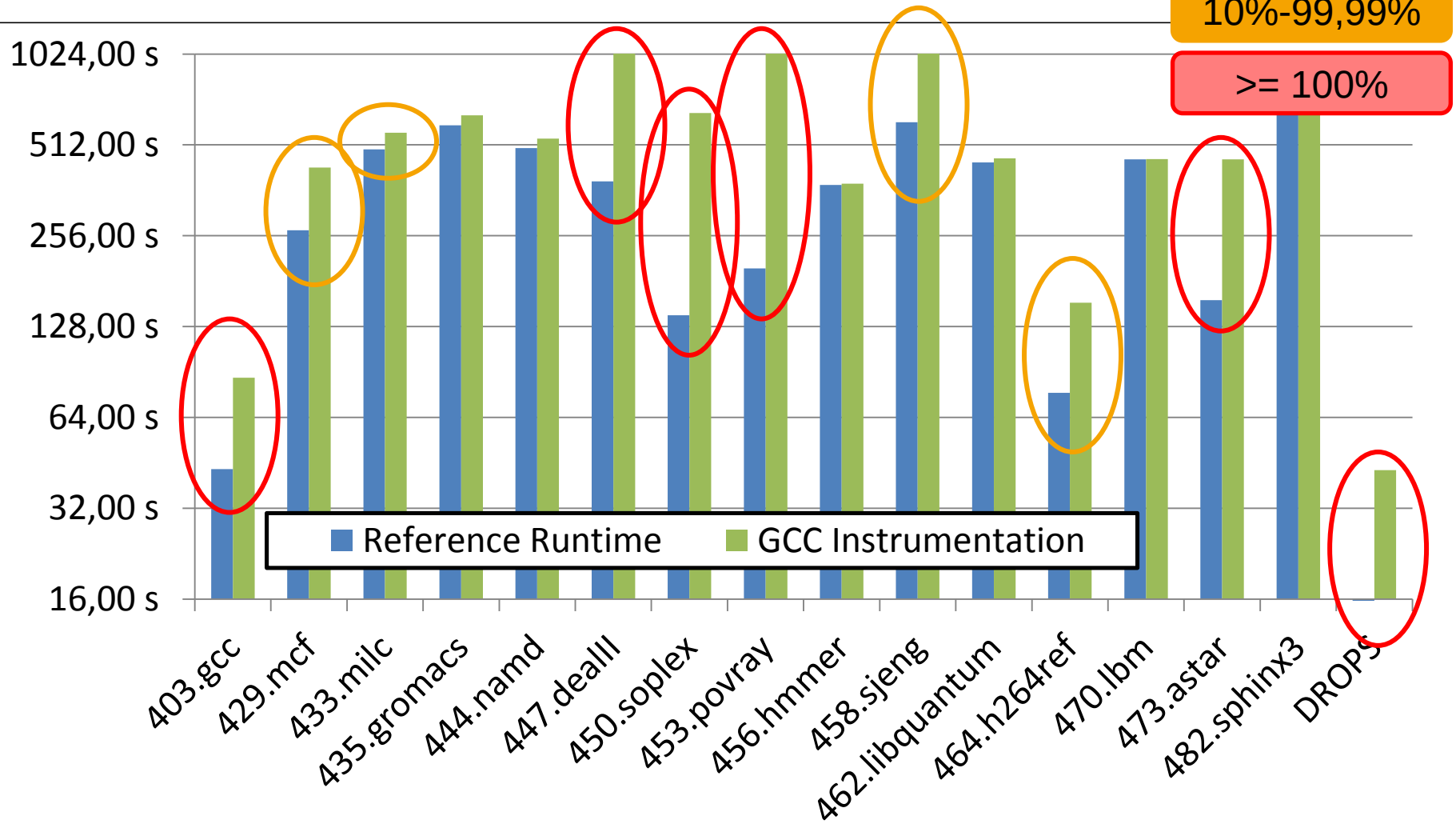
10%-99,99%



SCIENTIFIC COMPUTING

SPEC CPU Measurements Part 1/3: Reference Time & Full Instrumentation

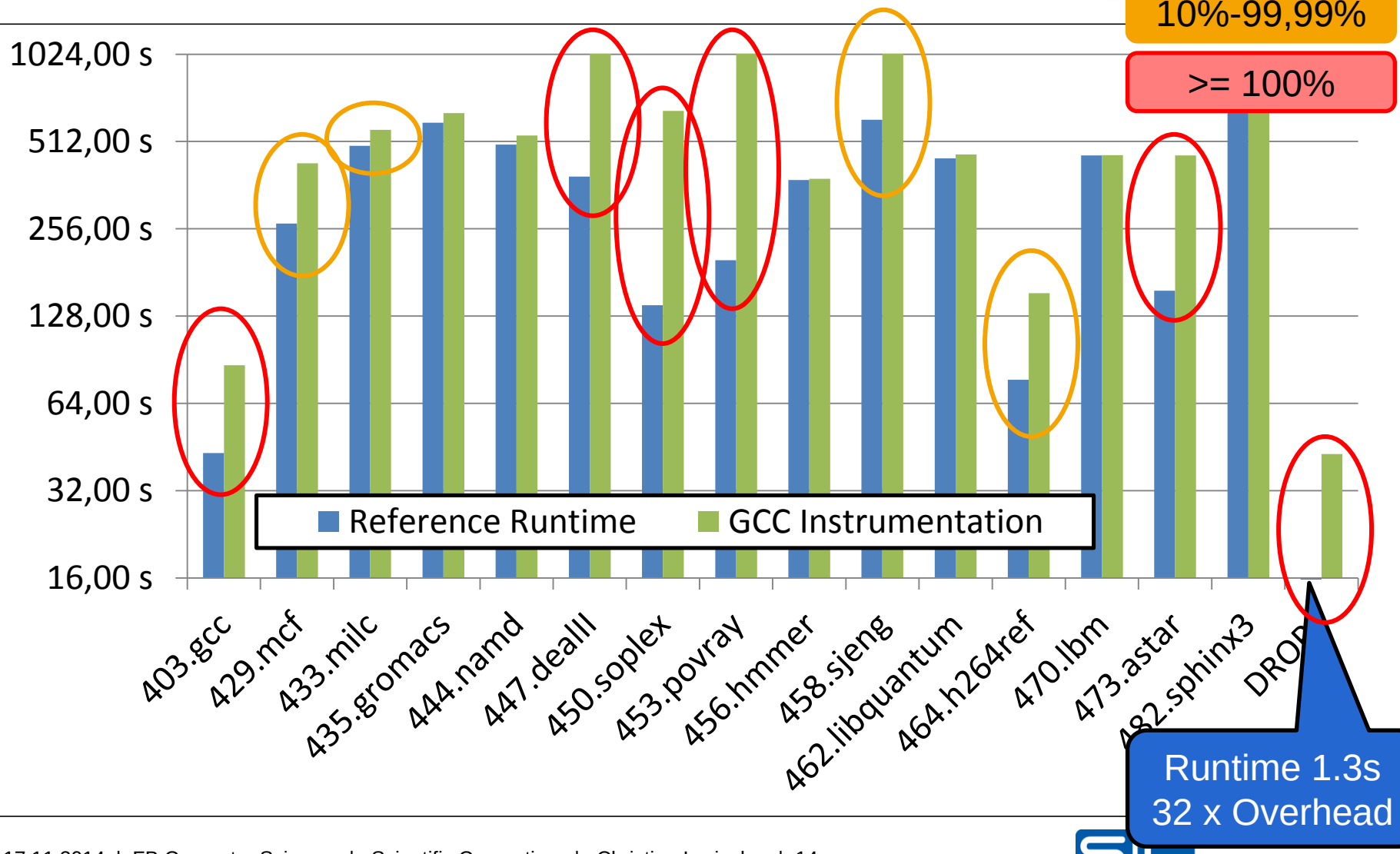
TECHNISCHE UNIVERSITÄT DARMSTADT



SCIENTIFIC COMPUTING

SPEC CPU Measurements Part 1/3: Reference Time & Full Instrumentation

TECHNISCHE UNIVERSITÄT DARMSTADT



SC COMPUTING

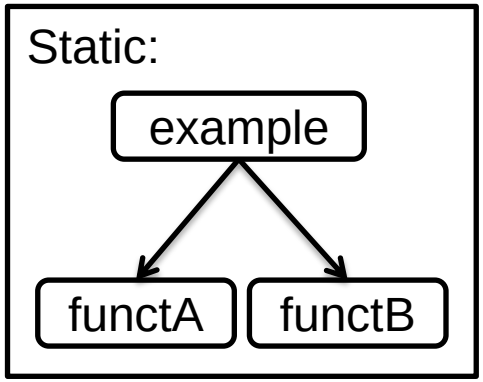
Call Graph

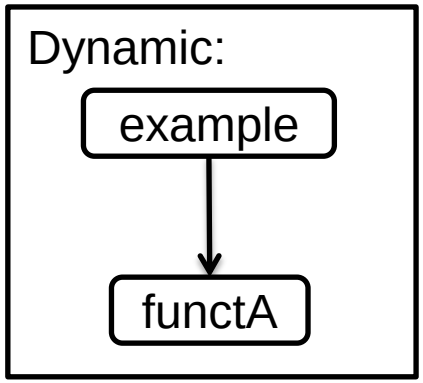
TECHNISCHE UNIVERSITÄT DARMSTADT

- **Describes Relationship of Functions of a Program**
- **Static (Potential) Call Graph:**
  Derived from the source / program
- **Dynamic (Actual) Call Graph:**
  Observed at Runtime. Generated e.g. from a ScoreP Profile

The Call Graph describes how a given function is reached.

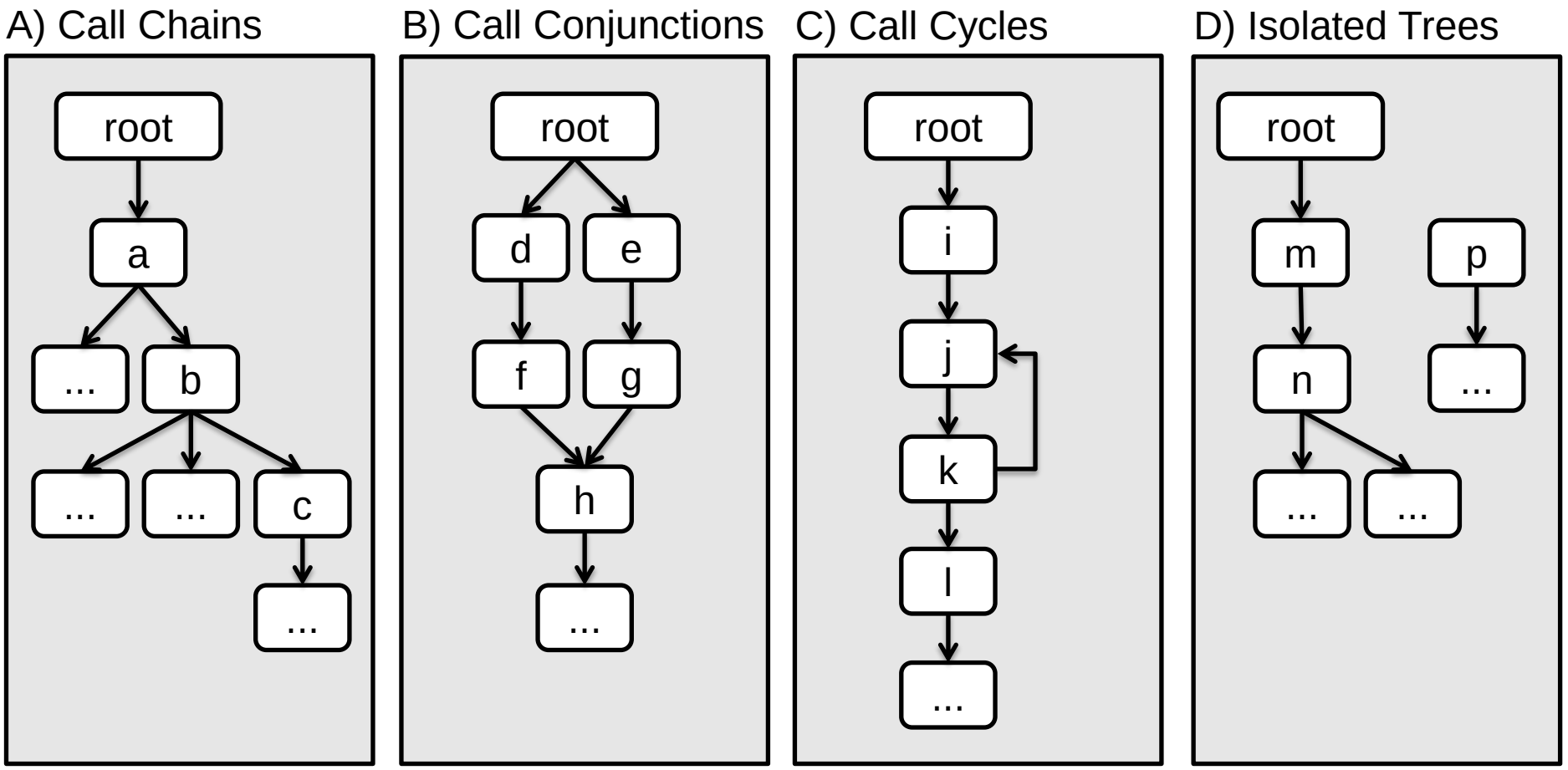
```
void example(int arg){
  if (arg >= 0)
    functA();
  else
    functB();
}
```



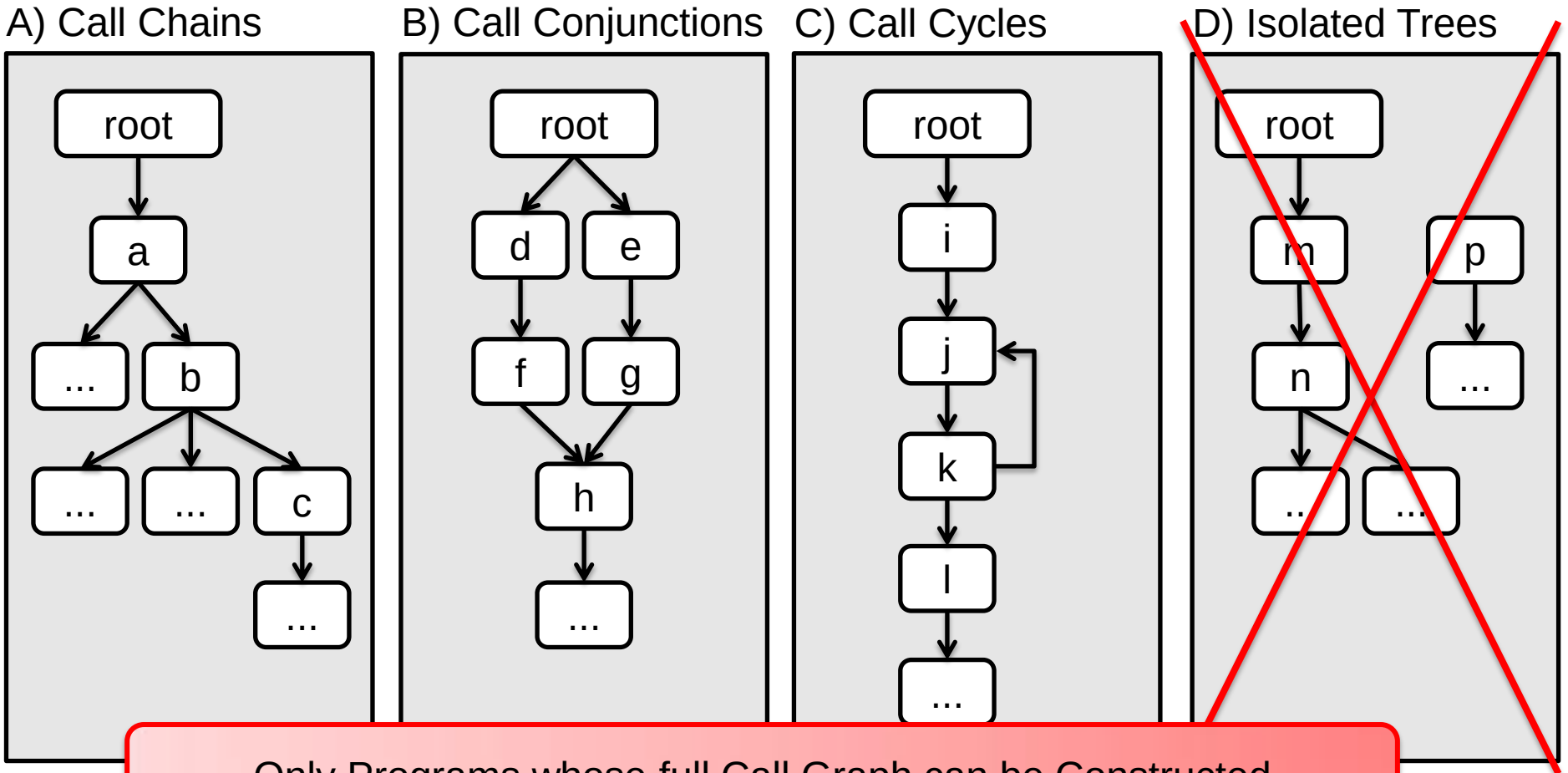


SCIENTIFIC COMPUTING

Four Types of Patterns



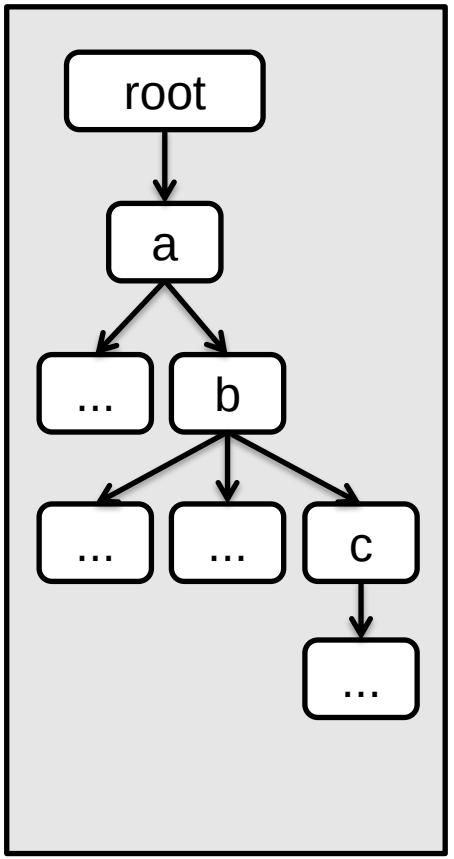
Four Types of Patterns



Only Programs whose full Call Graph can be Constructed

Call Chains

- **Sequences of Functions with a Single Parent**
- **Last node with this Property is either**
  - main function **or**
  - called “head“ of the chain



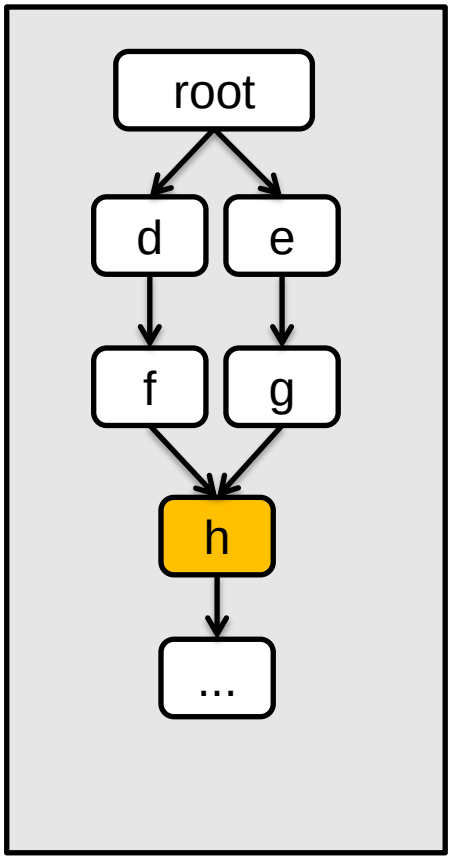
Call Conjunctions

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Node with Multiple Parents**
- **Head of one or multiple Call Chains**

Solution:

- **Instrument Parents (and Store an Identifier for the Path taken)**
- **Cycle-Free Call Graphs require only a fixed length array of Identifiers to reconstruct context**

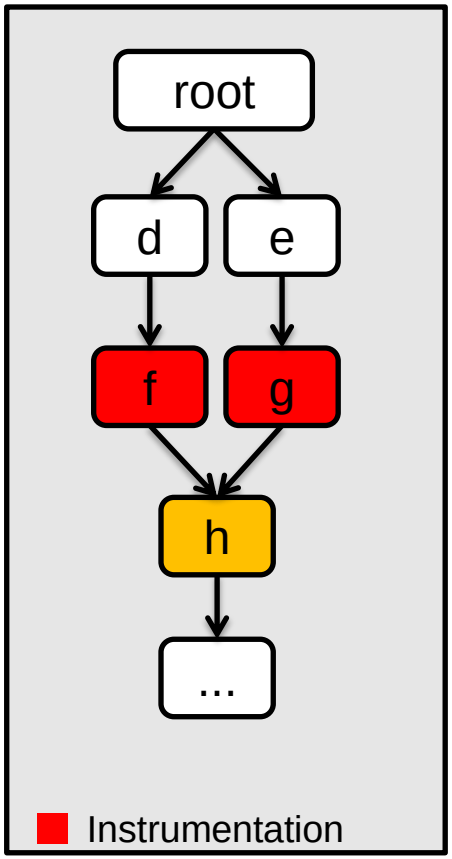


Call Conjunctions

- **Node with Multiple Parents**
- **Head of one or multiple Call Chains**

Solution:

- **Instrument Parents**
  (and Store an Identifier for the Path taken)
- **Cycle-Free Call Graphs require only a fixed length array of Identifiers to reconstruct context**



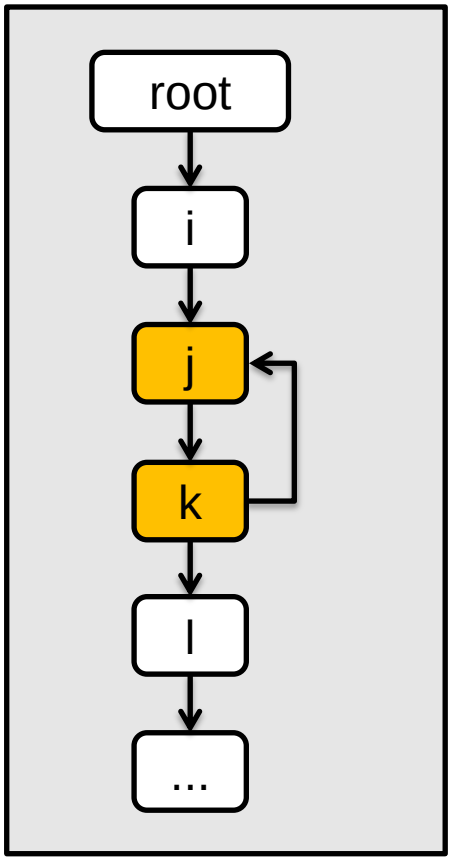
Call Cycles

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Node, where one of the Parents is a Child of the Conjunction Node**

Solution:

- **Instrument Parents and Store an Identifier for the Path taken**
- **Cycles Require a Stack of Identifiers to Reconstruct Context**



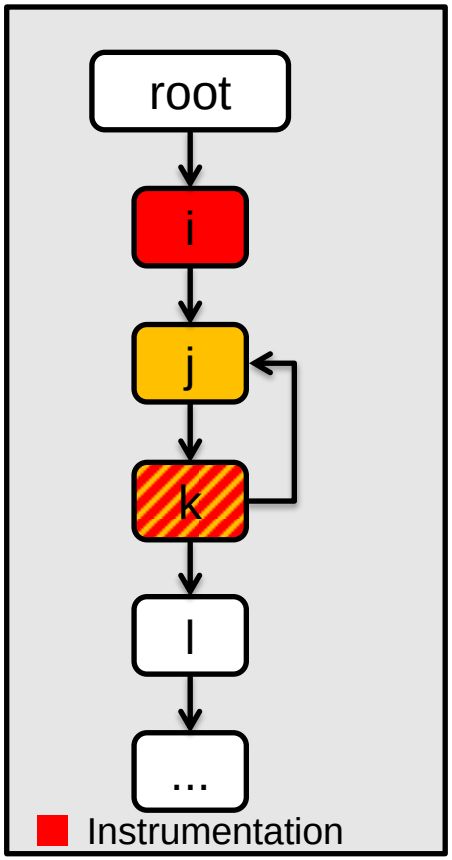
SCIENTIFIC COMPUTING

Call Cycles

- **Node, where one of the Parents is a Child of the Conjunction Node**

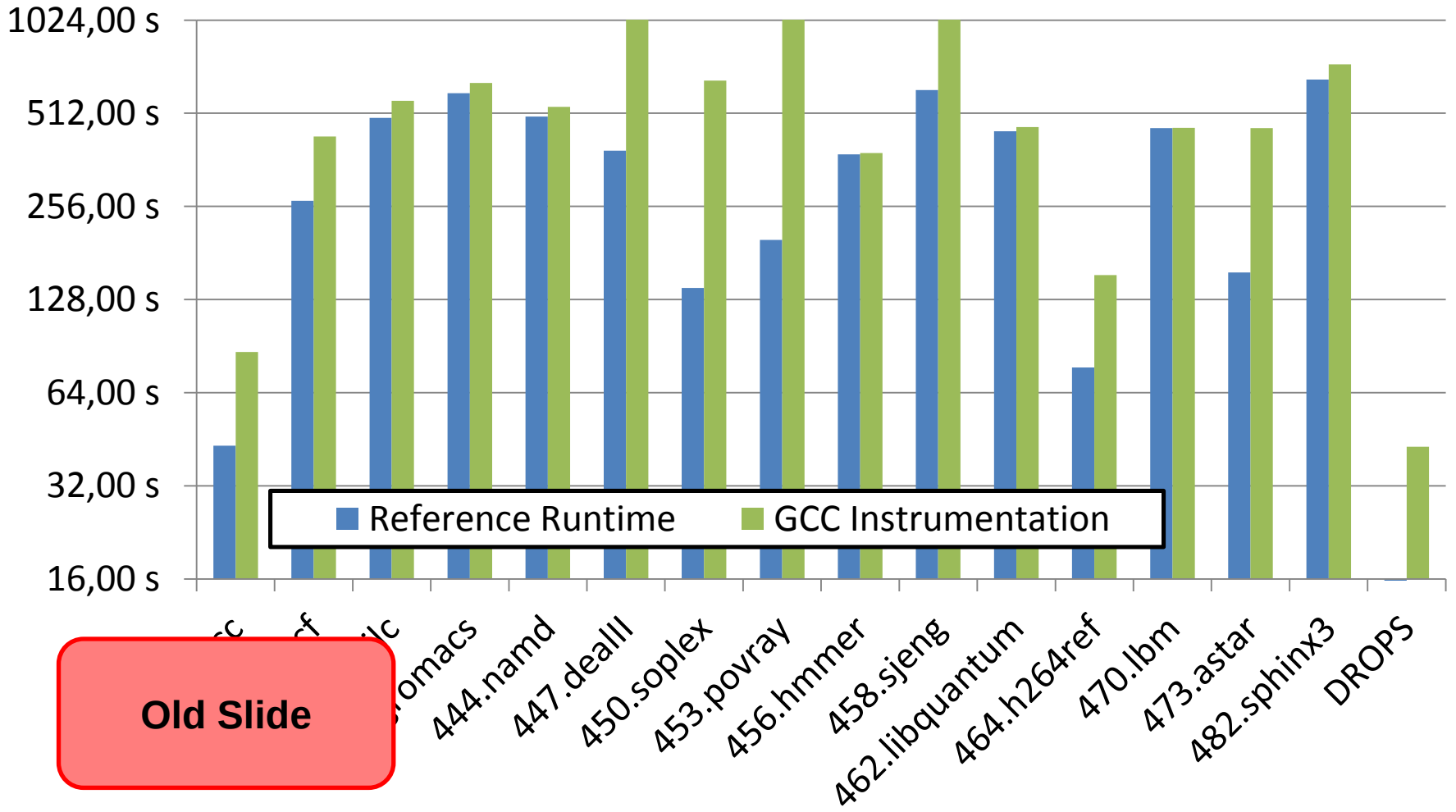
Solution:

- **Instrument Parents and Store an Identifier for the Path taken**
- **Cycle Requires A Stack of Identifiers to Reconstruct Context**



SPEC CPU Measurements Part 1/3: Reference Time & Full GCC Instrumentation

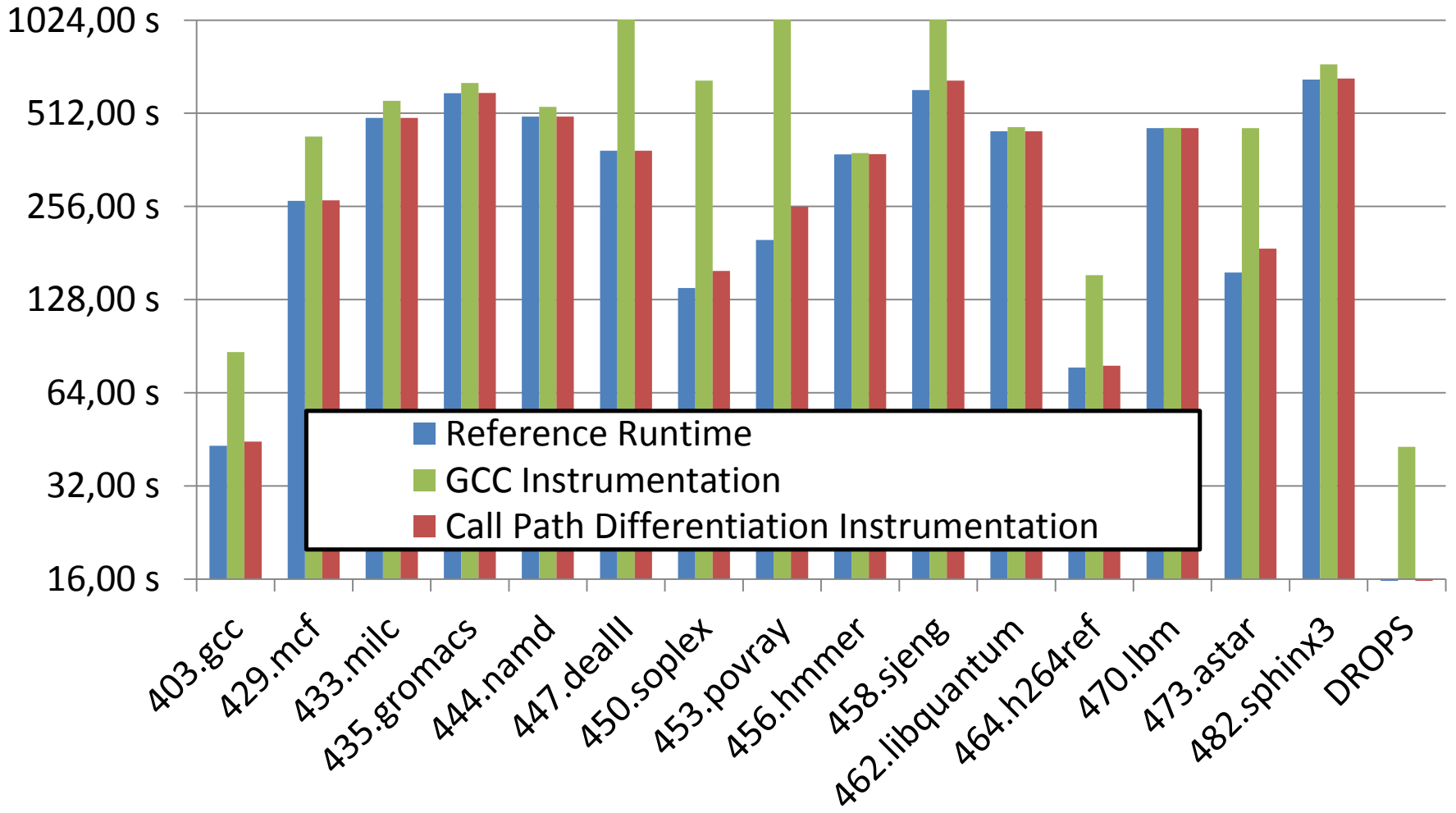
TECHNISCHE UNIVERSITÄT DARMSTADT



SCIENTIFIC COMPUTING

SPEC CPU Measurements Part 2/3: Call Path Differentiation Instrumentation

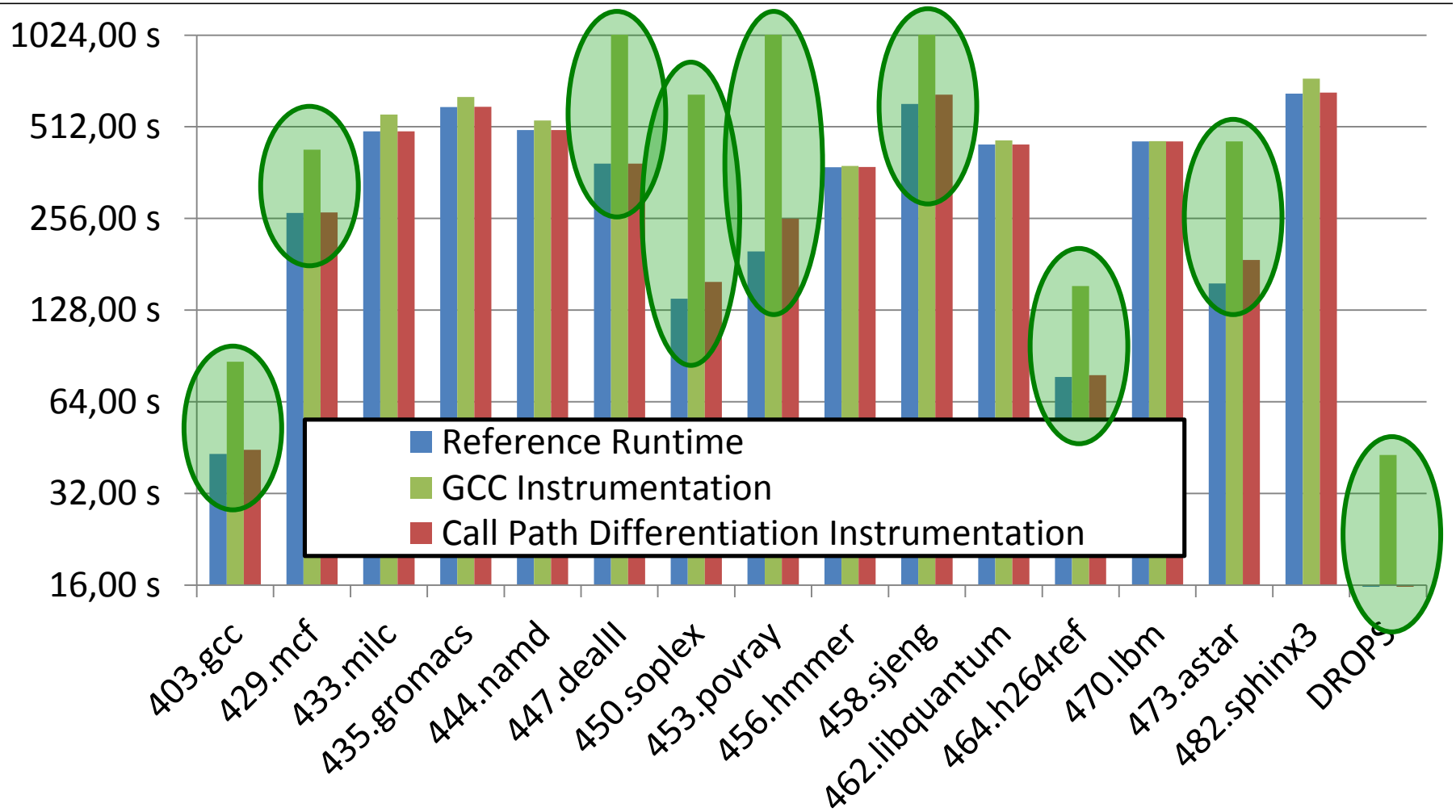
TECHNISCHE UNIVERSITÄT DARMSTADT



SCIENTIFIC COMPUTING

SPEC CPU Measurements Part 2/3: Call Path Differentiation Instrumentation

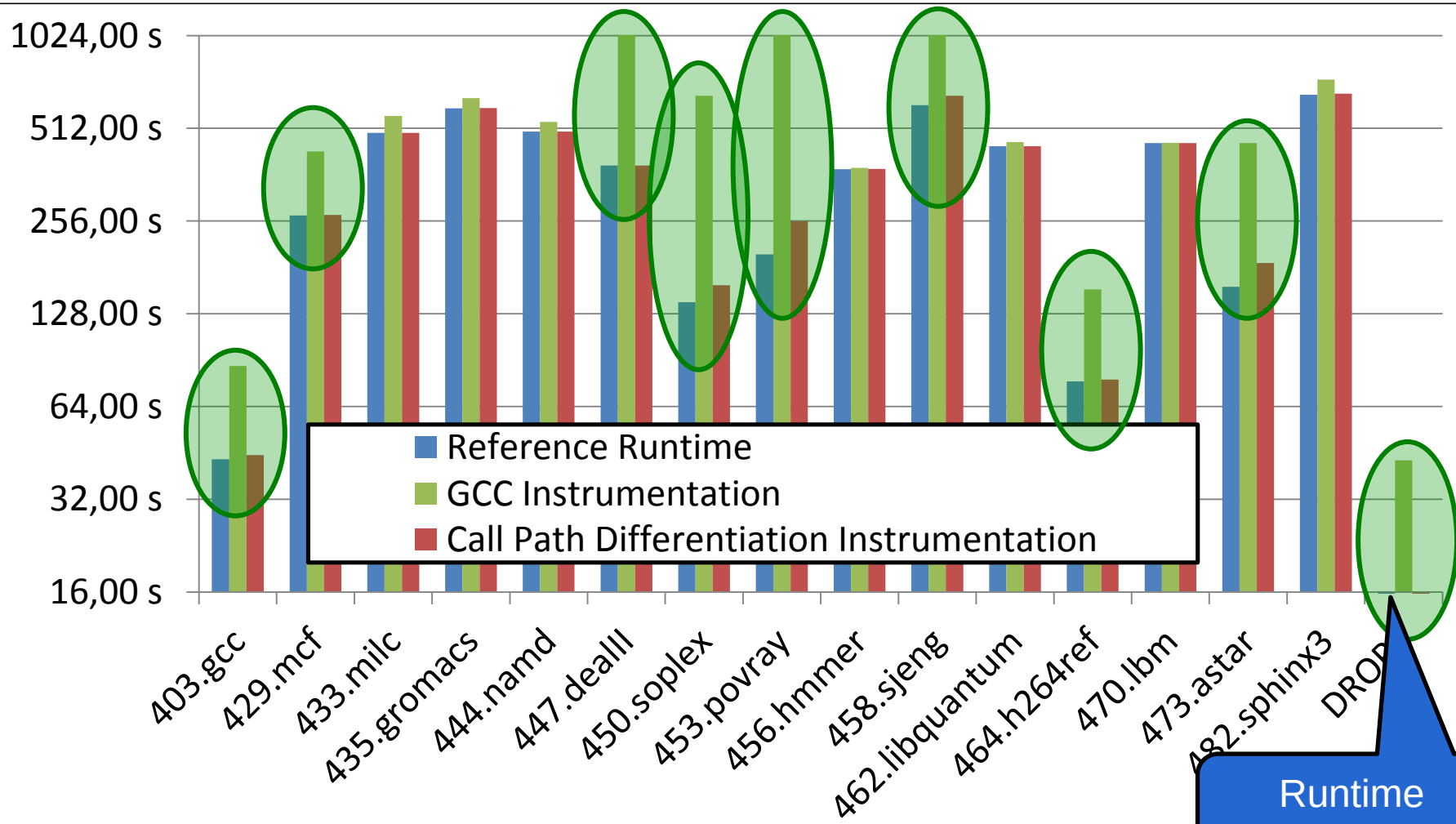
TECHNISCHE UNIVERSITÄT DARMSTADT



SCIENTIFIC COMPUTING

SPEC CPU Measurements Part 2/3: Call Path Differentiation Instrumentation

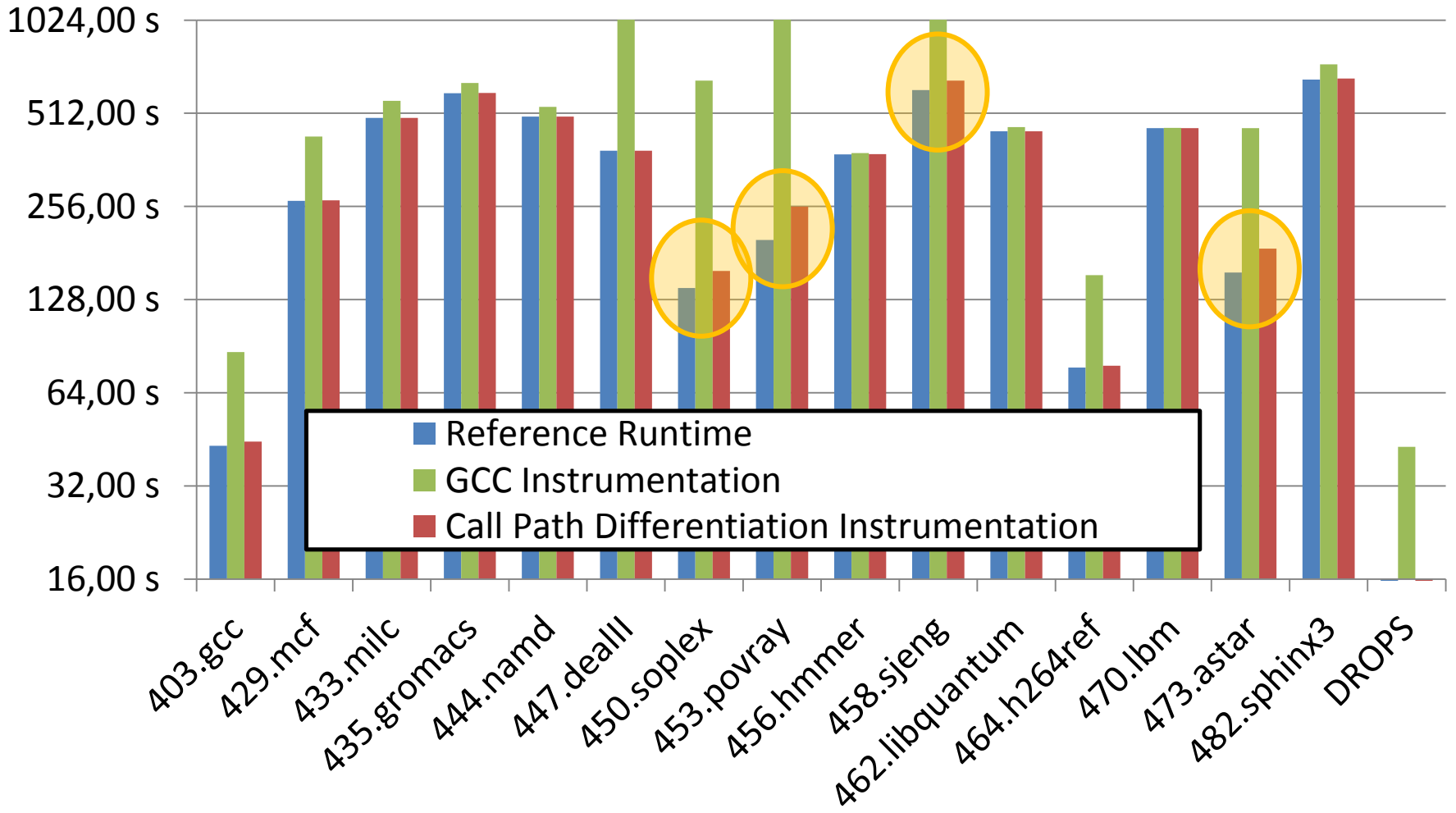
TECHNISCHE UNIVERSITÄT DARMSTADT



SC COMPUTING

SPEC CPU Measurements Part 2/3: Call Path Differentiation Instrumentation

TECHNISCHE UNIVERSITÄT DARMSTADT



SCIENTIFIC COMPUTING

Example: SPEC MPI 104.MILC
Restricting paths to MPI Functions

Num Procs	Base Runtime	Full Instrumentation	MPI Instrumentation	C.P.D. Instrumentation
4	1403.54	5886.09 – 319%	1417.53 - 0.99%	1430.36 – 1.19%

Example: SPEC MPI 104.MILC
Restricting paths to MPI Functions

TECHNISCHE UNIVERSITÄT DARMSTADT

Num Procs	Base Runtime	Full Instrumentation	MPI Instrumentation	C.P.D. Instrumentation
4	1403.54	5886.09 – 319%	1417.53 - 0.99%	1430.36 – 1.19%

Top MILC Functions:

- scalar_mult_add_su3_matrix: $3.41 * 10^9$
- uncompress_anti_hermitian: $2.18 * 10^9$
- make_anti_hermitian: $2.16 * 10^9$

Example: SPEC MPI 104.MILC
Restricting paths to MPI Functions

TECHNISCHE UNIVERSITÄT DARMSTADT

Num Procs	Base Runtime	Full Instrumentation	MPI Instrumentation	C.P.D. Instrumentation
4	1403.54	5886.09 – 319%	1417.53 - 0.99%	1430.36 – 1.19%

Top MILC Functions:

- scalar_mult_add_su3_matrix: $3.41 * 10^9$
- uncompress_anti_hermitian: $2.18 * 10^9$
- make_anti_hermitian: $2.16 * 10^9$

Top MPI Function:

- MPI_Comm_rank: $52.4 * 10^6$

Top CPD Functions:

- mynode: $52.4 * 10^6$
- neighbor_coords_special $29.3 * 10^6$

Example: SPEC MPI 104.MILC
Restricting paths to MPI Functions

TECHNISCHE UNIVERSITÄT DARMSTADT

Num Procs	Base Runtime	Full Instrumentation	MPI Instrumentation	C.P.D. Instrumentation
4	1403.54	5886.09 – 319%	1417.53 - 0.99%	1430.36 – 1.19%

Top MILC Functions:

- scalar_mult_add_su3_matrix: $3.41 * 10^9$
- uncompress_anti_hermitian: $2.18 * 10^9$
- make_anti_hermitian: $2.16 * 10^9$

Top MPI Function:

- MPI_Comm_rank: $52.4 * 10^6$

Top CPD Functions:

- mynode: $52.4 * 10^6$
- neighbor_coords_special $29.3 * 10^6$

Type	Calls		No. of. Functions			No. of. Functions
Full	$23.9 * 10^9$	Analysis Based	124 of	255	Profile Based	124
C.P.D.	$81.8 * 10^6$		57 of	135		57
MPI-Only	$52.4 * 10^6$		11			11

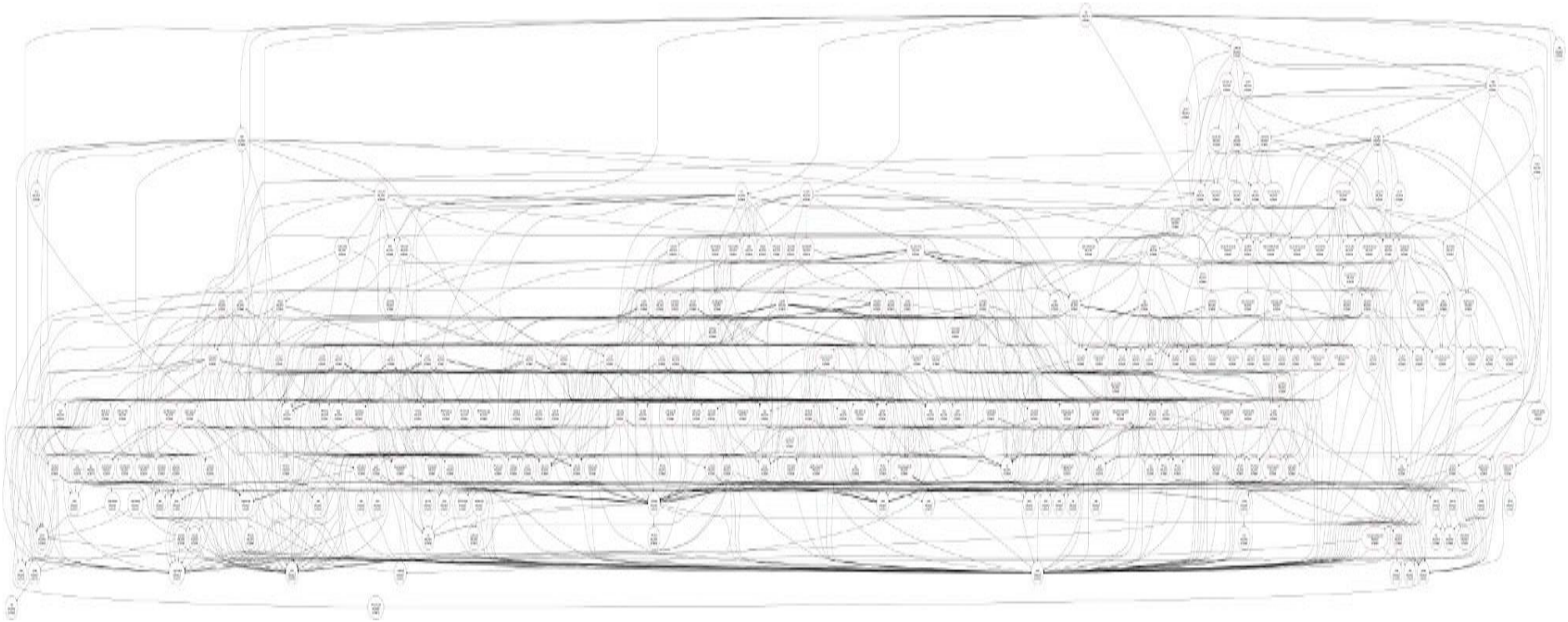
SCIENTIFIC COMPUTING

The Call Graph of SPEC 104.MILC

TECHNISCHE UNIVERSITÄT DARMSTADT

Nodes – Functions: **256**

Edges – Function Calls: **748**



Future Work

TECHNISCHE UNIVERSITÄT DARMSTADT

Challenges

- **Incomplete Static / Dynamic Call Graphs**
- **Tool Integration**

Improvements

- **Improved Instrumentation**
  - Better Locations
  - Fewer Probes
- **Compiler Transformations to Reduce Overhead**

Take Away

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Call Path Differentiation Instrumentation:**
  - Instrumentation to Differentiate Call-Paths to Target Functions
    - Example: MPI Functions
  - Lower Overhead for MPI Functions Call Paths

Questions?