

# IT Performance

Martin Klier

Klug GmbH integrierte Systeme, Teunz

24.03.2009

## 1. Vorstellung

Martin Klier, 29

Datenbank- und Systemadministrator

Fachliche Schwerpunkte: Systemperformance / Tuning, hochverfügbare Systeme, Cluster und Replikation

Web: <http://www.usn-it.de>

## 2. Allgemein + Begriff

Unter „Performance“ versteht man im IT-Bereich im Allgemeinen die Leistungsfähigkeit eines Systems. Dieser Vortrag soll veranschaulichen, daß Systemperformance von vielen verschiedenen Faktoren abhängt, und wie einzelne davon bewertet werden können.

## 3. Prozessoren (CPU)

Vorstellung von Eigenschaften und diversen Fakten

## 4. Flüchtiger Speicher (RAM)

Einführung in Charakteristika und Eckwerte als Grundlage für den weiteren Ablauf

## 5. CPU vs. RAM

Vorstellung verschiedener Technologien für den Datenfluss zwischen Recheneinheit und Arbeitsspeicher. Hauptsächlich bezogen auf aktuelle x86- und x86\_64-Architekturen.

## 6. Skalierungsgrenzen

Praktische Darstellung eines theoretischen Faktums: Die Skalierbarkeit von Problemen und Lösungen ist begrenzt!

## 7. Massenspeicher (HDD/SSD)

Überblick über die Technik, die Stärken und die Schwächen von Festplatten auf Magnetbasis und Solid State Disks mit Halbleiter / MOSFET-Speicher.

## 8. RAID

Fortgeschrittene Überlegung zur Kombination von Massenspeichern, Abwägung gängiger Verfahren unter dem Geschwindigkeitsaspekt in verschiedenen Situationen.

## 9. Rechnerverbindungen

Betrachtung herkömmlicher Netzwerktechnik und aktueller Low-Latency-Verfahren unter dem Aspekt einer Scale-Out-Lösung.

## 10. Zusammenfassung

# The Speed-Up-Loop

aus: The Daily WTF

“So what do you think about the opportunity,” Ben’s recruiting agent asked. He thought about it for a few moments. It wasn’t exactly what he was looking for, but then again, he had been out of work since November of 1989 – nearly three whole months – and figured he should probably get back in to the swing of things. He told the recruiter that he’d like to talk to the client and asked to schedule an interview for the following week.

“Actually,” the recruiter replied, “they need someone A-S-A-P. Can you go in any sooner? As in, later today?”

Fortunately, Ben had not only showered that day, but was clean-shaven. He agreed to the same-day interview and, two short hours later, Ben arrived at their offices to meet with Wayne, the lead developer.

“Thanks for coming in so soon,” Wayne started off, “the previous contractor just left all of a sudden. No matter, I see that you were at Initech for a few months, what did you do there?”

“I was on the data interch—”

Wayne cut him off. “Okay, so have you done worked with inline assembly before?”

“Yeah, actually on the last proj—”

Wayne cut him off again. “How familiar are you with multi-task programming?”

As Ben launched into the typical, early 1990’s multi-task programming spiel, he noticed that Wayne had shifted his attention to his notepad. Making occasional “uh huh” affirmations, Wayne started scribbling a few notes.

“Sounds good,” Wayne interrupted yet again. “Now take a look at this code. Tell me what’s wrong with it.” Wayne tossed the notepad over with the following hand-written C-code.

```
int i;
char *p = 0x10000;
for (i = 0; i < 10000000;i++)
{
    *p++ = 0;
}
```

Ben mentioned the obvious integer overflow, and then added “but anyway, I’d just use memset.”

“Great,” Wayne said, “you’ve got the job! Can you start now?”

“I guess,” Ben said, a bit confused. It was 3:03PM – exactly three minutes after starting the interview – and Ben didn’t have any other plans for that afternoon. “Let’s, uh, get started?”

“Perfect!” Wayne said enthusiastically. “Okay, so first and foremost, this will be the easiest job you’ve ever done. Really, we don’t need any extra help, but since another contractor just left, we had to fill in the void. You know how that goes, right?”

Wane continued, pausing only for a brief moment. “I tend to follow the 80/20 rule here. You’ve heard of that before?”

Wayne didn’t give Ben a chance to respond. “Nothing – and I mean nothing – in IT takes less than 80 hours, and whatever you think it’ll actually take, multiply it by 20, and tell management that. You see, 80/20.”

That wasn't quite the 80/20 rule that Ben was familiar with, but he figured, to each his own. After going over a few other "rules" of the programming department, Wayne went on to explain what they needed Ben to work on.

It was a fairly-boring logistics application that ran in the company's home-built MTWS, or Multi-Tasking Windowing System. Although Microsoft's Windows 2.11 had been on the market for a little while, no one at the company seemed to trust it, which is why they built their own.

Wayne was the designer and primary coder of MTWS. It was a DOS-based Window manager that used ANSI graphics to draw "windows". Each window could house one of four different applications (one of which Ben would be responsible for maintaining), and actually allowed dragging, resizing, and inter-window communications.

After giving a brief tour of the MTWS and the logistics application, Ben left for the day and arrived early the next morning to get started. The next few days were rather uneventful. As were the next few weeks. And the next few months.

Under Wayne's direction, Ben would spend about an hour each day doing maintenance or other small changes to his assigned application, and the other seven hours avoiding maintenance or, really, any other type of work.

With all of his downtime and no web to surf, Ben spent his time digging through the innards of MTWS. It was actually well-commented, well-structured C code that had a dash of assembly here and there. One day, while perusing the graphics subsystem, he noticed a little loop buried in the code...

```
for(i=0;i<1000000;i++) {}
```

It appeared that the loop would run whenever the screen was updated. He double checked. It did, in fact, run on every screen update.

Figuring that he found a bug or a vestigial piece of code, Ben asked Wayne what he should do.

"Ha," Wayne chuckled. "That, my friend, is what we call a 'speed-up loop.' We put those in for insurance purposes, really."

Ben shook his head, trying to figure out how he missed that it was yet another work-avoidance technique.

"The idea is," Wayne continued, "whenever we have those really slow weeks – you know, the kind where don't actually fix any bugs or make any other changes – we just drop one of the zeros on the loop. And then we just tell the manager that we ran into some speed issues with the latest change request, but after a whole lot of deep-juju optimization, we were able to speed things up significantly and should be able to do the change request the next week."

"See," Wane said after pausing for a moment, "it's insurance."