

# Taming the Wild West of HPC Resource Management

Thoughts on scalable performance  
management in exascale computing

Carlos Maltzahn

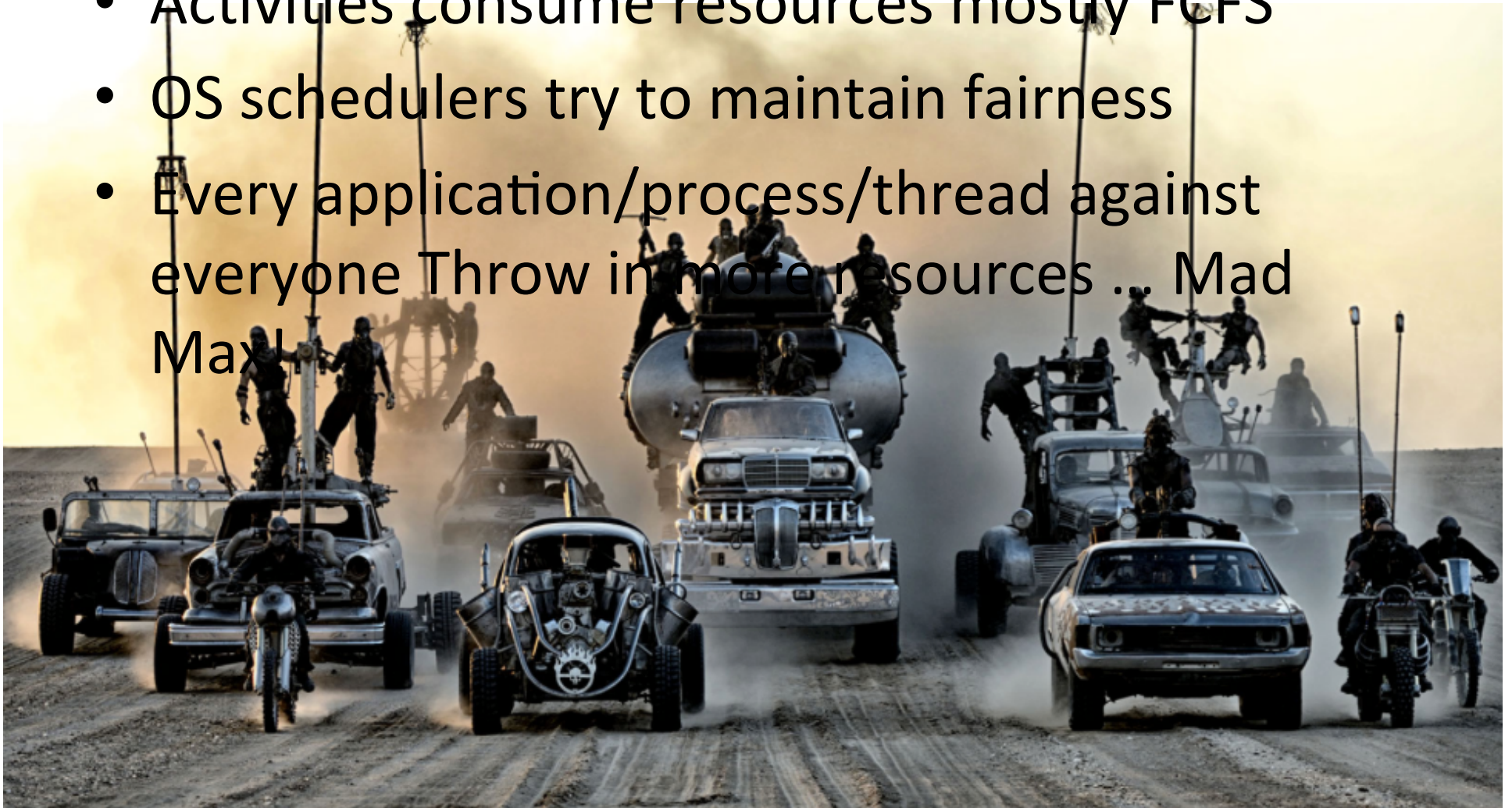
10/19/15

# Riff on two recent papers

- Jonathan Mace, Peter Bodik, Madanlal Musuvathi, and Rodrigo Fonseca. Retro: targeted resource management in multi-tenant distributed systems. In NSDI '15
- Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, Ion Stoica. BlinkDB: Queries with Bounded Errors and Bounded Response Times on Very Large Data. In EuroSys'13 (Best Paper)

# The Wild West

- Activities consume resources mostly FCFS
- OS schedulers try to maintain fairness
- Every application/process/thread against everyone Throw in more resources ... Mad Max!



# Law & Order

- Law:
  - Global policies to make a system efficient
- Order:
  - Mechanisms to locally monitor and enforce



# Law & Order

- Law:
  - Global policies to make a system efficient
- Order:
  - Mechanisms to locally monitor and enforce
- Retro (the paper):
  - *Workflows, Resources, Control points*
  - *Abstract Load and Slowdown*
  - Policies can be applied to any resource and combinations thereof.
  - Relies on *causal metadata facility*

# Causal Metadata Propagation

- **End-to-End tracing**
  - Similar, but incompatible contents
- **Same *propagation***
  - Flow along thread while working on same activity
  - Store and retrieve when deferred (queues, callbacks)
  - Copy when forking, merge when joining
  - Serialize and send with messages
  - Deserialize and set when receiving messages

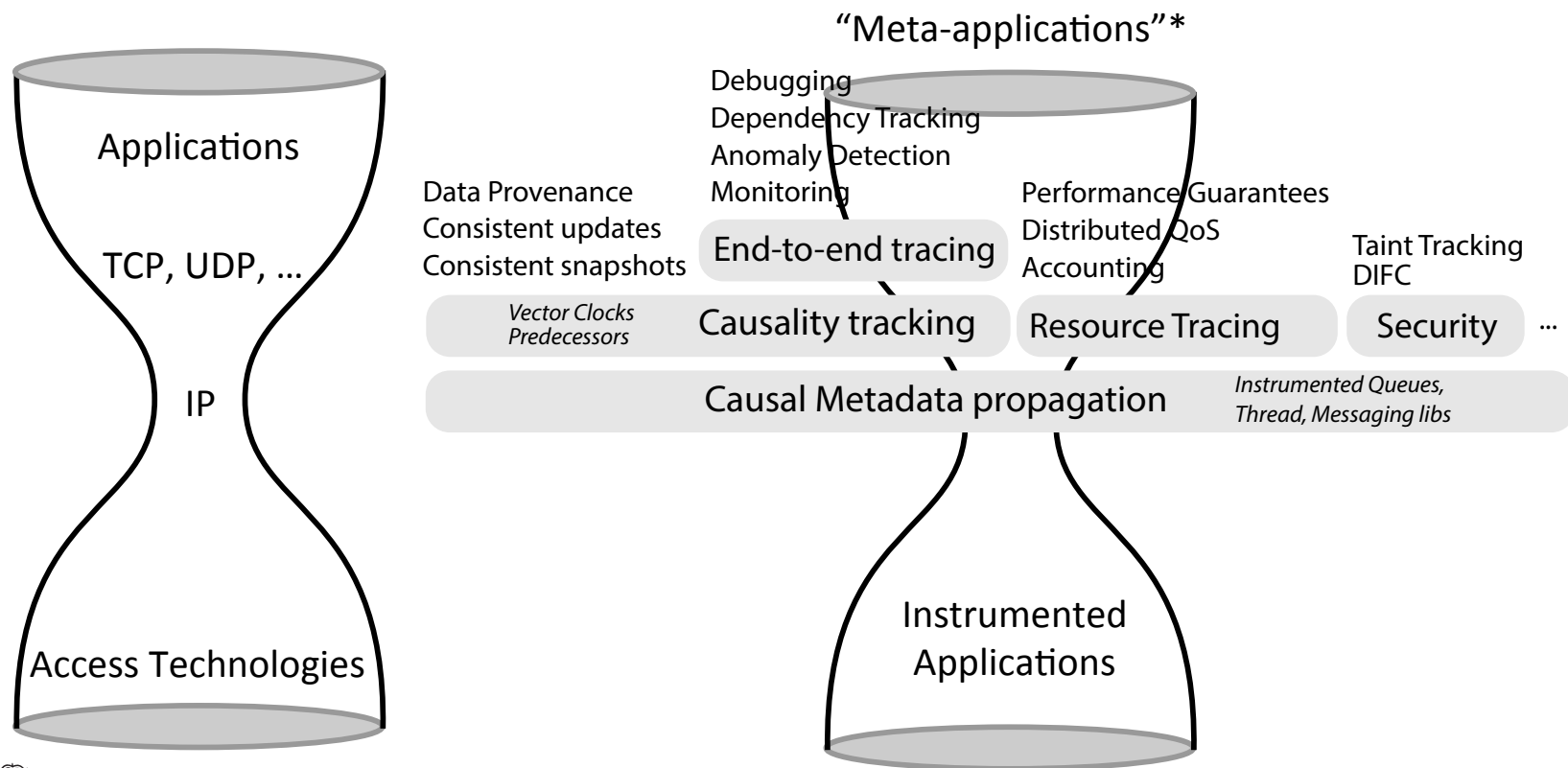


# Causal Metadata Propagation

- **Not hard, but subtle sometimes**
- **Requires commitment, touches many places in the code**
- **Difficult to completely automate**
  - Sometimes the causality is at a layer above the one being instrumented
- **You will want to do this only once...**



# Obligatory ugly hourglass picture



\*Causeway (Chanda et al., Middleware 2005) used this term

Slide from Rodrigo Fonseca's presentation at HPTS 2015



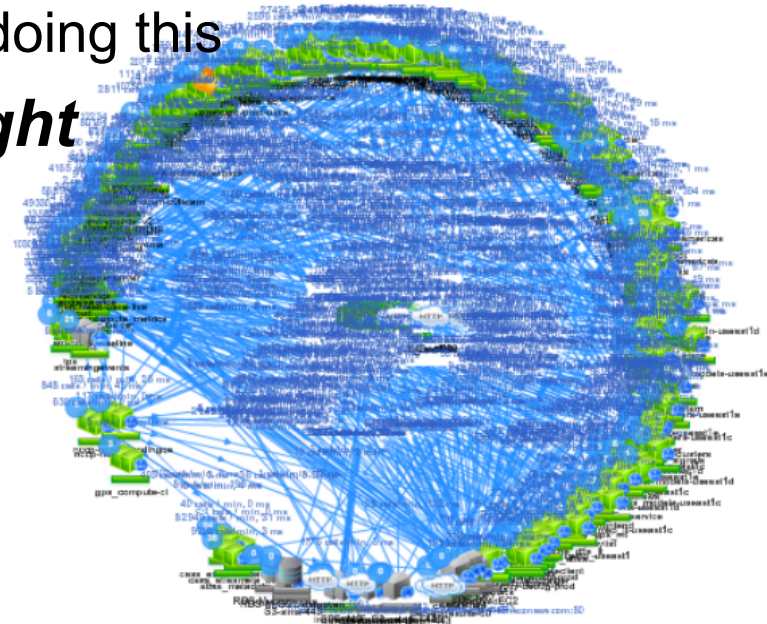
# Proposal: Baggage

- **API and guidelines for causal metadata propagation**
- **Separate propagation from semantics of data**
- **Instrument systems once, “baggage compliant”**
- **Allow multiple meta-applications**



# Why now?

- We are losing track...
- Huge momentum (Zipkin, HTrace, ...)
  - People care and ARE doing this
- Right time to do it *right*



# Baggage API

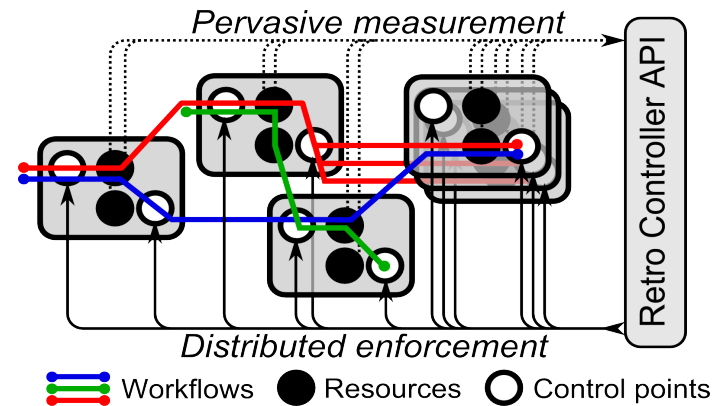
- **PACK, UNPACK**
  - Data is key-value pairs
- **SERIALIZE, DESERIALIZE**
  - Uses protocol buffers for serialization
- **SPLIT, JOIN**
  - Apply when forking / joining
  - Use Interval Tree Clocks to correctly keep track of data



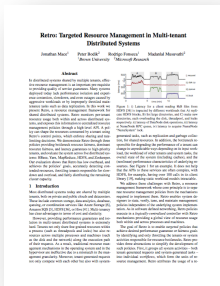
Paulo Sérgio Almeida, Carlos Baquero, and Victor Fonte. Interval tree clocks: a logical clock for dynamic systems. In *Opodis '08*.

Slide from Rodrigo Fonseca's presentation at HPTS 2015

# Retro



- Propagates TenantID across a system for real-time resource management
- Instrumented most of the Hadoop stack
- Allows several policies – e.g., DRF, LatencySLO
- Treats background / foreground tasks uniformly



Jonathan Mace, Peter Bodik, Madanlal Musuvathi, and Rodrigo Fonseca. Retro: targeted resource management in multi-tenant distributed systems. In *NSDI '15*

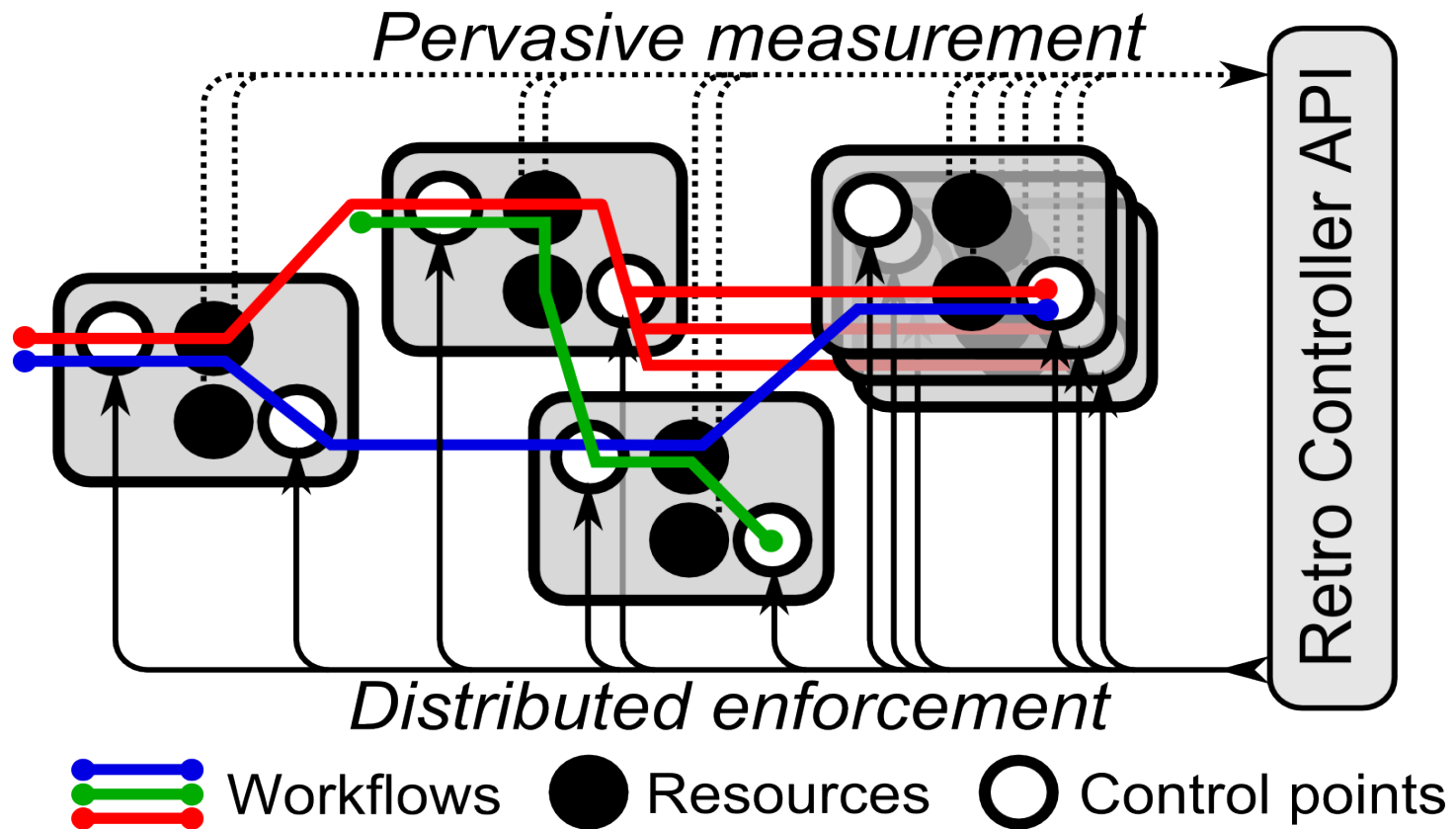


Figure from Rodrigo Fonseca's presentation at HPTS 2015

# Best Case vs Worst Case Estimates

- Retro: models best case and measures load & slowdowns for each (workflow, resource)
  - No isolation, no absolute guarantees
- Horizon [HPDC'10]: uses 90<sup>th</sup> %ile of worst case times to determine deadlines
  - Good isolation, meets >90% of all deadlines
  - Instead of load & slowdown, use utilization and urgency based on deadlines

# BlinkDB

```
SELECT avg(sessionTime)  
FROM Table  
WHERE city='San Francisco'  
WITHIN 2 SECONDS
```

Queries with Time Bounds

```
SELECT avg(sessionTime)  
FROM Table  
WHERE city='San Francisco'  
ERROR 0.1 CONFIDENCE 95.0%
```

Queries with Error Bounds

# Latency vs Resolution Trade-off

- BlinkDB:
  - Offline sampling based on queries
  - Online latency error trade-off modeling using subsampling of different samples
- SIRIUS:
  - Multi-resolution data placement based on workload profiles (could be a separate “warm-up” job)
  - Online latency resolution trade-off modeling using small measurements



# Summary

- Causal metadata propagation
  - Enables end-to-end debugging, performance guarantees, security
- Workflow, resource, control point
  - Good for tracking & enforcing resource utilization
- Worst case, not best case estimates
  - For utilization and deadlines
  - Good for performance isolation
- Multi-resolution as a form of sampling
  - Which resolutions of what?
  - Online modeling of latency/resolution trade-offs
  - Multi-resolution “multi-placement”

# Thanks!



## MULTIPASS

There are some things money can't buy, for everything else there's  
MultiPass.