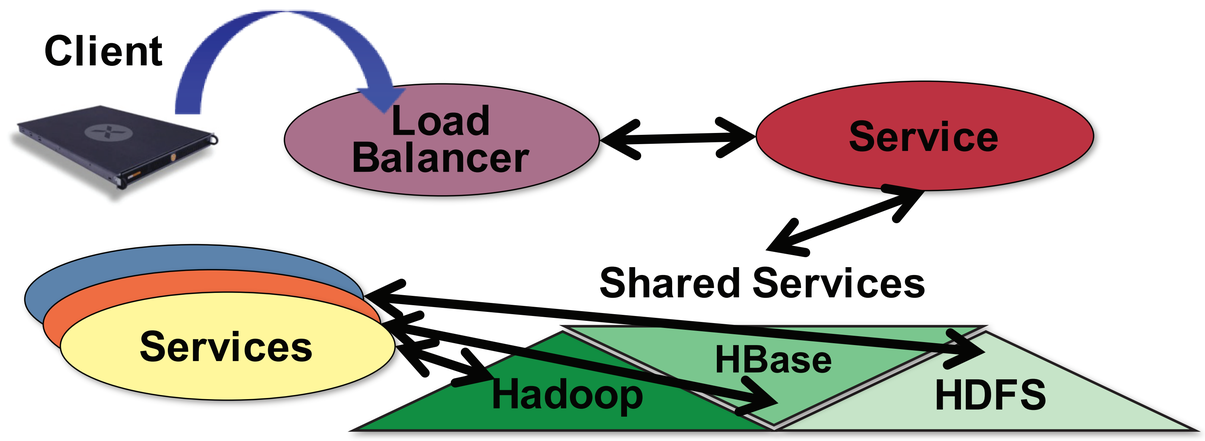
MANAGING INTER-SERVICE PERFORMANCE DEPENDENCIES

Elie Krevat, Greg Ganger (CMU)

OVERVIEW

- Many services are composed of other services
- Downstream service delays have cascading effects
- High-level service may need faster response
- Goal: Automatically identify + mitigate issues

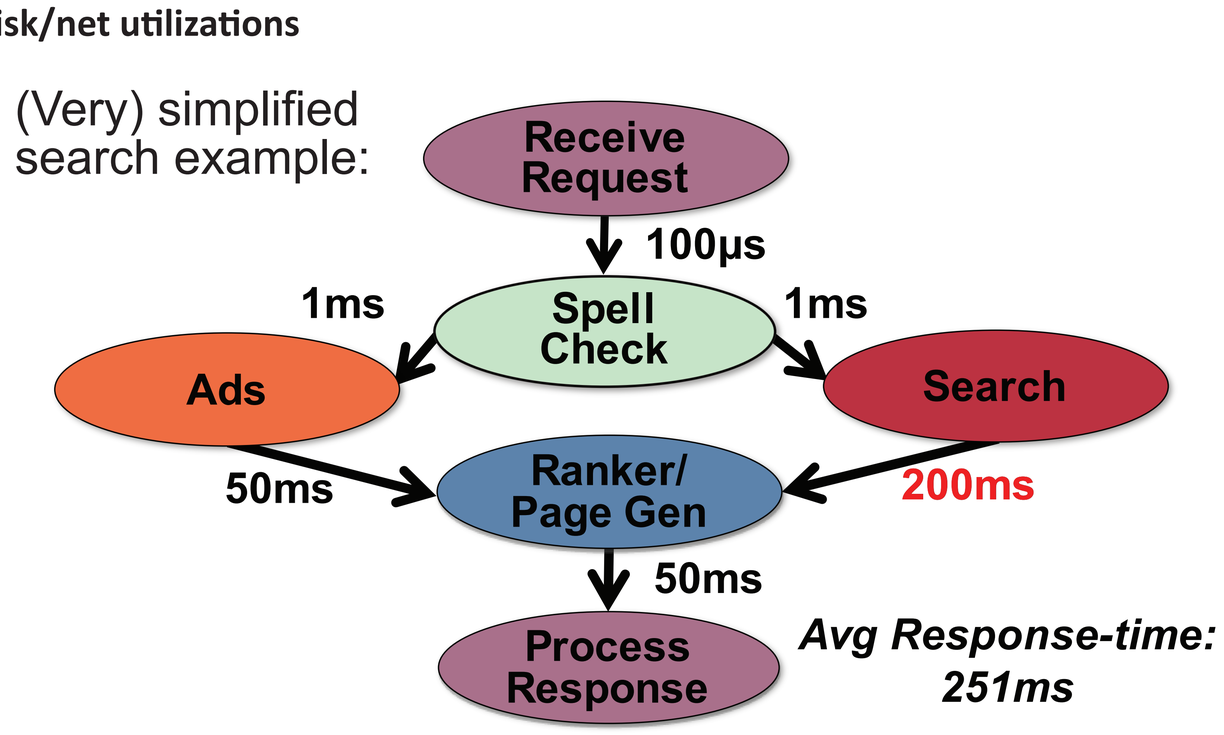


SHARED SERVICE DEPENDENCIES

- Dependencies exist with downstream services
  - Managed by separate teams
  - Respond to requests from multiple services

INITIAL APPROACH

- Find bottleneck service from end-to-end trace
- Evaluate potential gain
  - Extract critical path from trace analysis
  - Analyze frequency & distribution of responses
  - Inspect measured CPU/disk/net utilizations
- Apply resources



END-TO-END TRACES

- Tracks flow of requests through system
- Low overhead when sampling requests
- Usage is growing (e.g., Google Dapper)
- Can expose inter-service dependencies & effects

APPLYING RESOURCES

- Idyllic model: Per-client service levels (SLAs)
  - Each service negotiates own contract
  - Renegotiate, if downstream too slow
  - Downstream service figures out how
- Alternative model: Fixed global perf targets
  - Option #1: Request global target increase
  - Option #2: Run dedicated instance locally
  - Option #3: "Lend" priority to bottleneck service

GETTING STARTED

- Refining expectations re:
  - Inter-service performance dependencies
  - How service levels are managed
  - Options for applying resources
- Needed: traces, case studies, anecdotes

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · Princeton University · UC Berkeley