amdb:
An Access Method Debugging and Analysis Tool

Marcel Kornacker, Mehul Shah,
Joe Hellerstein
UC Berkeley

Motivation

- **Access method (AM) design and tuning is a black art.**
  - Which AM do I use to index my non-traditional data type?
  - How well do existing AMs perform for my workload?
- **Generalized search trees (GiST) provide a framework for AM implementations**
- **amdb is a debugging and analysis tool for GiST-implemented AMs**

Overview of Generalized Search Trees

GiST AM Parameters

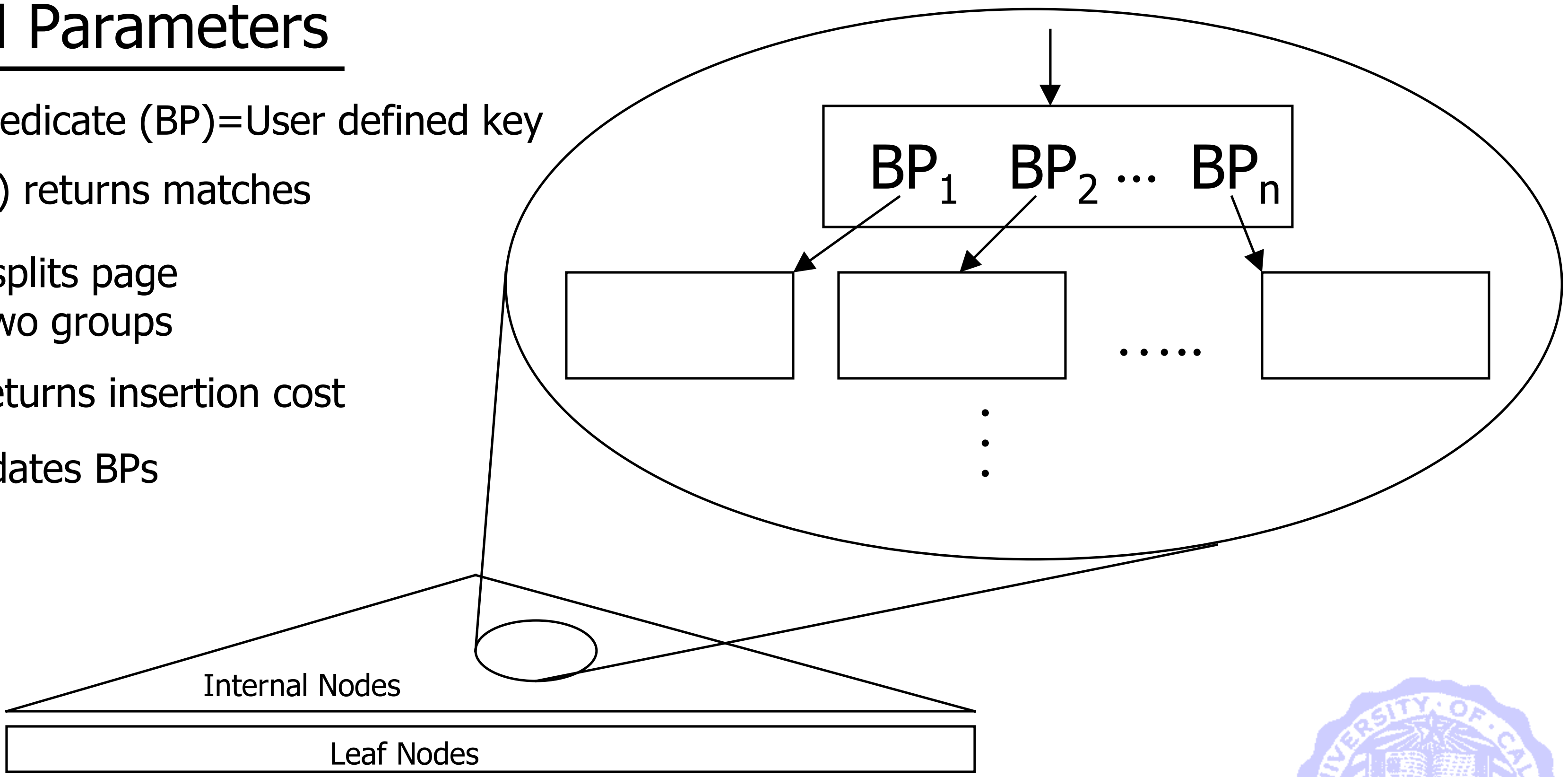
Bounding predicate (BP)=User defined key

Consistent( ) returns matches

PickSplit( ) splits page
items into two groups

Penalty( ) returns insertion cost

Union( ) updates BPs



amdb Features

- Tracing of insertions, deletions, and searches
- Debugging operations: breakpoints on node splits, updates, traversals, and other events
- Global and structural views of tree allow navigation and provide visual summary of statistics
- Graphical and textual view of node contents
- Analysis of workloads and tree structure
- Analysis of GiST AM parameters: BPs, PickSplit(), and Penalty()

Debugging Operations

Stepping Controls

Tree View

Shows structural organization of index.

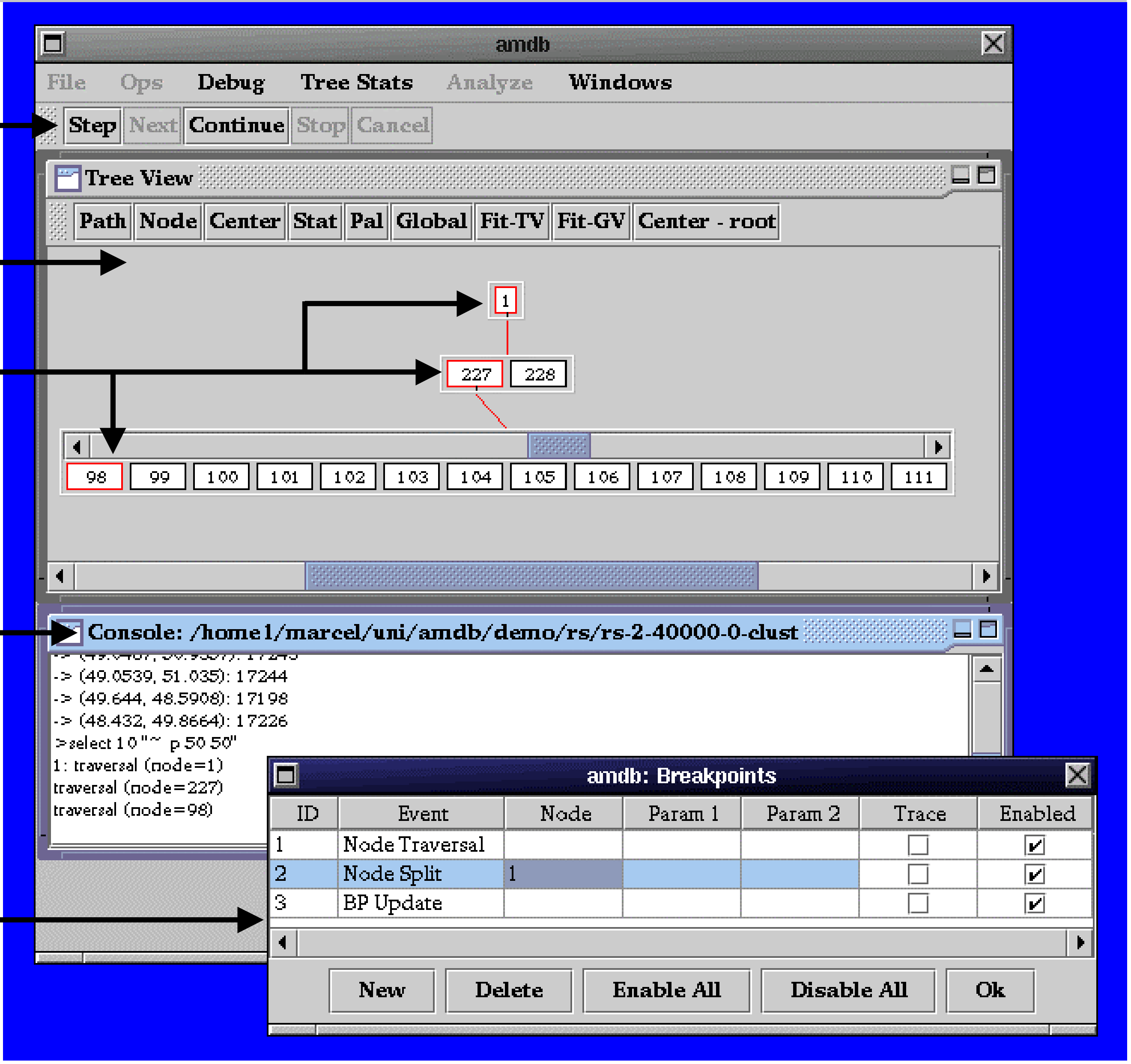
Highlights current traversal path during debugging steps.

Console window

Displays search results, debugging output, and other status info.

Breakpoint Table

Defines and enables breakpoints on events



```
-> (49.0539, 51.035): 17244
-> (49.644, 48.5908): 17198
-> (48.432, 49.8664): 17226
>select 10 "~ p 50 50"
1: traversal (node=1)
traversal (node=227)
traversal (node=98)
```

ID	Event	Node	Param 1	Param 2	Trace	Enabled
1	Node Traversal				☐	✓
2	Node Split	1			☐	✓
3	BP Update				☐	✓

Node Visualization

Node View

Displays bounding predicates (BPs) and items within nodes.

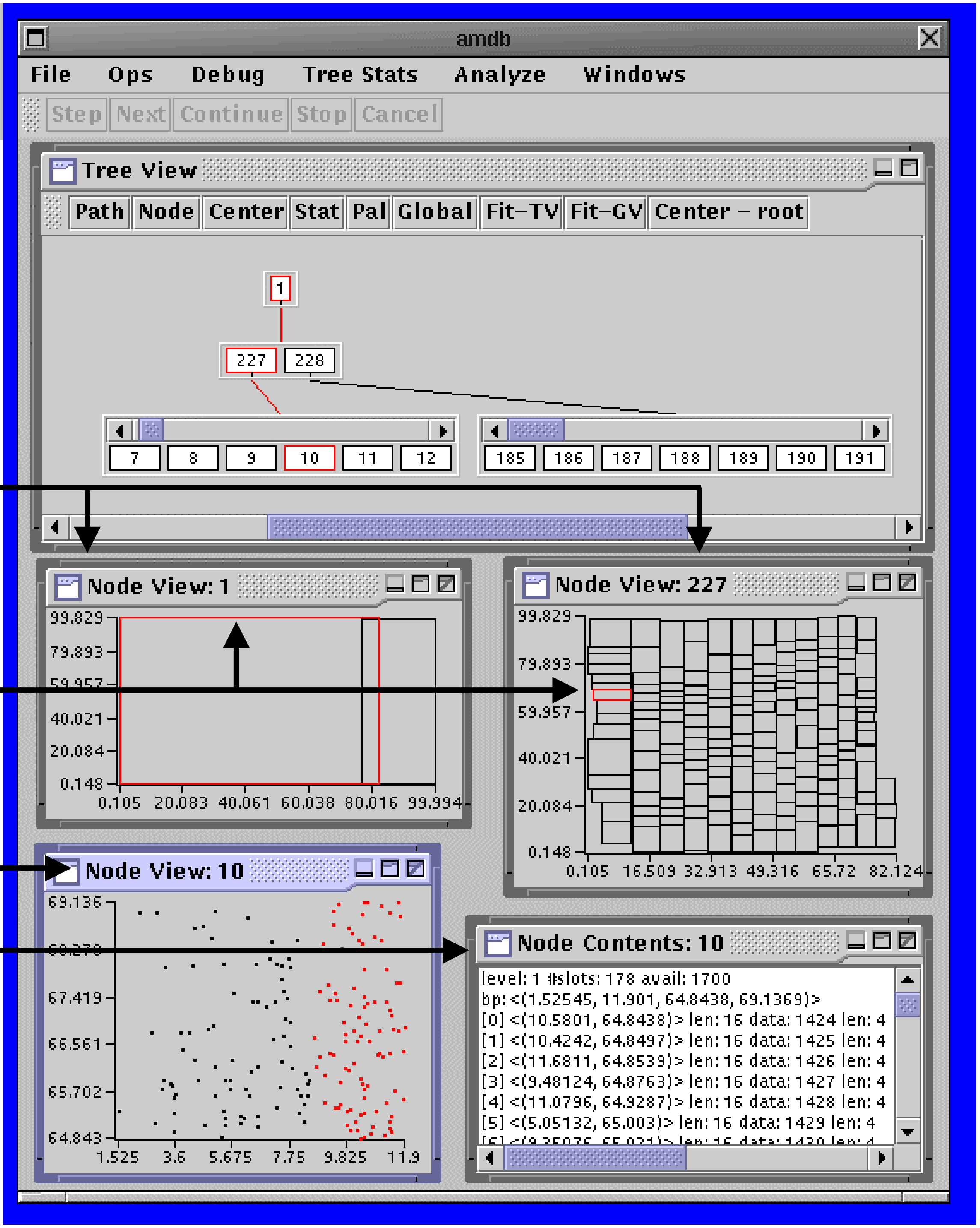
Highlights BPs on current traversal path.

Split Visualization

Shows how BPs or data items are divided with PickSplit( )

Node Contents

Provides textual description of node



```
level: 1 #slots: 178 avail: 1700
bp: <(1.52545, 11.901, 64.8438, 69.1369)>
[0] <(10.5801, 64.8438)> len: 16 data: 1424 len: 4
[1] <(10.4242, 64.8497)> len: 16 data: 1425 len: 4
[2] <(11.6811, 64.8539)> len: 16 data: 1426 len: 4
[3] <(9.48124, 64.8763)> len: 16 data: 1427 len: 4
[4] <(11.0796, 64.9287)> len: 16 data: 1428 len: 4
[5] <(5.05132, 65.003)> len: 16 data: 1429 len: 4
[6] <(9.35076, 65.021)> len: 16 data: 1430 len: 4
```

Analysis Framework

- **Analysis of index in context of user-specified workload:**
  - performance cannot be assessed independently of workload
  - metrics must reflect workload performance, not data semantics
- **Analysis procedure:**
  - Is data clusterable, given workload?
  - Assess tree and use metrics to pinpoint defects
  - Evaluate performance of PickSplit( ) and Penalty( ) methods

Performance Metrics

- **Factors affecting performance**
  - Clustering, Page Utilization, BP coverage and size
- **per-query metrics:**
  - based on required vs. observed I/Os
  - measure performance loss/overhead for each factor
- **per-node metrics:**
  - measure contributions to performance loss over entire workload
- **Penalty() and PickSplit() metrics:**
  - measure deterioration of workload performance

Leaf-Level Statistics

Global View

Provides summary of node statistics for entire tree

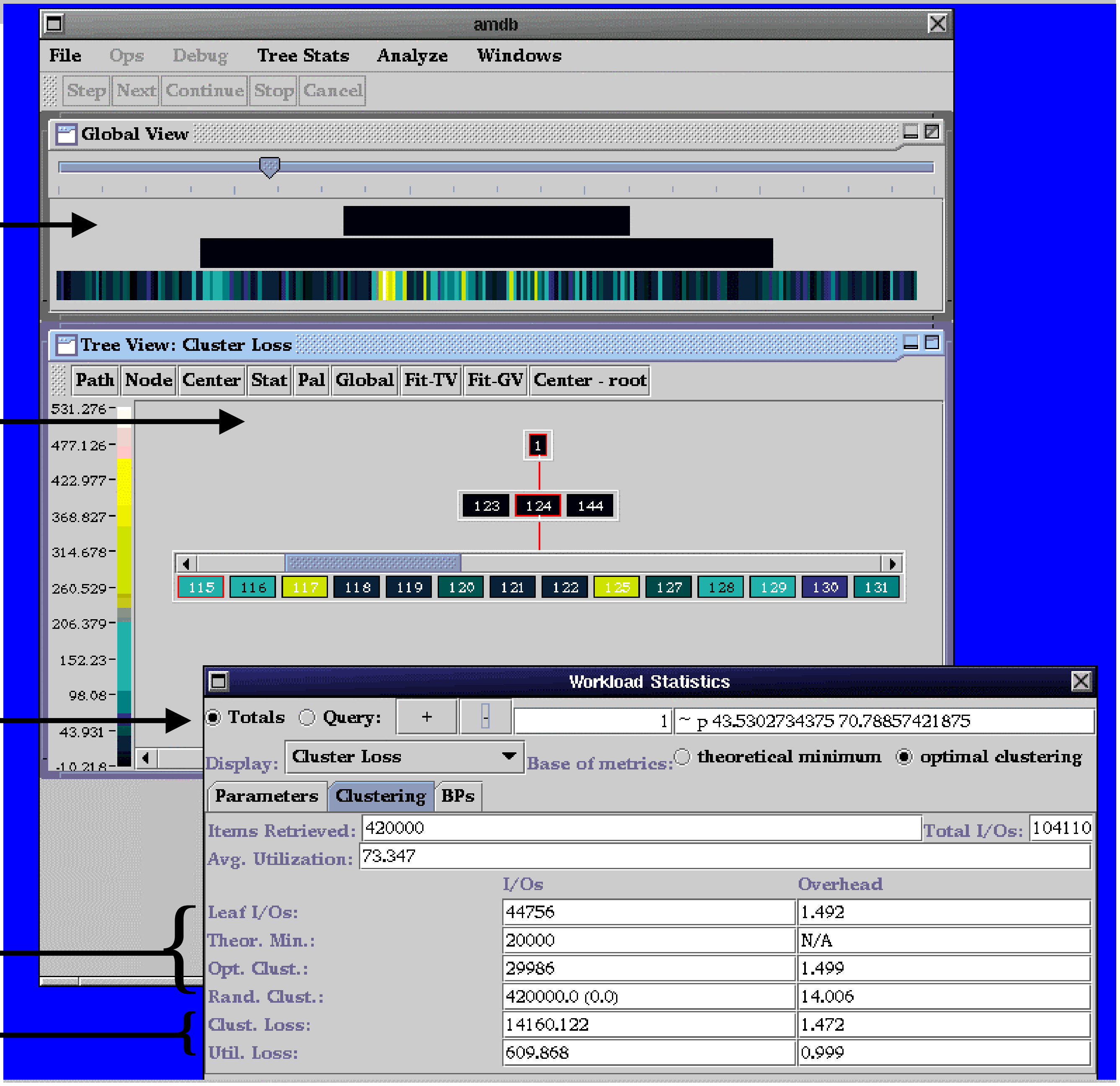
Tree View

Also displays node stats

Total or per query breakdown

I/O counts and corresponding overheads under various scenarios

Breakdown of losses against optimal clustering



[✓ Totals] Query: + 1 ~ p 43.5302734375 70.78857421875

Display: Cluster Loss Base of metrics: theoretical minimum [✓ optimal clustering]

Parameters | Clustering | BPs

Items Retrieved: 420000 Total I/Os: 104110

Avg. Utilization: 73.347

	I/Os	Overhead
Leaf I/Os:	44756	1.492
Theor. Min.:	20000	N/A
Opt. Clust.:	29986	1.499
Rand. Clust.:	420000.0 (0.0)	14.006
Clust. Loss:	14160.122	1.472
Util. Loss:	609.868	0.999

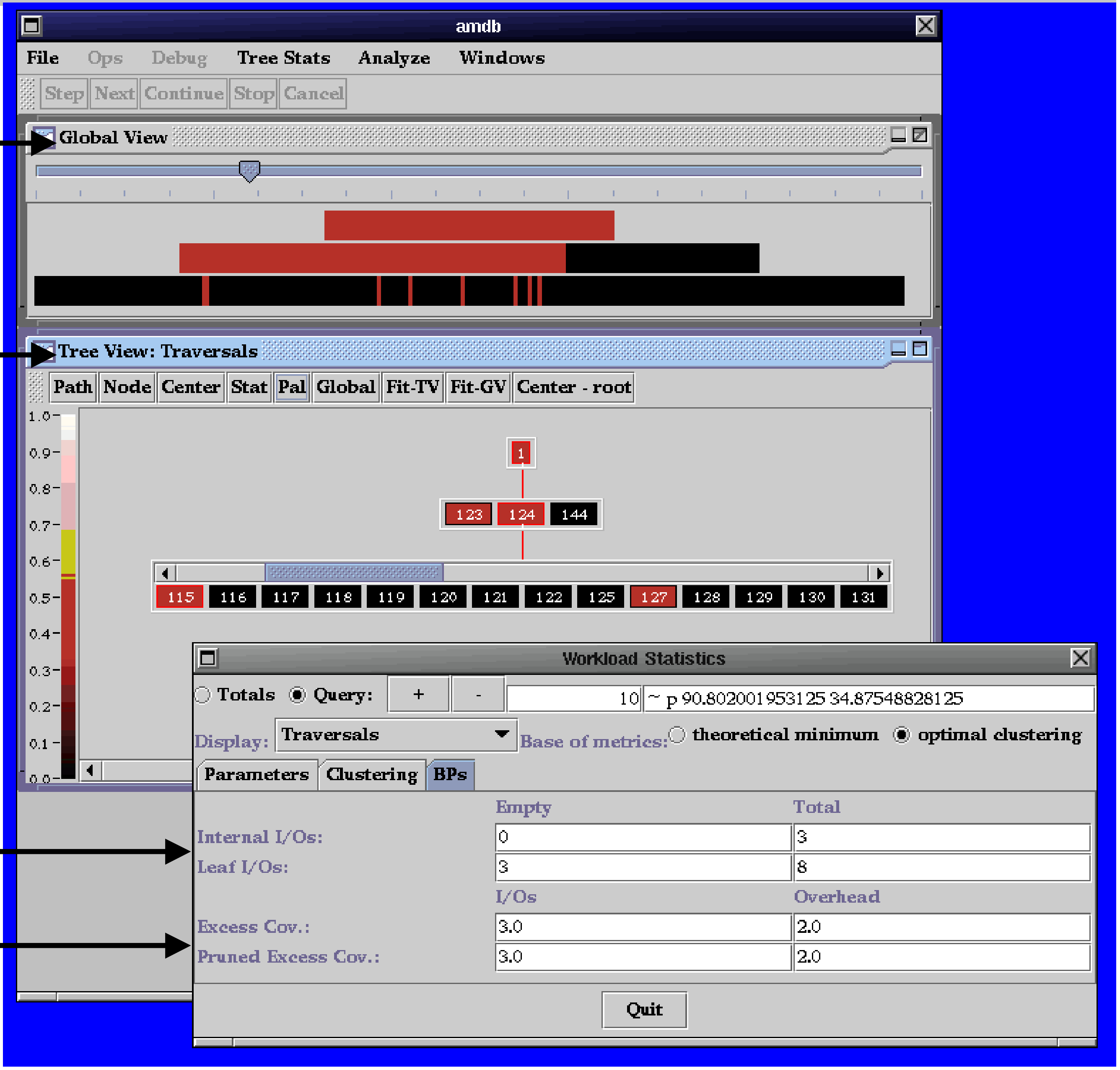
Bounding Predicate Statistics

Views highlight nodes traversed by query

Query breakdown in terms of "empty" and required I/O.

Excess Coverage

Overheads due to loose/wide BPs



Totals [✓ Query:] 10 ~ p 90.802001953125 34.87548828125

Display: Traversals

Base of metrics: theoretical minimum [✓ optimal clustering]

	Empty	Total
Internal I/Os:	0	3
Leaf I/Os:	3	8
	I/Os	Overhead
Excess Cov.:	3.0	2.0
Pruned Excess Cov.:	3.0	2.0