Schnelleinstieg Xpress/Mosel für längere Rechenexperimente

für die Rechner des Instituts für Numerische und Angewandte Mathematik

Universität Göttingen

Robert Schieweck

Stand 10. März 2014

Inhaltsverzeichnis

1 Einleitung

1.1. Was hier erklärt wird Dieses Tutorial erklärt, wie man unter Ubuntu die Xpress Mosel per Konsole verwendet, um längere Rechenexperimente durchzuführen. Insbesondere wird erklärt

- wie man Xpress im batch-Betrieb verwenden kann, also viele verschieden Rechnungen nacheinander durchführen kann, ohne vorm Rechner sitzen zu müssen
- wie man sich per `ssh` auf einem anderen Rechner einloggt, um Xpress dort zu verwenden, um den eigenen Rechner nicht mit den Rechnungen zu belasten
- wie man Xpress weiterlaufen lassen kann, nachdem man sich ausgeloggt hat
- wie man mit der mosel-Programmiersprache Xpress steuern kann
- wie man relevante Kenngrößen der Rechenergebnisse mit mosel auslesen kann

1.2. Was hier nicht erklärt wird Dies ist keine Einführung in die mosel-Programmiersprache. Grundlegende Kenntnisse werden vorausgesetzt. Es werden nur die Dinge erklärt, die für die Durchführung von automatisierten, längeren Rechnungen nötig sind. Die Benutzung unter Windows wird hier auch nicht erklärt, aber das meiste sollte mit ein wenig Internetrecherche übertragbar sein.

2 Auf einem anderen Computer rechnen lassen

Da längere Rechenexperimente viel Systemressourcen verbrauchen, ist es meistens sinnvoll, sie auf einem anderen Rechner als dem eigenen Arbeitsplatzgerät durchzuführen, damit man selber weiterarbeiten kann bzw. rechenstärkere Server verwenden kann.

2.1. Per `ssh` einloggen Nachdem man den eigenen Arbeitsplatzrechner hochgefahren hat, öffnet man eine Konsole und tippt

```
>ssh c4.num.math.uni-goettingen.de
```

ein, um sich beispielsweise auf dem Server `c4.num.math.uni-goettingen.de` einzuloggen. Mit dem Befehl `logout` schließt man die `ssh`-Verdindung wieder.

2.2. Hardwareausstattung des Rechners überprüfen Da schnellere Computer für kürzere Laufzeiten sorgen, sollte man für wissenschaftlich verwertbare Rechenexperimente immer die zugrunde liegende Hardwarekonfiguration mit angeben. Diese erfährt man mit den Konsolenbefehlen `lscpu` und `cat /proc/cpuinfo`.

2.3. Auslastung des Servers überprüfen Da man für wissenschaftlich verwertbare Rechenexperimente nur auf Computern rechnen sollte, auf denen sonst keine rechenintensiven Prozesse laufen, sollte man sich vor Beginn der Experimente mit dem Konsolenbefehl `top` die aktuelle Auslastung des Servers (auf dem ja potenziell auch andere rechnen) anzeigen lassen. Durch

drücken der Taste Q schließt man die Anzeige wieder. CPU-Auslastungen von über 100 Prozent bedeuten, dass mehrere Kerne für einen bestimmten Prozess arbeiten. Beispielsweise beansprucht ein Prozess mit 500 Prozent CPU-Auslastung 5 Kerne mit voller Last.

3 Xpress zum Rechnen bringen

3.1. Den Xpress Optimizer starten Mit dem Konsolenbefehl `mosel` wird die Xpress Mosel Konsole gestartet. Hier kann man mit Xpress-internen Kommandozeilenbefehlen arbeiten. Diese findet man zum Beispiel im **Xpress-Mosel Language Reference Manual** im *Abschnitt 1.3 Running Mosel*. Das entsprechende PDF findet man leicht per Suchmaschine im Internet.

3.2. Xpress beim Starten schon Befehle geben (batch-Modus) Will man viele Rechnungen auf einmal laufen lassen, geht das besser, wenn man nicht jede Rechnung per Hand in der Mosel-Konsole startet, sondern Xpress beim Start sagt, welche Rechnungen er nacheinander durchführen soll. Das kann man mit der Option `-c` des Befehls `mosel` erreichen.

Folgender Befehl startet Xpress und sorgt dafür, dass automatisch die Datei `beispiel.mos` ausgeführt wird. Danach schließt sich Xpress wieder automatisch:

```
>mosel -c 'exec"beispiel.mos"'
```

Falls man in der Datei `beispiel.mos` Parameter definiert hat, kann man diese hier auch direkt übergeben:

```
>mosel -c 'exec"beispiel.mos"FILE="..\\data\\daten.txt"""'
```

Dieser Befehl sorgt dafür, dass dem Parameter `FILE`, der in `beispiel.mos` definiert sein muss, der Wert `..\\data\\daten.txt` übergeben wird. Das ist also die Datei `daten.txt`, die relativ zu `beispiel.mos` einen Ordner höher und dann im Ordner `data` liegt.

Mehrere Parameter kann man so übergeben:

```
>mosel -c 'exec"beispiel.mos"FILE="..\\data\\daten.txt",x=0.01"'
```

Man beachte die einfachen Anführungszeichen `'`, in die der `-c`-Befehl eingeschlossen ist und die doppelten Anführungszeichen `"`, in die jeweils das Argument von `exec` und die Parameter eingeschlossen sind.

Mehrere Kommandos, die dann hintereinander von Xpress ausgeführt werden, werden mit Semikolon `;` getrennt übergeben:

```
>mosel -c 'exec"beispiel1.mos"FILE="..\\data\\daten1.txt""";
           exec"beispiel2.mos"FILE="..\\data\\daten2.txt"""'
```

Der Zeilenumbruch hier dient nur der besseren Lesbarkeit!

Es gibt auch andere Kommandos als `exec`, die man Xpress mit der Option `-c` übergeben kann. Eine Übersicht findet man zum Beispiel im **Xpress-Mosel Language Reference Manual** im *Abschnitt 1.3 Running Mosel*. Das entsprechende PDF findet man leicht per Suchmaschine im Internet.

3.3. Nach dem Ausloggen weiterrechnen lassen Wenn man längere Rechnungen laufen lassen will ist es häufig unpraktisch, während der gesamten Rechnung eingeloggt zu sein. Mit Hilfe des `nohup`-Befehls (steht für *no hang up*) kann man dafür sorgen, dass ein Befehl weiterausgeführt wird, auch wenn man sich ausloggt. So kann man z. B. Rechnungen über Nacht laufen lassen. Die Syntax ist folgendermaßen:

```
>nohup [Befehl] > [Output] &
```

Für `[Befehl]` kann man dann einen der obigen `mosel`-Befehle verwenden. Für `[Output]` gibt man eine Datei an, in die alles, was sonst während der Ausführung von `[Befehl]` in der Konsole angezeigt wird, in Textform gespeichert wird, so dass man sich nach Beendigung der Rechnung die Ergebnisse anschauen kann. Das `&`-Zeichen sorgt dafür, dass man die Konsole nach der Annahme des `nohup`-Befehls wieder Befehle annimmt und man sich so beispielsweise ausloggen kann. Folgendermaßen könnte also eine typische Verwendung von `nohup` aussehen:

```
>nohup mosel -c 'exec"bsp1.mos";exec"bsp2.mos""' > ausgabe.txt &
>logout
```

3.4. Rechnungen unterbrechen Falls eine Rechnung mal länger als gedacht dauert oder sogar abstürzt, kann man sie auf zwei Arten unterbrechen.

Wenn man kein `nohup` benutzt hat und die Konsole daher die Fortschritte der Rechnung anzeigt, kann man einfach Strg + C in der Konsole drücken.

Hat man `nohup` benutzt, wird der Fortschritt der Rechnung nicht in der Konsole angezeigt und daher kann man obige Lösung nicht verwenden. In diesem Fall lässt man sich mit `top` die process ID oder `PID` des Rechenprozesses anzeigen (z.B. `13242`) und kann diesen dann mit

```
>kill 13242
```

beenden.

Falls man in die aktuelle Konsole keine Befehle eintippen kann, weil man das `&` vergessen hat, kann man auch einfach eine weitere Konsole öffnen und die Rechnung von dort mit `kill` abbrechen.

4 Xpress steuern und Infos zur Rechnung erhalten

4.1. Xpress steuern Xpress besitzt einige sogenannte *control parameters*, die einem erlauben, Einfluss darauf zu nehmen, wie Xpress die Rechnungen durchführt.

Die control parameters können in `.mos`-Dateien mit den Funktionen `setparam(`*`Name`*`,`*`Wert`*`)` und `getparam(`*`Name`*`)` gesetzt bzw. gelesen werden. *`Name`* ist dabei ein String, der den Namen des control parameters angibt und *`Wert`* ein Wert, der diesem zugewiesen werden soll. Der Wert kann verschiedene Typen haben (String, integer, real, ...). Typische Anweisungen in einer `.mos`-Datei sehen so aus:

```
setparam('XPRS_maxtime', 300)
setparam('XPRS_verbose', true);
```

Die erste Zeile setzt ein Zeitlimit von 300 Sekunden zum Lösen und die zweite weist Xpress an während des Lösens regelmäßig einen Zwischenbericht über den Lösungsfortschritt anzuzeigen. Das ist bei komplexeren Problem hilfreich, kann aber auch zu extrem viel Textausgabe führen.

Eine vollständige Liste der *control parameters* mit Beschreibungen findet man im **Xpress-Optimizer Reference Manual** in *Kapitel 9: Control Parameters*. Das entsprechende PDF kann man per Suchmaschine im Internet finden. Achtung: Um die control parameters in einer `.mos`-Datei zu nutzen, muss dem Namen `XPRS_` vorangestellt werden wie im obigen Beispiel.

Alternativ kann man sich innerhalb der Mosel-Konsole eine Liste aller control parameters und problem attributes (hier leider keine Unterscheidung und keine Beschreibung, aber die Namen sind teilweise selbsterklärend) anzeigen lassen:

```
>mosel
FICO Xpress Mosel 64-bit v3.4.0
(c) Copyright Fair Isaac Corporation 2001-2012. All rights reserved
>examine mmxprs
```

4.2. Infos über das gelöste Problem erhalten Die sogenannten *problem attributes* dienen dazu, nach/während dem Lösen weitere Informationen über das Problem zu erhalten. Sie werden wie die control parameters mit `getparam` ausgelesen. Machmal werden auch die *problem attributes* als *control parameters* bezeichnet, was ein bisschen verwirrend ist, da sie nichts kontrollieren, sondern nur Auskunft geben. Typische Verwedungen:

```
x := getparam('XPRS_bestbound')
n := getparam('XPRS_nodes')
```

Die erste Zeile setzt `x` auf die beste duale Schranke an den Zielfunktionswert nach dem Lösen. Im Idealfall sollte sie dem besten gefundenen Zielfunktionswert entsprechen, aber möglicherweise waren ja die 300 Sekunden zu kurz, um eine Optimallösung zu finden. Die zweite Zeile setzt `n` auf die Anzahl der Knoten im Branch&Bound-Tree, die bisher gelöst wurden.

Eine vollständige Liste der *probem attributes* mit Beschreibungen findet man im **Xpress-Optimizer Reference Manual** in *Kapitel 10: Problem Attributes*. Das entsprechende PDF kann man per Suchmaschine im Internet finden. Achtung: Um die problem attributes in einer `.mos`-Datei zu nutzen, muss dem Namen `XPRS_` vorangestellt werden wie im obigen Beispiel.

Wie bei den control parameters kann man sich auch die problem attributes in der Mosel-Konsole anzeigen lassen.

5 Offizielle Dokumentation

Offzielle Anleitungen der Firma FICO findet man unter

`http://www.fico.com/en/Products/DMTools/xpress-overview/Pages/Xpress-Documentation.aspx`

Dafür ist möglicherweise eine Registrierung oder Xpress-Lizenz nötig. Man findet aber viele Anleitungen auch so per Suchmaschine, siehe vorige Abschnitte.