AP4Z™ **the** Application Profiler for Z

Introduction

AP4z labs

Why Application tuning on z/OS

The times, they are a-changin'

For a long time processor speed increase provided enough growth and helped hiding application problems, but:

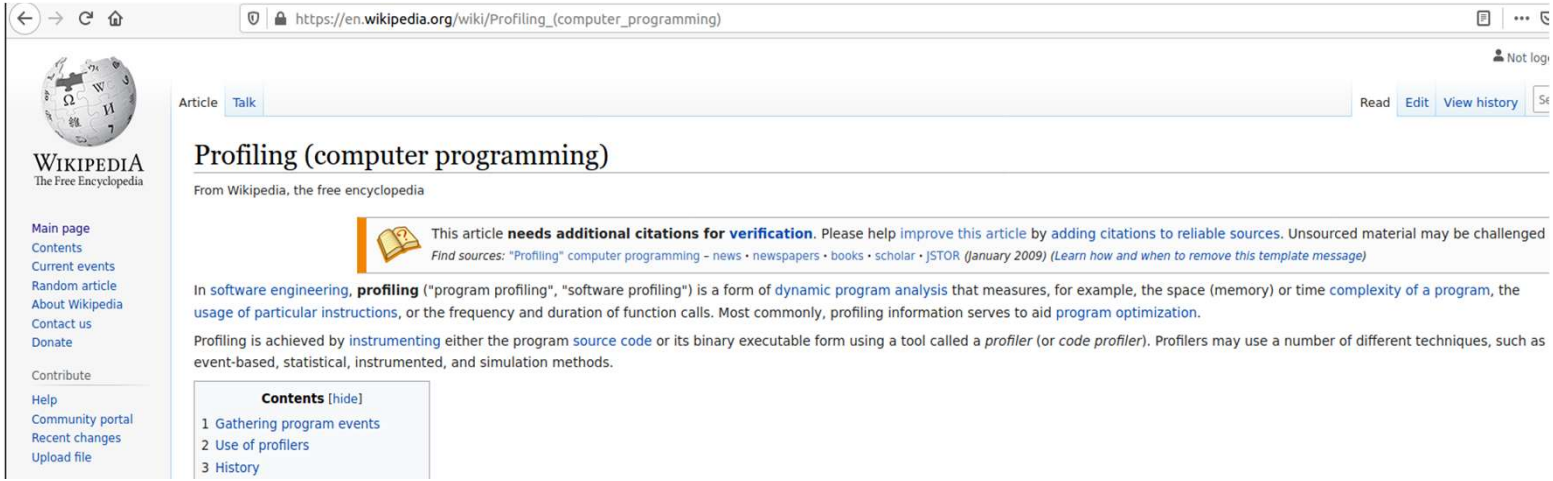
- Significant performance gains are now predominantly achieved through explicit use of new HW/SW functions.
- Historically the focus for performance tuning has been on infrastructures more than on applications, but:
- After so many years, opportunities for infrastructure optimization are lessening.

Why Application tuning on z/OS

Tuning existing applications is emerging as a very promising area for larger optimizations / saving, but:

- The required skills cross the boundaries between infrastructure and applications.
- Under z/OS there are no tools specifically designed to support such activity.
- This is different than for other platforms where application optimization through profiling is a common practice. See for example the JavaTM Platform Profiling Architecture specifications.

What do we mean by application profiling



A lightweight monitor that tracks the actual execution of entire application workloads,
measures noticeable activities like for example dynamic program calls,
identifies active modules, their attributes, inter-relationships, call frequency and duration.

Application Profiling and modernization

Application modernization may provide significant benefits. Examples of modernization projects include:

- Partial or full rewrite of existing applications using Java.
- Redeploy of individual functions or entire applications onto other platforms.

Application profiling supports modernization projects:

- Help assess the complexity by understanding the relationships between programs, applications.
- Detect and fix unexpected or non compliant relationships, enforce application boundaries.
- Reduce the scope creating lists of programs actually in use vs programs not used anymore.

Application Profiling in support of tuning projects

Application profiling supports tuning projects:

- Reducing the scope of the tuning effort spotting programs not used anymore.
- Showing the actual relationships between programs, applications.
- Identifying highly used and top consumer programs that if tuned can provide most benefit.
- Isolating inefficient application modules.

Identifying the most promising tuning actions profiling reduces cost and risk associated with optimization projects. It allows to better estimate potential savings, to achieve them sooner, to precisely measure them.

Application Profiling and day by day operations

Application profiling supports day by day application management to:

- Increase accuracy of Quality Assurance Test by assessing in advance what should be tested after a change.
- Make sure production runs pick the right module from the right library (regression avoidance).
- Detect potential security breaches: Programs which should not run, unexpected module version changes.
- Assess module level CPU usage trends, identify which programs changed their behaviour and when.
- Help speed-up Root Cause Analysis on performance slow-downs.

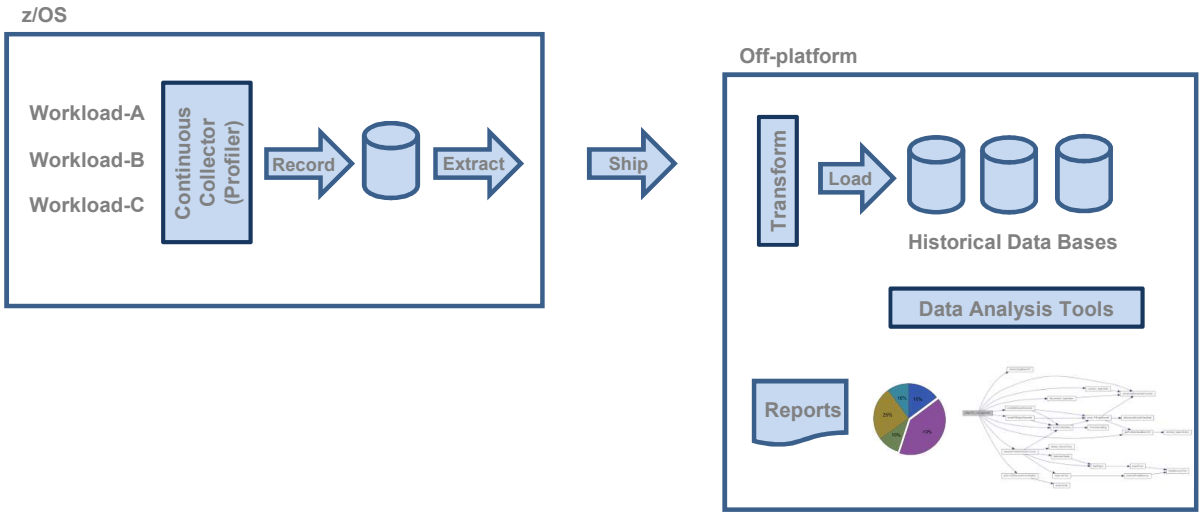
This reduces effort and cost associated with application management.

Attributes required to support the above activities

Supporting the above activities requires a tool:

- Specifically designed to profile applications' execution at scale.
- Aimed at providing a full picture about programs / subprograms and their actual relationships.
- Able to report Elapsed Time and CPU consumption at the individual (sub)program level.
- Automatically collecting each active module's compile date, used compiler release, and compiling options.
- Able to build a call graph showing who actually calls who.
- Simple, with no special system requirements, and with a negligible impact in terms of CPU consumption.
- Able to build a long term history of active programs their performance, their relationship to each other.

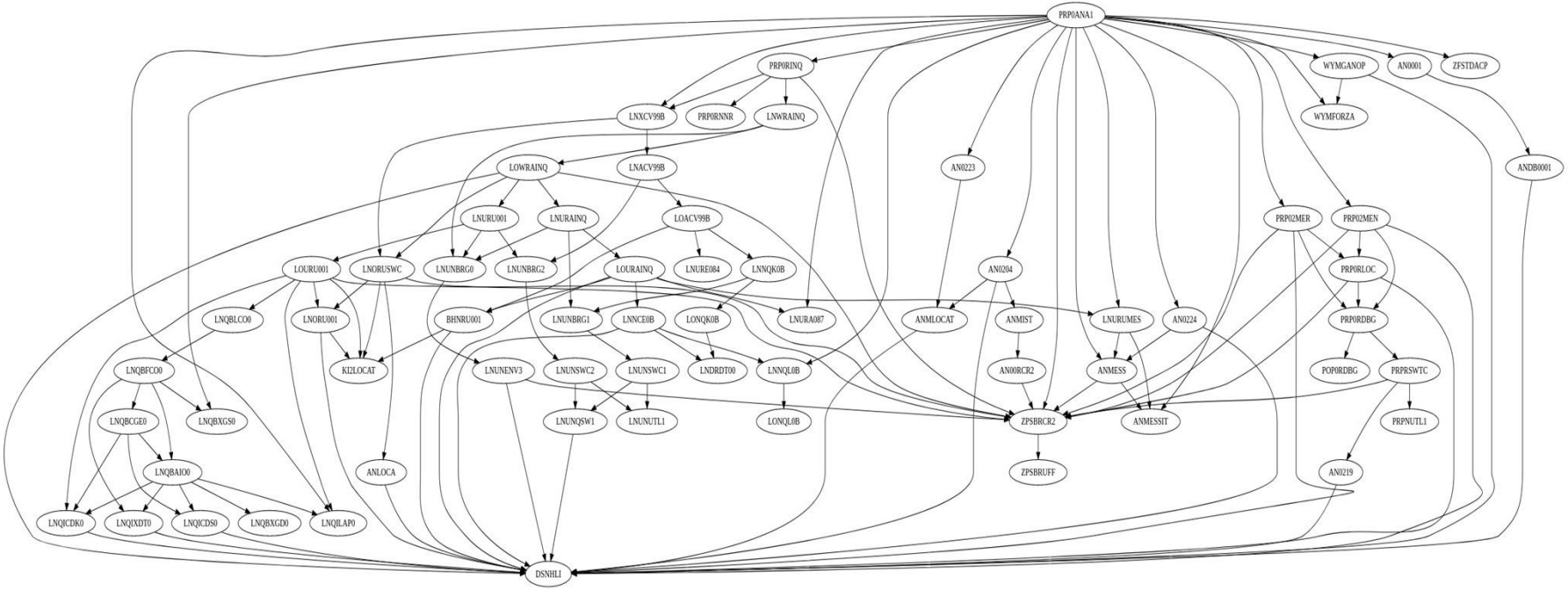
AP4Z Technical Architecture



AP4Z (in blue) is made of a z/OS part, the *data collection engine*,
and a distributed part, the *data analysis engine*

AP4Z – Language Overview

AP4Z – Application Call Tree



Thank you