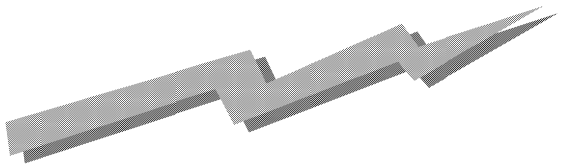
Tips, Hints, for Service

Warp 4.0 Performance

Robert Paulsen, Rich Wahl
Paulsen @ AUSNOTES
tie 678-4512



Presentation Outline

- **Overview**
  - Overall performance characteristics of Warp 4.0 as compared to Warp 3.0
- **Performance Tips**
  - Quick hints and tuning tips
- **Performance Problem Analysis**
  - How to analyze performance problems
- **Support for the Support Team**
  - Other organizations
  - Performance tools

Overview
Warp 4.0 Differences from 3.0

- **Overall Speed**
  - No noticeable difference from 3.0 when similarly configured.
- **Memory Requirements**
  - Bigger than Warp 3.0: 16 megabytes; 24 for speech.
  - Bigger PMShell (Open32), bigger data structures (256 color exploitation).
- **CPU Requirements**
  - More demanding than 3.0: 486DX 33 MHz, P90 (75) with VoiceType.
- **Disk Space**
  - Bigger than Warp 3.0: 200 megabytes, 225 with VoiceType.
- **Boot Time**
  - Longer than Warp 3.0. One test system (486 33 MHz, FAT):
    - Non-networked went from 55 to 103 seconds.
    - Networked went from 112 to 183 seconds.
- **Install Time**
  - About the same as, maybe somewhat longer than, 3.0: Approx. one hour.

Overview
Warp 4.0 Differences from 3.0 (cont.)

- **Networking**
  - No important performance changes.
- **Java**
  - No special hardware requirements.
  - Uses JIT compiler.
  - About as fast as other JIT implementations.
- **Speech**
  - Pentium 90, L2 Cache, 24 megabytes (75/16 possible);
  - Same performance as on the "other" operating system.

Overview
Warp 4.0 Differences from 3.0 (cont.)

- **FAT File System**
  - Asynchronous read-ahead algorithm tuned for faster processors.
  - Bigger hash table for cache.
  - Cache allocation defaults changed: 5% of system memory. 3.0 was 10%, but capped at 1.6MB. (This results in smaller cache than Warp 3.0 when less than 32MB, larger cache when above 32MB.)
  - Changes result in somewhat better performance on various file-intensive benchmarks.
- **MultiMedia/Graphics**
  - 32-bit GRADD tuned for 32-bit applications, implies some additional overhead for 16-bit apps.
  - Some graphics and PM benchmarks show approx 6% - 15% slowdown.
  - New MIDI APIs allow development of faster real-time MIDI apps on IDE systems.

Overview
Warp 4.0 Differences from 3.0 (cont.)

- **Memory Management**
  - System Loader has improved "fixup" scheme. Reduces working set. Helps counteract the overall increases due to new function.
  - Continued page tuning also helps.
- **Misc.**
  - Asynchronous Focus Change
    - "Single Input Queue" fix prevents applications from blocking focus change requests.
    - *OS/2 System -> System Setup -> System -> user interface*
  - 50 New printer Device Drivers (supported)
  - Device Driver Pak on CD-ROM (as-is)
  - Plug & Play -- automates installation process
  - DDC-2 (Display Data Channel) allows automatic setting of refresh rates on displays that support the standard -- reduces flicker.

Performance Tips

See Performance Tuning white paper by Duc Vianney.

- **Simplify**
  - Do not install features you are not going to use.
- **For constrained memory systems:**
  - Eliminate extraneous process (Warp Guide, Warp Center).
  - Remove unused device drivers from CONFIG.SYS.
- **FAT vs. HPFS**
  - Use FAT for small partitions, HPFS for large.
  - If FAT only, remove HPFS cache. ("IFS=...").
  - If HPFS only, default is no FAT cache. (Different from 3.0)
- **PATH, LIBPATH, DPATH**
  - Put in order of most frequent useage.

Performance Tips (cont.)

- **Speech**
  - Use sleep mode. "Go to sleep" and "Wake up please" are faster than closing and reopening. Uses minimum system resources.
  - Can use VoiceType settings notebook to have VoiceType start up in sleep mode.
- **DOS, Win-OS/2**
  - Allow installation process to "migrate" applications so settings tuned for each application will be used.
- **Desktop**
  - Use WarpCenter "trays" to hold and organize frequently used applications or objects. This is faster than "drilling down" through menus or folders.

Performance Problem Analysis

How should performance problems be dealt with?

Based on process documented by Tony White of PSCC.

1) Define the Problem

- **Be quantitative, not qualitative.**
  - What specific application or activity is slow?
  - How "slow" is it? Compared to what expectation or previous experience?
  - What kind of activity is it -- disk, computation, networking?
  - When does the problem occur? Always? At specific times? With some other activity?
- **Define the system and environment.**
  - System configuration (CPU, memory, etc.)
  - Type of connectivity (LAN, Token Ring, Ethernet)
  - Other application(s) loaded or in use
  - What recent changes have been made?
- **Collect Data! This will help focus understanding of the problem.**
  - Use tools like SPM/2 to determine CPU utilization, memory use.
  - What is the network activity?
  - How do Client and Server resource utilization compare?

Performance Problem Analysis

How should performance problems be dealt with? (cont.)

2) Hypothesize Solutions

- **Based on above analysis, brainstorm possible solutions.**
  - Solutions will generally involve relieving bottlenecks
- **Perhaps a bottleneck can be relieved by adding more or faster resources.**
  - more memory, a faster processor, faster disk, faster file system, etc..
- **Perhaps there tuning parameters available to tailor the system or application to the task.**
  - Cache sizes, swapper location, etc..
  - see the PERFORMANCE TUNING white paper by Duc Vianney (soon to available on a web site).
- **Perhaps the application needs a redesign.**
- **Perhaps the system topology needs to be changed.**
  - Physical changes: how many servers, controllers, etc.
  - Logical changes: balancing workloads between servers

Performance Problem Analysis

How should performance problems be dealt with? (cont.)

3) Design Tests

- **Prioritize the tests.**
  - Most likely to least likely to show results
- **Focus the tests.**
  - Only change one thing at a time.
- **Document the tests.**
  - What feature or resource does each test exercise?
  - What does each test intend to show?
  - What are the expected results?

Performance Problem Analysis

How should performance problems be dealt with? (cont.)

4) Run the Tests

- **Document the test setup.**
  - Exactly what system, configuration, parameters, etc., were used?
- **Measure and document results.**
  - Be quantitative.
- **Be repeatable.**
  - Non-repeatable results will only lead to confusion.
  - Always start with the same setup, except for the test's intended variations.
  - Again, change only one thing at a time.

Performance Problem Analysis

How should performance problems be dealt with? (cont.)

5) Analyze Results of All Tests

(If all the above is done properly, this step is easy.)

- **Draw conclusions.**
- **Make recommendations.**
- **Implement changes.**
- **Re-measure.**
- **Document final results.**
  - Puting it in writing helps you to clarify your own thinking and promotes completeness and correctness.

Support for the Support Team -- Other Organizations

- **Personal Systems Competency Center**
  - Provides fee-based, on-site, service.
  - Contact: Go through marketing rep, who can contact Mary Ann Mayers at T/L 522-5819.
- **SWAT Team**
  - Handles "Crit Sit".
  - Contact: Lindell R. Small, manager, at T/L 523-3643.
- **Performance Departments**
  - Internals Performance department: Concentrates on development process.
  - Externals Performance department: Concentrates on customers, industry analysts, marketing, etc.. Supports PSCC and SWAT Team.
  - Contact: Reach via PSCC or Swat team.

Support for the Support Team -- Performance Tools

- **CPU Monitor Plus -- BonAmi Software**
  - Commercial Product
- **PM Patrol -- WallyWare Software**
  - Shareware: http://homepage.interaccess.com/~wally
- **SPM/2 -- IBM Product**
  - Theseus2 -- OS2TOOLS disk (internal use only)
- **Other internal tools (e.g. Dekko, TPROF)**
  - Some require special hardware, use new APIs.
  - Available to SWAT Team, PSCC, when Performance Departments are involved.
- **Note: Warp Center's System Activity Monitor uses new Performance API.**
  - Less intrusive than Pulse
  - More accurate than Pulse