Technical Publication Costs Cut in Half with Laser Printing

Steve Wilk & Sam Boles

Hewlett-Packard

Synopsis

Technical publications enjoy all the standard maladies of other publications: the high cost of type-setting, the rising cost of printing, the long turn-around time from camera-ready copy to multiple copies ready for distribution. But technical publications tend to be unique -- at least in degree -- in their volatility and the urgency of timeliness.

Hewlett-Packard, by being in the computer business, is by definition in the technical publication business. An HP team of development engineers and performance engineers have implemented a methodology using HP3000 document processing facilities and the HP2680 laser printer to cut time costs and dollar costs in half for internal-grade technical publications.

The Problem

Where It All Begins

This story begins in the Performance Center of the Hewlett-Packard computer facility in Cupertino, California. That's where we work as part of the Systems Performance Team running customer benchmarks and doing the performance characterization of many of Hewlett-Packard's hardware and software products.

Big-Ticket Benchmarks . . .

As any of you who've done a benchmark know -- perhaps too well -- they're *expensive.* A complex benchmark emulating a hundred-terminal interactive load across a range of job mixes and configurations can cost *tens of thousands* of dollars -- even *hundreds of thousands.*

. . . and What They Buy

Thru such benchmarks our Systems Performance Team generates some very valuable intelligence on a variety of HP3000 configurations as well as performance optimization techniques.

This intelligence has a flavor to it that is uniquely *real-world*. The benchmarks are run with *customer data* against *customer files* that often are taken directly off a *customer's production* system.

This performance intelligence is also *leading edge.* The benchmarks are often run on newly-released or pre-release products. The environment is such that we can experiment with configurations to calibrate resource sensitivity: changing, running, measuring time after time, validating or rejecting our hypotheses, each time understanding the subtle interplay of system concurrencies a little better - or at least appreciating the *limits of our understanding* a little better.

The environment also enables us to deploy *specialists* on special issues. If it's an ATP timing delay issue or a VIEW screen download issue, we can get the lab expert - perhaps in the next building - to bring to the problem the level of expertise unique to the *author* of the code or the *designer* of the board.

A Central Clearing-House

The Performance Center then is the forum for the early encounters of our products and your world. It has the hardware and software resources, the human resources and the instrumentation and metrics to derive valuable intelligence from these encounters.

So What's The Problem?

The problem is *disseminating* that valuable intelligence. *Sharing* with the HP support organization around the world, and, thru them, the customer base around the world. *Leveraging* the heavy investment in performance characterization of HP hardware and software products. Basically, *spreading the word.*

The Birth of *pn* ^{2}

Last year we decided to establish a formal mechanism to communicate our performance information to the field. It's name ended up as *Performance News Notes* -- or *pn* ^{2}, as we affectionately nick-named it.

It's a monthly publication, 10 to 20 pages in length (for 1-shot reading). It's current circulation is 2500 readers in the HP organization around the world.

In our original specifications we set out some objectives for the publication:

Quality: It didn't need to be slick and in color, but it did need to be readable and professional in appearance.

Cost: We always aim at zero and compromise to the extent that quality and other necessities dictate.

Speed: It had to be fast. Especially with new product characterization we couldn't tolerate a multiple-day multiple-iteration typesetting cycle, and a 4-day to 2-week printing cycle. But our cost objectives wouldn't allow the premium to expedite these cycles. Also we wanted to use the HP bulk mail system to minimize national and international distribution costs, so labeling and sorting speed were another consideration.

Flexibility: It had to have graphics capability since much of our information is best communicated in graph form. It needed multiple character sets for more intelligible typography.

The Solution -- Set-up Phase

The Laser Printer

We decided to try the *Laser Printer*. It looked like a reasonably good fit for our quality, speed, cost and flexibility objectives. We had an HP2680A on one of the systems in our department, so we gave it a go.

Getting Started with IDS/3000

We spent a few hours getting set up. We used IDSCHAR ("IDS" means Interactive Design System) software and a 2647 graphics terminal to design our logo. That took a couple of hours, counting the tweaking that you're always tempted to do even though *you're the only one* who'll ever notice that one dot out of the 180 per inch that's a little bit out of line.

Then we moved into forms design with IDSFORM software and the 2647. We needed a title page and a mailing label page. This took about 2 hours, plus another hour or 2 of moving things around till everybody -- or almost everybody -- was reasonably happy.

IFS/3000 for Device Specifics

IDSCHAR and IDSFORM generate dot matrices and vectors that are device-independent. This data is then compiled by IFS2680 ("IFS" means Interactive Formatting System), into an ENVIRONMENT file that is compatible with a specific I/O device, eg the HP2680 laser printer. So we used IFS2680 to compile our ENVIRONMENT specifications into an ENVIRONMENT file that the HP3000 could then download to the laser printer.

The ENVIRONMENT file loads character sets (up to 32 different character sets at a time) and forms (up to 32 different forms at a time) into the memory (we have a megabyte of main on our HP2680) that's used by the processor in the laser printer to control the laser so you get an 8-point Helvetica Bold instead of a 10-point Roman Italic. Building the ENVIRONMENT file took another hour or two till we were satisfied.

In all we probably spent about a day getting set up. Some of this time was *learning curve* and some was that "one last touch" of the *amateur artist*, but most was the time necessary to design the form or logo and enter the points and other specs.

The real advantage was the fact that we could do it all ourselves without the start-stop iterations, queues and communications entropy of a graphics department. The initial set up was done in *one continuous uninterrupted block* of time. There were no meetings or key resources to schedule. There was no "now let me see where I was two days ago when I last worked on this" re-think time.

(Of course, the only time you can get that big a block of uninterrupted time is on weekends, so the one-day set up time didn't cost the company anything.)

Typesetting and Composition

We chose TDP (Text and Document Processor) for this because we wanted 2-column capabilities (to enable speed readers to bounce only vertically) and right justification of proportional character sets (to get more balanced "Linotyping"). Also, we needed the capability of including charts from DSG (Decision Support Graphics) integrated with the text.

We included several character sets in our ENVIRONMENT file:

Helvetica Bold: 8-, 14-, 24-point
Helvetica Italic: 8-point
Roman Bold: 10-point
Roman Italic: 10-point
Line Printer: 8-point
Math: 12-point
Inverse Line Printer: 12-point

We specified the appropriate FONTID's to TDP to couple with the ENVIRONMENT file, specified margins, columns, page length, and the like and proceeded to do our typesetting and layout.

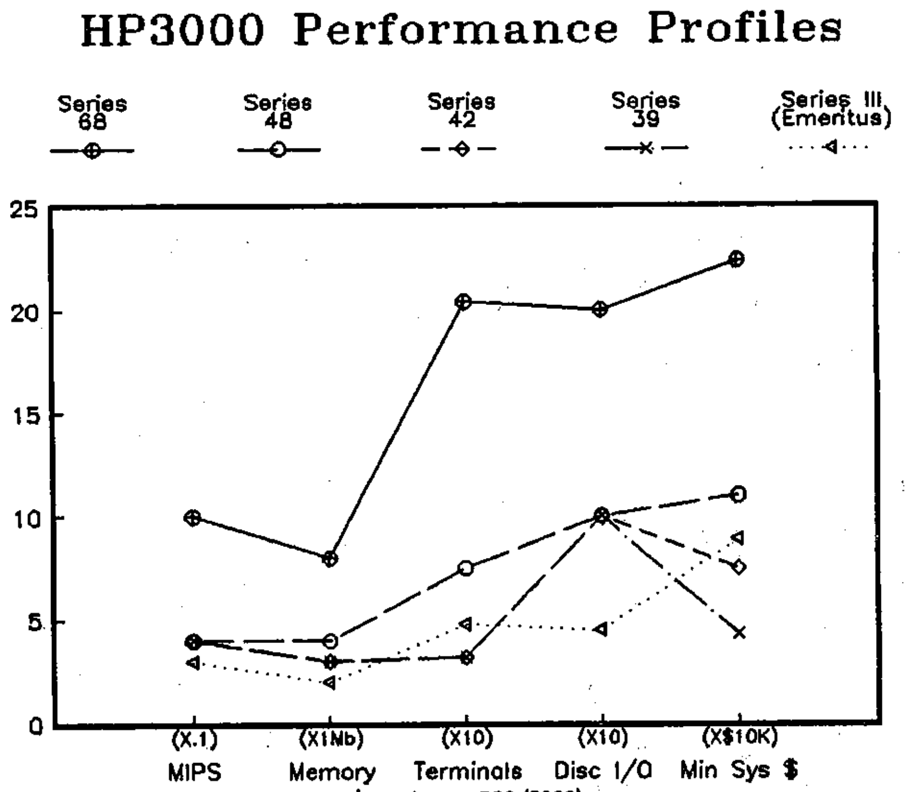
For the layout of charts (bar/line/pie) from DSG and illustrations from HPDRAW, we found that we were doing a lot of trial and error placement, so we set up a template like this:

```
/* ILLUSTRATE rasterfile
/NAME RASTER rasterfile
/ILLUSTRATE figurefile:figure #lines
```

We would NAME the raster file each time we changed the size or the figure (having explicitly purged the old raster file as needed) and therefore had to do another vector-to-raster conversion via the ILLUSTRATE command. Then for the next pass we'd comment out the NAME and second ILLUSTRATE, and de-comment the first ILLUSTRATE so we pulled in the raster form without having to do a vector-to-raster conversion when not needed:

```
/ILLUSTRATE rasterfile
/* NAME RASTER rasterfile
/* ILLUSTRATE figurefile:figure
      #lines
```

This facilitates getting charts like the following integrated with the text:



(graphics by DSG/3000)

This template technique we extended to the issue level, so for the subsequent issues the engineer who had responsibility for that issue could "cookbook" from the archive file of the previous issue without re-inventing the TDP commands. This saves time and gives consistency to the appearance of the publication.

We ran through a number of iterations on our first issue, balancing artwork, policing "widows" (TDP can do a lot of this automatically), cleaning up typo's and again falling into the "one last touch" syndrome of the amateur artist. Once again we found a significant *productivity surge in cutting out the middleman* -- this time the typesetter. We could do a lot of the local typesetting on the fly as we wrote the articles at the terminal. The global typesetting and layout commands were largely templated (about 75%) from the previous issue. The cut-and-paste we did by computer. And the inevitable last-minute "stop-the-presses" newsflash could be accommodated with only a little pain in a matter of minutes rather than days via the typesetter's queue.

Then finally, there we were with our first issue, camera-ready. We had done the logo and forms design, the typesetting, the graphics and the layout. We were ready for the printer.

We went to our in-house offset service. They were booked solid for the next week and a half. That was 10 days. We couldn't wait that long. If we paid the expedite premium we could get it by the end of next week. That was 5 work days. Better, but still *not good enough.*

What to do?

It was then that someone said, "Why not do it on the laser?"

"The whole thing?" we said.

"Why not? It prints 45 pages a minute."

The next day was Saturday. There's only sporadic use of the laser printer weekends (especially since we had finished our prototype the previous weekend). We needed 1300 copies at 12 pages each. 12 * 1300 = 15,600 pages. 15,600/45 -> 347 mins -> 6 hours.

It looked do-able ... were we in for an unpleasant surprise.

The Solution -- Manufacturing Phase

We decided to give it a go. That Saturday morning, we set OUTFENCE 10 to prevent interleaving by random print-outs and to enable us to drain the queue as needed but in a controlled fashion. We did a HEADOFF 14 to save paper and reduce the splitting work. We did a TDP FINALQ of our text file to suppress the TDP message at the end. We set COPIES=120 on the SPOOL file to get multiple copies. And we were off and running.

But not for long.

We found that after the first copy, the whole SPOOL file (including the heavy-duty ENVIRONMENT portion) was being down-loaded. Because our ENVIRONMENT file had so much in it, the laser printer was going into a warm-up cycle between each copy. (To conserve energy, the infra-red fusing drum cuts off if not in use for a few seconds, then it has to warm up to the fusing temperature before the printing resumes.)

This cut our print rate to 10 pages a minute. Our 6 hour print job now looked like 25 -- if everything went well. And Mr. Murphy, the silent partner in all such pioneering ventures, said that was not likely to happen. Besides, we were getting an extra page from TDP and that meant every other copy had to be refolded to get the cover on the front. (We used an even number of pages to prevent this but it wasn't working.)

SPOOK to the Rescue

We were just about to hang it up and wait the 10 days for the "real" printer to do the job when someone said, "Why not use the SPOOK APPEND command, build a new SPOOL file with one copy of the ENVIRONMENT and multiple copies of the text. We can even cut out the extra page in the process."

We tried it and it worked. The print speed was up to 35-40 pages a minute -- not far from the 45 ppm rating of the HP2680. We decided to give it another try.

But before we got started, someone said, "How about those name-address labels we were going to print separately on the gummed stock -- why don't we just splice them in while we do the SPOOK APPEND to solve the performance problem?"

Labels on the Fly

We had already tried the TDP MAILER facility to do this. The MAILER is good for multiple addressees for text that does not have a heavy FINAL formatting load, since its formatting performance is about O(n). With 12 pages of 2 column right-justified proportional text plus 8 graphics inserts, we were taking several minutes for a FINAL even with the graphics already in raster format. An O(n) performance would give us 1300 FINAL's at several minutes each. This was not operationally feasible, but a SPOOK APPEND splice might be.

We tried it. The redundant APPEND's (you're APPENDing a text set for each addressee and an ENVIRONMENT set for each 10-15 addressees) builds at a slower (about 2x) rate than your printing rate. If you have enough temporary disc space, you can fire up the APPEND's Friday night, start your printing Saturday morning and have a fairly level work pressure for the operator who's doing the splitting, QA and boxing for the distribution center. And this does save the time for the label stick-on step.

It wasn't elegant, but the prototype was functionally and performance-wise feasible, so we launched it.

Short on Finesse, but it Worked

Well, with that prototype methodology, we managed to brute-force the first issue in about 18 hours. (The 18 hours includes some cockpit errors I'd rather not talk about. Let's just say that engineers are not necessarily the best operators.) Not exactly the 6 hours we had dreamt of -- but we had beat our alternatives by a week or more and had established that we had the technology and it was just a Small Matter Of Methodology (remember "SMOP" of Large Systems Games fame: Small Matter Of Programming . . . well, this is "SMOM") that needed to be developed.

By the third issue (if you want to find out how short a month is, commit to getting out a monthly publication), we got a bit of prototype software and STREAM commands in place to make our brute force methodology a little more tolerable.

Later, we contracted a 17-year-old student who's working his way thru college doing laser typesetting and printing. Off-loading the grunt work to the student has freed up engineering resources that can be used to get the methodology to an operationally sound condition.

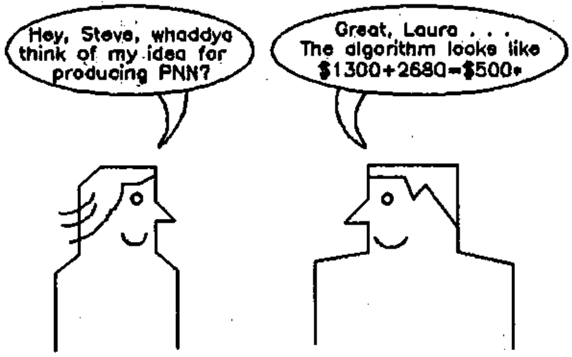
The Results

Structured Development

The approach we've used in evaluating laser technology for typesetting, composition and low/medium volume printing is an extension of Structurism. We're all aware of Structured Programming and Structured Design. This is Structured Development or Prototyping.

We did the first issue with engineers doing the operations, throwing together ad hoc software as necessary to establish that the technology is capable of producing the results. That's established now.

In the next step of development we prototyped some software in BASIC and PASCAL to automate some of the more onerous clerical procedures. The idea here was that if a viable methodology for production were not attainable, we would throw away less code of less value.


Hey, Steve, whaddya think of my idea for producing PNN?
Great, Laura . . . The algorithm looks like $1300+2680=$500.

(graphics by HPDRAW)

With the crude prototype in place for more than 6 issues, we've seen some interesting results:

12 page performance news notes
2 column right justified proportional
7 fonts, 8 pieces of artwork
(bar and line charts, illustrations)

7:00 PM Friday: last text edit
6:00 AM Monday: 1700 copies printed,
pre-sort labeled, split
Variable cost: $500

Variable cost includes labor, paper, toner, carrier, drum cost and maintenance. The machine is idle otherwise, so the sunk costs of printer depreciation, real estate etc. are excluded.

As word spread on the results of laser typesetting, layout and printing, we applied the methodology to other publications. Here's a recent scenario that may have set a (world-class?) record:

Friday at 3:00 PM, a marketing manager asks if it's possible to have 1000 copies of a 20 page (5 character fonts, 2 column, 1 piece of artwork, no labels) strategy document by Monday (3 calendar days away).

Friday night: Logo design done, forms design started, preliminary environment file compiled.

Saturday: Sample of cover page and first text page built, reviewed, approved. Text data not yet available as ASCII file.

Sunday 11:30 AM: Text data ready for typesetting to start.

Sunday 4:00 PM: Last edits made to text file.

Monday 3:00 AM: Last of 1000 copies printed using 3 laser printers (extras idle on graveyard shift Sunday night.)

We normally call 5:00 PM Friday till 8:00 AM Monday 0 business days. Technically, 11:00 AM Sunday till 3:00 AM Monday is perhaps a turn-around of -1.5 days. In our experience, negative turn-around time is not all that common.

Futures

Our evaluation of the prototype is positive. We've saved at least $5,000 on *pn* alone, which annualizes at over $10,000.

We are currently evaluating a phase I production version that is operationally sound and simple and may have a 30-50% performance improvement (over the current label splicing methodology.) This would be non-supported contributed library class software.

This same facility may be expandable to include a serial number of each page for sensitive "Do not copy" documents. The control file for this may be the name-address interface file enhanced. This would be non-supported contributed library class software.

Long-term, we see satellite communication to field offices of the "spool file" equivalent and the generation of hard copy as needed at the field location.

Conclusion

The fact that we are continuing to use the prototype version of the laser typesetting/printing methodology is evidence of its economic and functional feasibility. The fact that we are intending to invest the engineering to productionalize the methodology reflects that it has potential for material contribution in the near term.

About the Authors....

Steve Wilk *is the Performance Center Manager at the Hewlett-Packard computer facility in Cupertino, California, with responsibility for performance benchmarks and product characterization. His 18 years in the industry cover a wide range, including programming, systems analysis and program management. With an undergraduate degree in mathematics, Steve received his MBA from the University of San Francisco.*

Sam Boles *is a Systems Engineer in the Systems Performance Center at the Hewlett-Packard computer facility in Cupertino, California. With HP for seven years, Sam's computer experience started back in the AUTOCODER days of the 1401/1410, migrated thru the 360/370 era, and now focuses on HP hardware and software performance characterization. Sam earned his MS at UCLA in Information Systems.*

sebiug12 1015 15Nov83