

# Tools zur Entwicklung paralleler Programme

*Isabel Loebich*

**Für die optimale Nutzung von Parallelrechnern müssen bestehende Programme, die auf einem Prozessor laufen, erst so umgeschrieben werden, daß sie mehrere Prozessoren gleichzeitig beschäftigen können. Für die Fehlersuche in einem parallelen Programm reicht ein normaler Debugger nicht aus. Schließlich braucht man eine Möglichkeit, den Erfolg der Parallelisierung zu bewerten. Der Artikel gibt eine Zusammenstellung der am RUS verfügbaren Tools, die den Programmentwickler bei den verschiedenen Schritten des Entwicklungsprozesses für parallele Software unterstützen.**

Ausgehend von vorhandenen sequentiellen Programmen können folgende Schritte unterschieden werden:

- Analyse des bestehenden Programms
- Parallelisierung
- Übersetzen des parallelen Codes
- Fehlersuche
- Optimierung der Parallelisierung.

Das RUS bietet zu den verschiedenen Schritten Tools an, die im folgenden kurz vorgestellt werden. Dabei wird nur auf die Tools eingegangen, die plattform-unabhängig an-geboten werden; daneben gibt es natürlich auf jedem Rechner noch spezielle Tools der jeweiligen Rechner-Hersteller.

## Analyse von Fortran-Programmen

Für die Analyse von sequentiellen Fortran-Programmen steht der FORGExplorer der Firma Applied Parallel Research (APR) zur Verfügung. Dieses Tool bietet verschiedene Möglichkeiten der Analyse:

- Datenflußanalyse über Aufrufparameter, Common-Blöcke und Equivalence-Anweisungen
- Verfolgung von Referenzen auf Variable durch den ganzen Aufrufbaum
- Daten- und Kontrollflußanalyse: Übersicht über:
  - Wertzuweisung und Benutzung von Variablen
  - Unterprogrammaufrufe, DO-, IF- und GOTO-Kontrollblöcke
- Übersicht über Common-Block-Verwendung
- Suche nach Variablen nach Kontext-Kriterien
- Konsistenzcheck (Unterprogrammaufrufe, Common-Blöcke)
- Instrumentierung des Programms zur Zeitmessung in Unterprogrammen/DO-Loops
- Neuformatierung des Programms nach wählbaren Kriterien.

FORGExplorer fungiert als Basis für das im folgenden vorgestellte Parallelisierungstool, ist aber auch ohne Parallelisierungsabsicht zum Kennenlernen eines Programms nützlich. FORGExplorer steht für alle gängigen Workstation-Plattformen als Shareware zur Verfügung und kann auch von unserem ftp-Server geholt werden:

<ftp://ftp.uni-stuttgart.de/pub/info/producer/apri>

## Parallelisierung von Fortran-Programmen

Aufbauend auf FORGExplorer stellt APR sowohl für Distributed Memory- als auch für Shared

Memory-Architekturen Parallelisierungstools zur Verfügung. Beide Tools sind einerseits als Batch-Programme verfügbar (`xhpf` bzw. `spf`) und ermöglichen damit eine automatische Parallelisierung, andererseits ist eine interaktive Version (FORGE/DMP bzw. FORGE/SMP) in die Benutzeroberfläche von FORGExplorer integriert, die die Parallelisierung von Hand unterstützt:

- Parallelisierung:
  - für Shared Memory-Architekturen
  - für Distributed Memory-Architekturen:  
Verteilung von Feldern und Parallelisierung von DO-Loops
- Generierung von parallelisiertem Code:
  - High Performance Fortran (HPF)
  - Message Passing Calls (eigene Runtime-Library)
  - Compiler-Direktiven für das jeweilige Zielsystem (Shared Memory-Systeme)
- Laufzeitstatistik:
  - CPU-Nutzung, Kommunikationskosten und Wartezeiten für Unterprogramme und DO-Loops
  - Laufzeitsimulation.

Für diese Tools hat das RUS derzeit Lizenzen für zehn Benutzer, die Runtime Library für FORGE/DMP mit Message Passing ist derzeit für bis zu 16 Prozessoren auf der intel Paragon installiert.

## HPF Compiler

Liegt bereits ein Programm in High Performance Fortran (HPF), als Ergebnis einer Parallelisierung mit FORGE/DMP oder als direkt in HPF geschriebenes neuerstelltes Programm, vor, kann (plattform-unabhängig) `pghpf` von der Firma The Portland Group (PGI) als Compiler benutzt werden. Dieser Compiler setzt zuerst das HPF-Programm in ein Fortran-77-Programm mit Message Passing-Aufrufen um. Anschließend werden die normalen Schritte des Fortran-77-Compilers zum Übersetzen und Linken durchlaufen. Wahlweise lassen sich die Zwischenergebnisse, also beispielsweise der Fortran-77 Code mit den Message Passing-Aufrufen, erzeugen. Das Laufzeitverhalten kann über Optionen oder Variablen kontrolliert werden.

`pghpf` steht für verschiedene Workstation-Plattformen zur Verfügung, außerdem u.a. für intel Paragon, CRAY T3E, Silicon Graphics und IBM SP2. Das RUS hat Lizenzen für SGI- und IBM-Workstations, bei intel Paragon und CRAY T3E wird ein auf `pghpf` basierender HPF-Compiler vom Rechnerhersteller mitgeliefert. Die derzeitige Version auf der intel Paragon ist allerdings veraltet; wir bemühen uns um einen Update, eventuell in Form eines Crosscompilers.

## Fehlersuche

TotalView ist ein Debugger mit graphischem Benutzerinterface, mit dem

- Programme mit mehreren Prozessen
- Programme auf anderen Rechnern im Netz
- verteilte Programme mit mehreren Prozessen auf diversen Rechnern im Netz

untersucht werden können. TotalView wird als plattform-unabhängiges Standard De-bugging Tool für Fortran-, C- und C++-Programme auf den Plattformen des RUS angeboten. Für Workstations stehen Lizenzen für 25 Benutzer zur Verfügung, eine Lizenz für 15 Benutzer auf der NEC SX-4 ist geplant. Auf CRAY T3E wird TotalView vom Hersteller mitgeliefert.

## Performance-Analyse

Die nachträgliche Analyse eines Rechenlaufs kann für Programme, die MPI für die Parallelisierung verwenden, mit VAMPIR (Visualization and Analysis of MPI Resources) erfolgen. Dazu muß zunächst auf dem Parallelrechner ein Tracefile produziert werden. Hierfür kann einerseits VAMPIRtrace benutzt werden, da diese Tracefiles direkt von VAMPIR gelesen werden können. VAMPIRtrace erfaßt ohne Änderungen am Code nur MPI-Aufrufe. Für Tracefiles in anderen Formaten stehen Konverter zur Verfügung (z.B. für ParaGraph oder TraceGen Tracefiles). Die Analyse des Tracefiles erfolgt dann auf einer Workstation.

Ein weiteres Performance Analyse-Tool ist ParaGraph. Hierfür muß ebenfalls zuerst ein Tracefile erzeugt werden: Auf der intel Paragon mit dem Debugger `ipd`, auf der NEC SX-4 für MPI/SX-Programme durch Linken einer zusätzlichen Library. Die Analyse des Tracefiles kann für auf der Paragon erzeugte Tracefiles auf einer Workstation erfolgen. Die Tracefiles auf der NEC SX-4 haben ein anderes Format und können nur mit PG auf der NEC SX-4 ausgewertet werden.

Mit beiden Tools können verschiedene Aspekte des Laufzeitverhaltens analysiert werden, die dynamisch als graphische Simulation des Programmablaufs oder aufsummiert in Diagrammen dargestellt werden können. VAMPIR bietet dabei detailliertere Informationen bis hin zu den Namen der Routinen und eine bequeme Bedienung mit der Maus. Bei ParaGraph stehen einige zusätzliche Aspekte wie z.B. eine übersichtliche Darstellung der Lastverteilung auf die Prozessoren zur Verfügung.

VAMPIR gibt es für alle wichtigen Workstation-Plattformen, VAMPIRtrace beispielsweise für die NEC SX-4, CRAY T3E, intel Paragon, Silicon Graphics. ParaGraph ist am Rechenzentrum der Universität Stuttgart auf intel Paragon und NEC SX-4 und auf verschiedenen Workstations installiert.

## Bereitstellung am RUS

Soweit sie auf Workstations zur Verfügung stehen, wurden alle genannten Tools, in `/sw` bereitgestellt, d.h. sie sind unter `/client/bin` aufrufbar; die man-Pages sind unter `/client/man` zu finden; die Handbücher in `/client/doc` in Form von PS-Files.

Details sind in der folgenden Tabelle zusammengefaßt:

Tool	Aufruf	Architekturen <sup>a</sup>	Lizenzen
FORGExplorer	<code>forgex</code>	sgi, aix, hp, dec, sparc	10 User
FORGEX/DMP	<code>forgex</code>	sgi, aix, hp, dec, sparc	10 User
FORGEX/SMP	<code>forgex</code>	sgi, aix, hp, dec, sparc	10 User
xhpf	<code>xhpf</code>	sgi, aix, hp, dec, sparc	10 User
spf	<code>spf</code>	sgi, aix, hp, dec, sparc	10 User
HPF Compiler	<code>pghpf</code>	sgi, aix	10 User
TotalView	<code>totalview</code>	aix, dec, sparc	25 User
ParaGraph	<code>paragraph</code>	sgi, sparc	-
VAMPIR	<code>vampir</code>	sgi, aix, hp, dec, sparc	Site

**a:** sgi: Silicon Graphics IRIX 5 und 6, aix: IBM RS6000, dec: DEC Alpha OSF/1, hp: HP/UX 9, sparc: SUN Solaris

Auf den Parallelrechnern direkt sind momentan folgende Tools verfügbar:

<b>Tool</b>	<b>Aufruf bzw. Benutzung</b>	<b>Architekturen<sup>a</sup></b>	<b>Lizenzen</b>
FORGE/DMP Runtime Library	USE_FORGE setzen Linken mit: -L\$FORGEDMPlib -lmpf bzw. -L\$FORGEDMPlib -ltpf Dokumentation: \$FORGEdocu/TECHNOTE/	paragon	16 Knoten
HPF Compiler	pghpf	t3e, paragon	-
TotalView	USE_PROG_ENV setzen totalview	t3e	-
VAMPIRtrace	USE_VAMPIR setzen Linken mit: -L\$VAMPIRpath -lVT bzw. -L\$VAMPIRpath -lVTnull Dokumentation: \$VAMPIRdocu/userguide.ps	sx4, paragon	Single Platform
Erzeugen von ParaGraph- Tracefiles	Linken mit: -lmpit bzw. -lmpitw Dokumentation: Online-Manual MPI/SX User's Guide (Appendix B) in <a href="http://www.hlr.de/nec-doku/">http://www.hlr.de/nec-doku/</a>	sx4	-
Erzeugen von ParaGraph- Tracefiles	in ipd mit dem Kommando: instrument -paragraph Dokumentation: /usr/share/ps.docs/ptug.ps	paragon	-
ParaGraph	PG	sx4	-
ParaGraph	paragraph	paragon	-

a: sx4: NEC SX-4, t3e: CRAY T3E, paragon: intel Paragon

Isabel Loebich, NA-5991

E-Mail: [loebich@rus.uni-stuttgart.de](mailto:loebich@rus.uni-stuttgart.de)