



OOoCon 2005

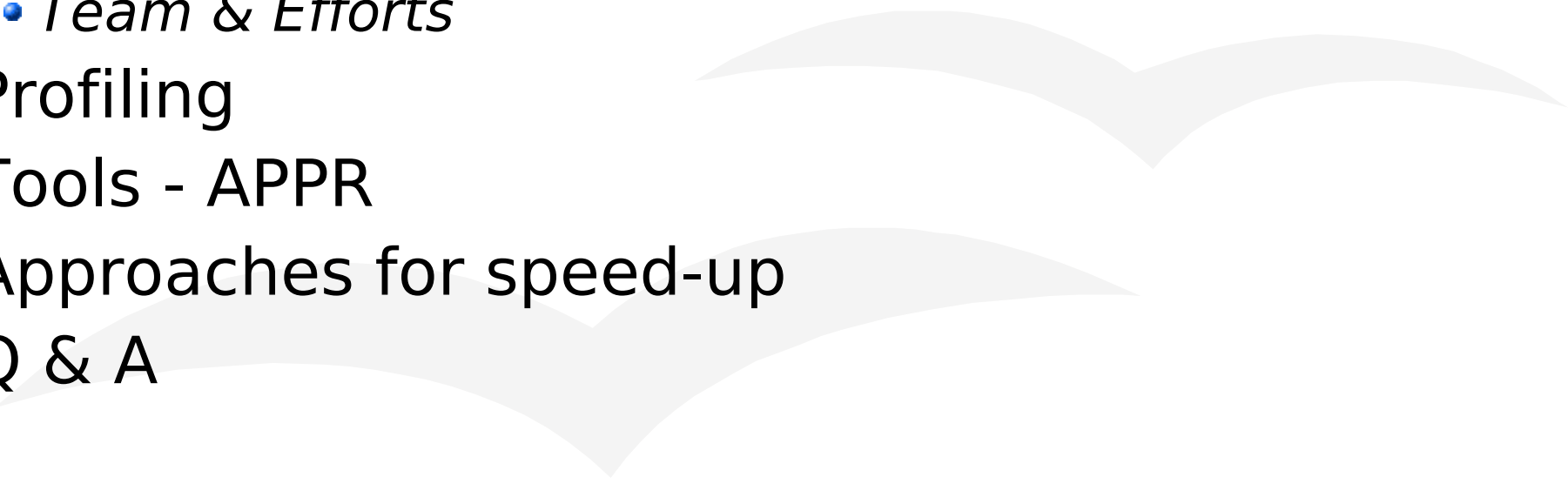
# **Speeding OpenOffice Startup**

## **Profiling, Tools & Approaches**

**Dhananjay Keskar**  
**Michael Leibowitz**



# Agenda

- Introductions
    - *Intel & OpenOffice*
    - *Team & Efforts*
  - Profiling
  - Tools - APPR
  - Approaches for speed-up
  - Q & A
- 



# Intel and OpenOffice

- Intel as the platform of choice for PC software
  - *Enable customer choices on Intel® Platforms*
- Channel Software Operation
  - *Help Intel® Channel address new market needs*
- Linux\* on Intel platforms
  - *Support customer demand for Linux & OSS*
  - *Help the channel build & sell PCs running Linux*
    - *Intel® Quick Start Kit for Linux*
  - *Invest & Participate in OSS Community*
- Robust, viable Desktop Linux
  - *Office automation suite is key part of PC usage*
  - *OpenOffice is an excellent choice on Linux*

***Contribute to improve and optimize OpenOffice***



# About Us

- Intel OpenOffice Team

- *dkeskar, mikeleib, ndev, yli34, maxy, bspencer...*
- *Dhananjay Keskar, Michael Leibowitz, Naren Devaiah, Yin Li, Max Alt, Bob Spencer, Stan Wang, Max Yu, Gordon Jin, Weichuan Lin*

- Timeline:

- *Jan/Feb - Gap Analysis – Features & Interoperability*
- *Mar/Apr - Build; read source, dev manuals*
  - *Focused Areas: Start-up, Rendering, Threading*
- *May/June – Project approval, initial profiling*
- *July – some quick patches*
- *Aug/Sep – tools, startup investigations & approaches*
  - *Expanded Team – EM64T*



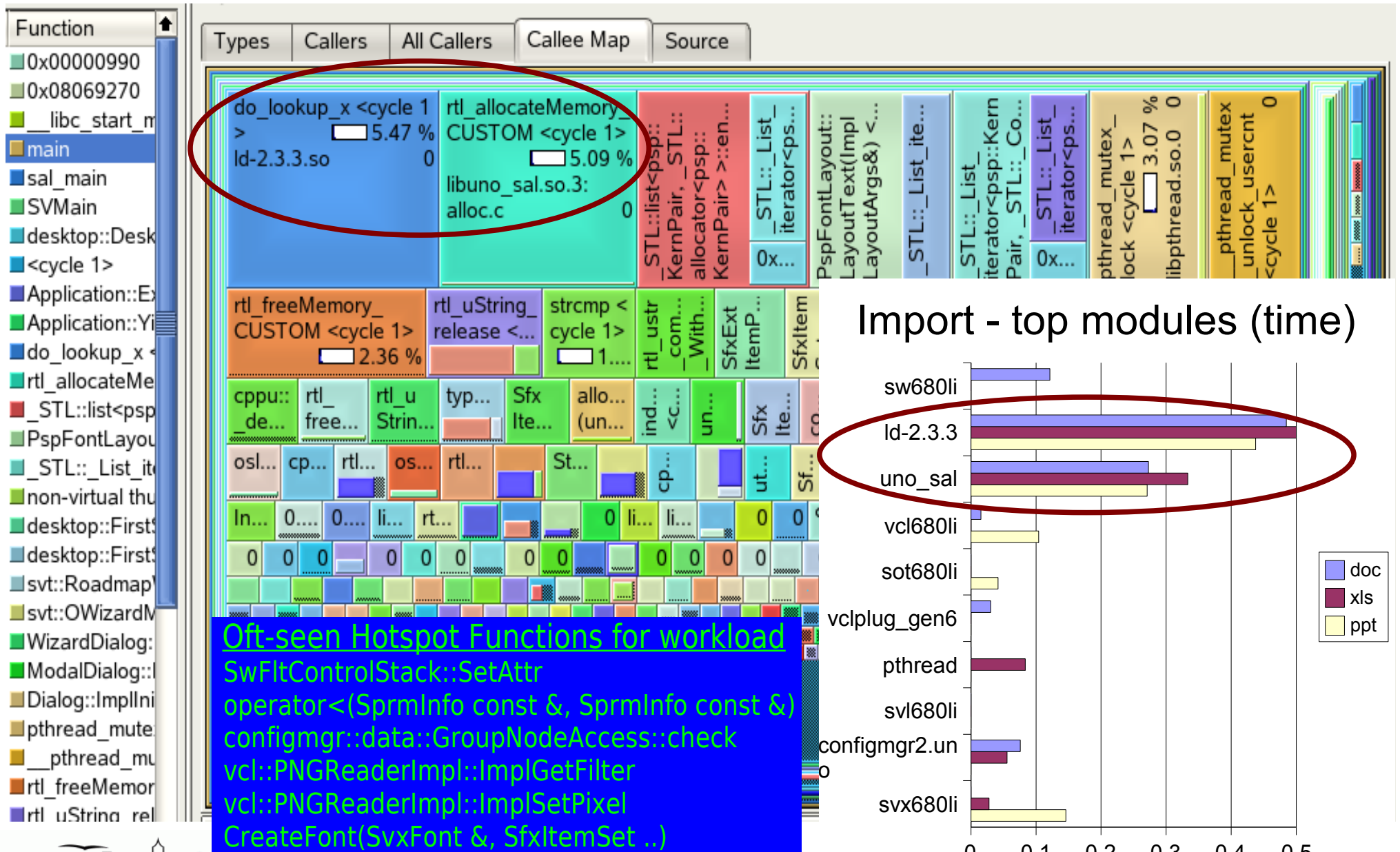
# Profiling



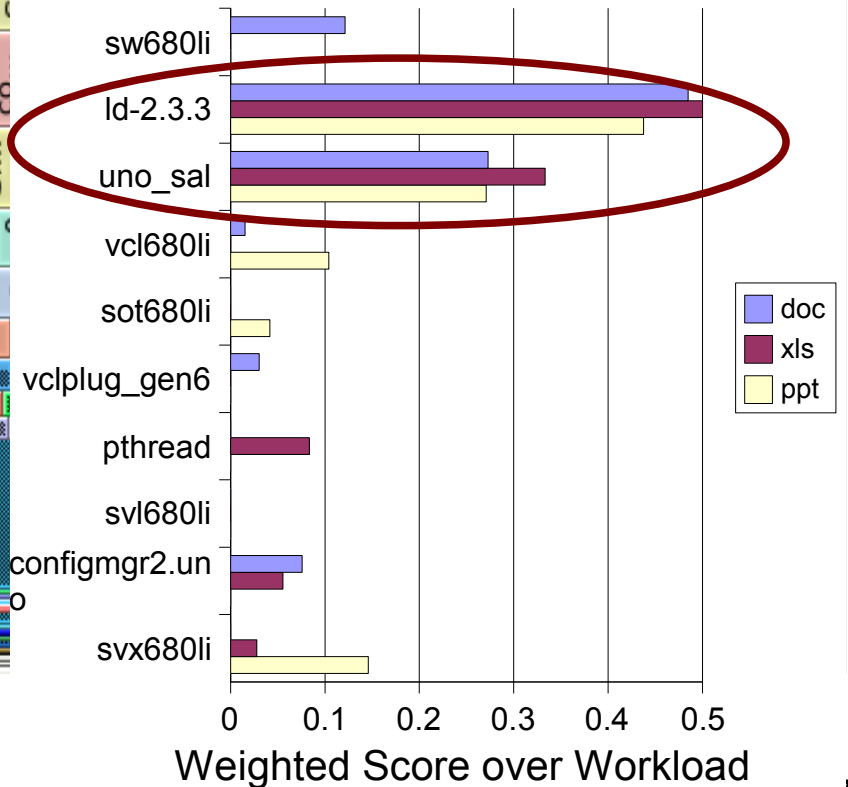
# Profiling

- Startup & load representative document set
- Initial Profiling Goals
  - *Get some first hand data, particularly with Intel tools*
  - *Get better understanding of the code & modules*
- Expectations:
  - *Code hotspots, functions that could be optimized*
  - *Call flow paths that could be tuned*
- Experiences:
  - *Code complexity – Many tools crash or confused*
  - *Call graphs did not help as expected.*
- What helped:
  - *Using multiple tools, knowing their strengths*
  - *Using the source, knowing UNO & interaction*
  - *Clues & context from helpful OpenOffice developers*

# Profiling



Import - top modules (time)





# Profiling Results

- Rework hot spot code
  - *ww8scan – quick patch, submitted.*
  - *Locked string operations – investigation*
- Load/Link
  - *Second thoughts – Symbol visibility, library count..*
  - *Prior & ongoing work, e.g. Michael Meeks*
- Better, more automated tooling for profiling
  - *APPR*
- Further investigate approaches for speed-up
  - *Configuration file operations & initialization*
  - *Disk access patterns*
  - *Smarter document load*





# Tooling



# APPR

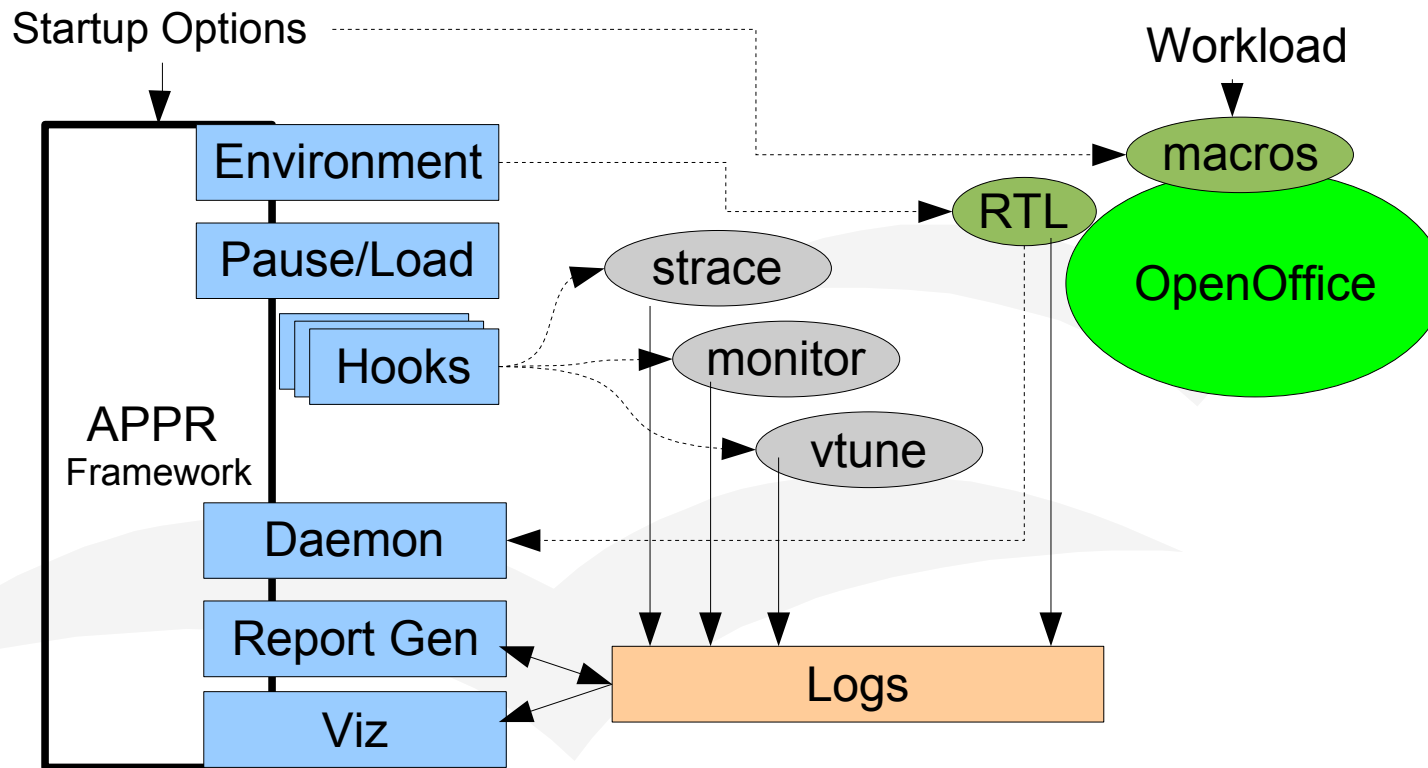
- Automated Profiling & Performance Regression
- Genesis:
  - *No one tool sufficient*
  - *No one document is representative*
  - *Manual profiling lengthy, tedious and error-prone*
- Goals:
  - *Gather & correlate measurable data of interest*
  - *Measure & track metrics across builds*
  - *... and across workload variation*
  - *Add metrics appropriate to work focus*
    - *e.g. Startup, File Load, Responsiveness...*
- Implementation begun early August
  - *yli34 (framework), mikeleib (correlation)*
  - *sh, StarBASIC, C, perl*



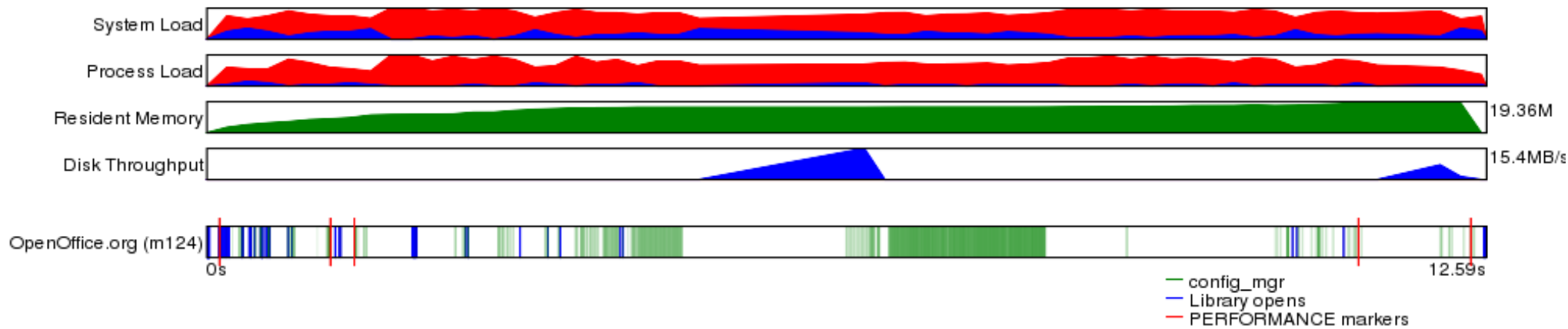
# APPR Features

- Benchmark a build
  - *Start-up time, phase timings, other correlated data*
  - *Empty shell/doc, load document or run workload*
  - *Compare with previous build*
- Correlated Data
  - *RTL log, strace, link data, system monitor*
  - *Vtune system sampling data*
- Visualization
  - *All data in one place, convenient to compare*
- Extensibility
  - *Add new metrics, new tools or change workload*
- Other uses, in addition to overall performance
  - *Feature tuning, Regression testing*

# APPR – How it works

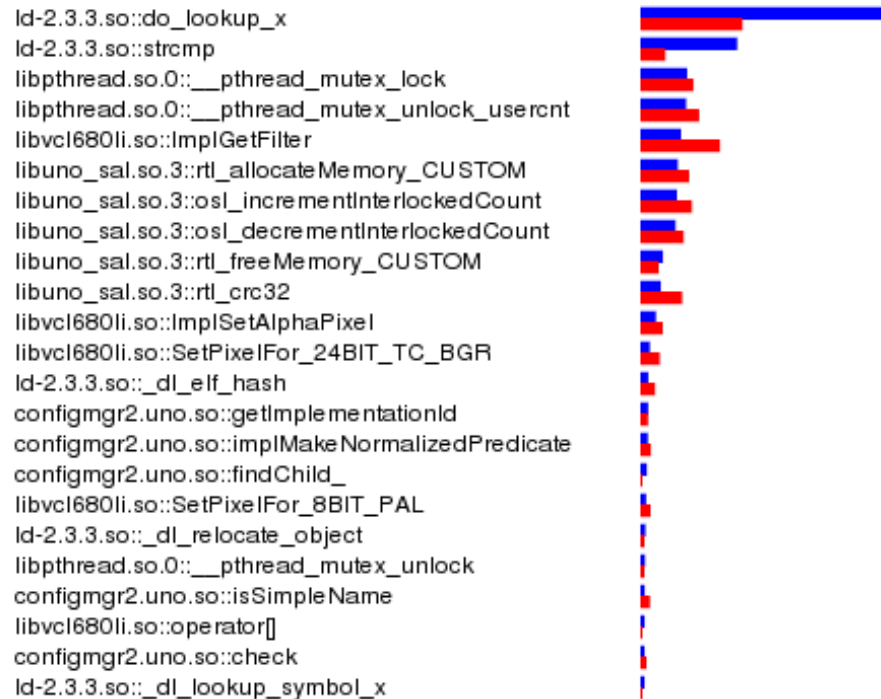


- Startup & synchronization
  - *Pause until all utilities loaded & ready*
- Monitoring & data gathering
  - *Trigger actions based on source markers*
- Correlation & visualization



| Runtime Linker Statistics              |        |
|--|--------|
| Total startup time in dynamic loader   | 0.107  |
| Time needed for relocation             | 0.107  |
| Number of relocations                  | 132875 |
| Number of relocations from cache       | 147391 |
| Number of relative relocations         | 22128  |
| Time needed to load objects            | 0.019  |
| Final number of relocations            | 132875 |
| Final number of relocations from cache | 147391 |

#### VTune Function Hotspots



APPR Visualized Example Output

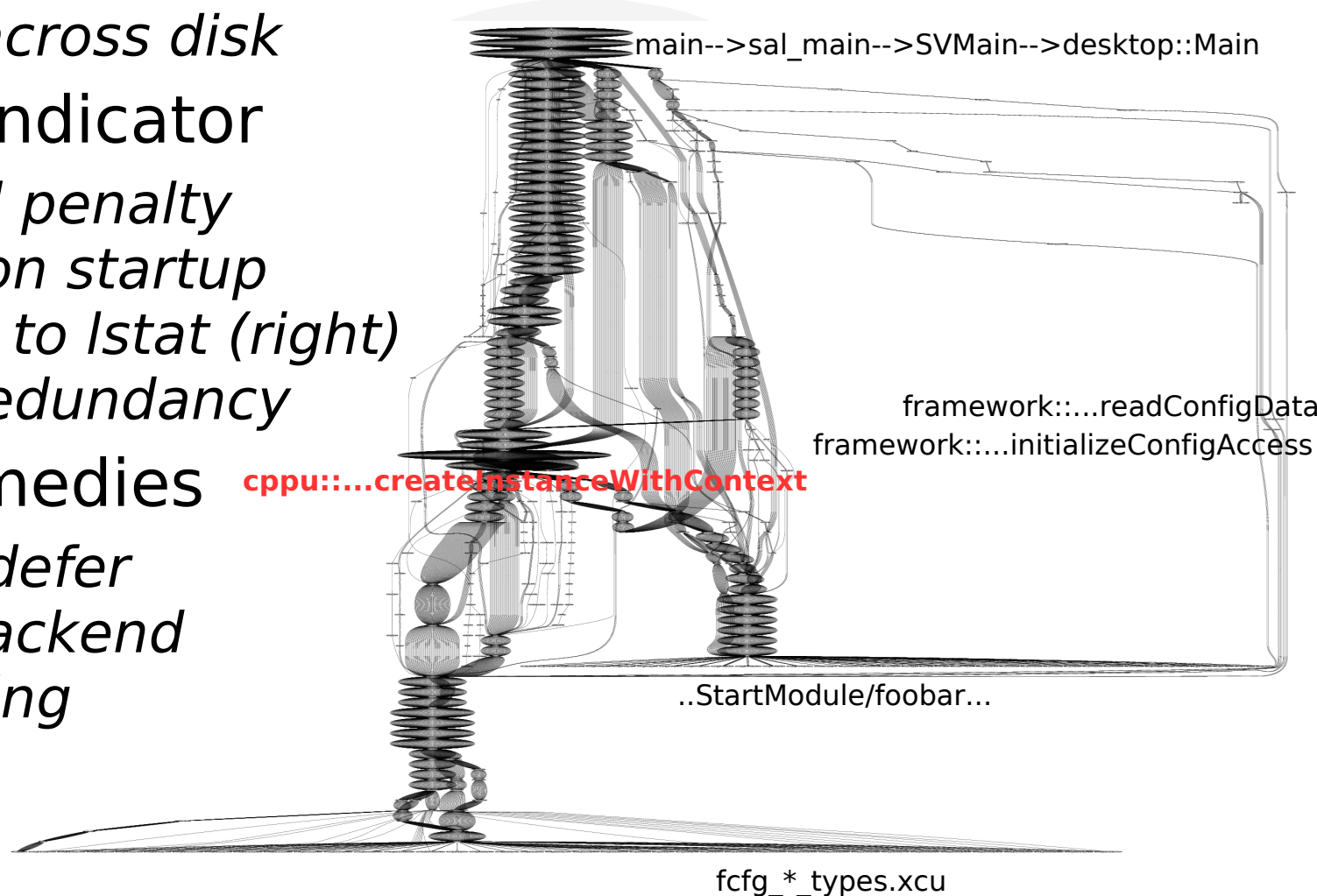


# Approaches under Investigation

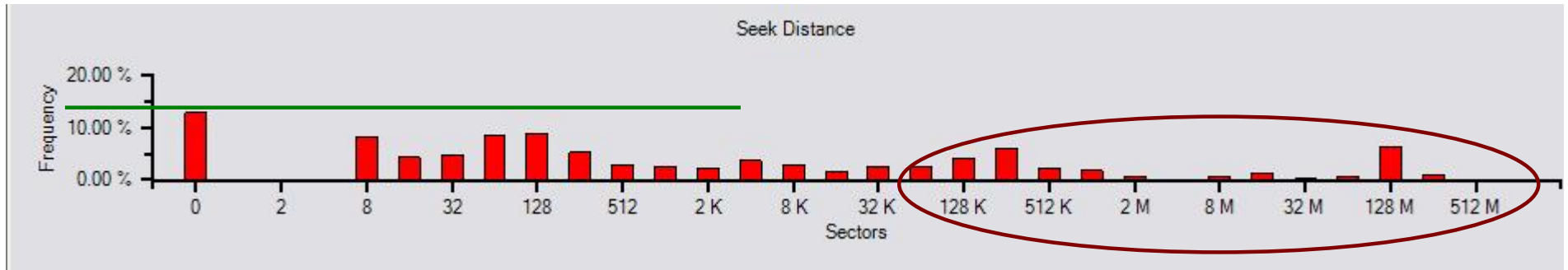
# Config Manager



- Profiling shows lot of config activity (green, above)
  - *Several hundred file reads & writes*
  - *Scattered across disk*
- Istat as an indicator
  - *System call penalty*
  - *487 times on startup*
  - *Multi-paths to Istat (right)*
  - *Apparent redundancy*
- Possible remedies
  - *Reduce or defer*
  - *Different backend*
  - *Initial caching*



# Disk Access Patterns



- Scattered disk accesses – seek penalty.
  - *Mix of code and configuration files at startup*
  - *“warm start” may access disk – config + page faults*
  - *Theoretical speedup: 1.5x to 3x*
- Goal: Identify & exploit access patterns
  - *Gather and analyze traces for workload + usage*
  - *Find deterministic sequences & map to file/offsets.*
  - *Change high level algorithms, e.g. config manager*
  - *Packaging changes*
  - *Disk block reordering utility*





# Threading & Rendering

- Threading can potentially improve startup.
  - *Real as well as perceived*
- Examples:
  - *Impress: Render 1<sup>st</sup> slide while file being read*
  - *Calc: Render visible sheet, background others*
- Goal:
  - *Start discussion on problems*
  - *Understand bottlenecks & work involved*
  - *Prototype specific threading based improvements*



# Conclusions

- Startup is one of the oft-heard user complaints
  - *OpenOffice 2.0 is faster than 1.1*
  - *But, still considered to be slow by many*
  - *Especially with document – empty or loaded*
- Potential for speedup, promising approaches
  - *Pre-linking, symbol visibility has helped*
- Call to Action:
  - *Discussion on ideas & gotchas*
  - *Give us your feedback & suggestions*
  - *Use APPR on code areas you care or know about*
  - *Better collaboration*