

# ARM DDT Debugger

2020

Trusted partner for your Digital Journey

© Atos - For internal use

**Atos**

# ARM DDT

---

- ▶ Parallel debugger (based on gdb)
- ▶ Cross-platform for multiple server and HPC architectures
- ▶ Native parallel debugging of Python applications
- ▶ Has market leading memory debugging
- ▶ Outstanding C++ debugging support
- ▶ Complete Fortran debugging support
- ▶ Has an offline mode for debugging non-interactively
- ▶ Handles and visualizes huge data sets

1

Installation

# Installation

## Download the remote client

---

- ▶ DDT is the parallel debugger part of ARM Forge.
- ▶ Installations of Forge on the cluster: **module avail forge**
- ▶ Use of the DDT Debugger is easiest through the remote client
  - GUI runs on the local desktop or laptop → responsive
  - Clients available for Windows, Linux and Mac OS/X
  - <https://developer.arm.com/tools-and-software/server-and-hpc/downloads/arm-forge>
  - You need the same minor version as the Forge installation on the cluster!
  - Currently version 20.1
  - No administrator access required

# Installation

## Load the private key in your SSH Agent

---

- ▶ SSH Agent:
  - Windows: Pageant
  - Linux & Mac OS/X: `ssh-add ~/.ssh/id_hlrn`

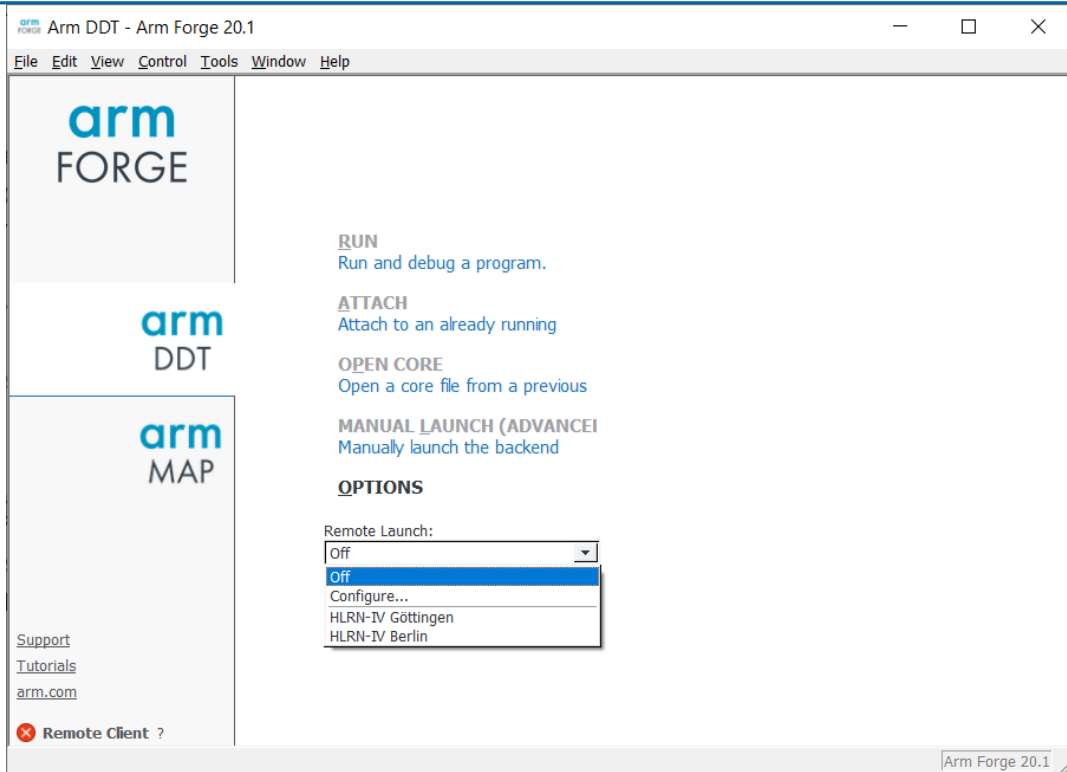
2

Setup

# Setup

## Configure remote launch

- ▶ Click ,Configure...`
- ▶ Then click ,Add`



# Setup

## Settings for `blogin.hlrn.de`

---

- ▶ Connection name: HLRN-IV Berlin
- ▶ Host name: [zzatodonn@blogin.hlrn.de](mailto:zzatodonn@blogin.hlrn.de)
- ▶ Remote installation directory: `/sw/tools/allinea/forge-20.1.3`



# Setup

## Settings for `glogin.hlrn.de`

---

- ▶ Connection name: HLRN-IV Göttingen
- ▶ Host name: [zzatodonn@glogin.hlrn.de](mailto:zzatodonn@glogin.hlrn.de)
- ▶ Remote installation directory: `/sw/tools/allinea/forge-20.1.3`

3

Starting to debug an  
application

---

# Starting to debug an application

## Loading your private key

---

- ▶ On Windows: start Pageant
- ▶ Add key to your login to the SSH Agent
  - ssh-add in Linux or Mac OS/X
  - ‚Add keys‘ in Pageant
- ▶ Make sure that you logged in before on the system through Putty or SSH and accepted the fingerprint.

```
The authenticity of host 'blogin.hlrn.de (130.73.234.7)' can't be established
ECDSA key fingerprint is SHA256:pNGlm//LyjJZi6tX0mz5SPSs4IBkuyJI/iWI10JbhgE.
ECDSA key fingerprint is MD5:e5:51:66:4e:b9:aa:78:0e:9b:5c:41:4a:2e:58:da:f3.
Are you sure you want to continue connecting (yes/no)?
```

# Starting to debug an application

## Connecting your remote client to the cluster

---

- ▶ Start DDT remote client
- ▶ Select ‚HLRN-IV Berlin‘ or ‚HLRN-IV Göttingen‘ under Remote launch
- ▶ ARM DDT is connected to the cluster and waits for connection requests.

# Starting to debug an application

## Modifying your job script

---

- ▶ Load the forge software in your job script: `module load forge/20.1.3`
- ▶ Prefix your `srun` or `mpirun/mpiexec` command with `ddt --connect:`

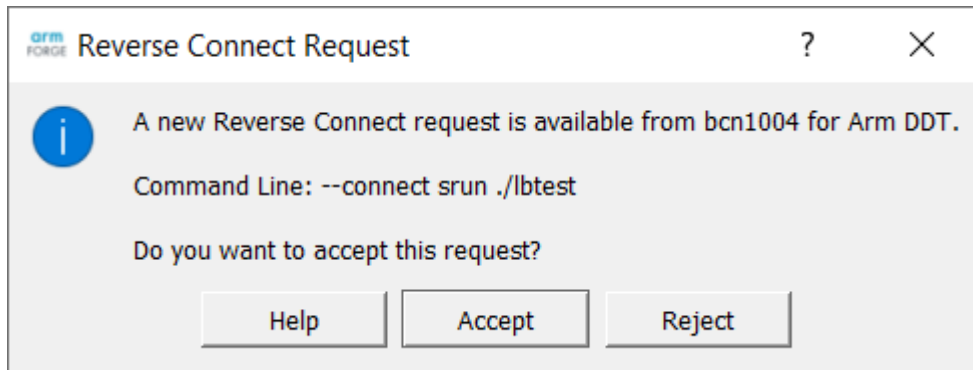
```
ddt -connect srun app
```

# Starting to debug an application

## Starting your job

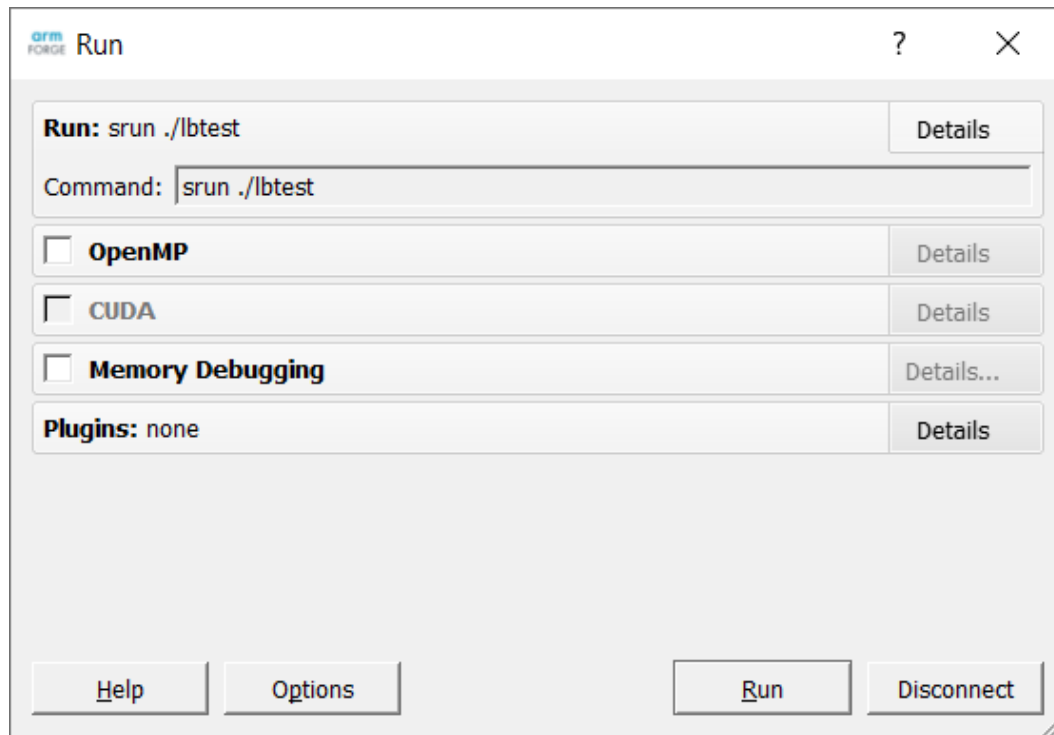
---

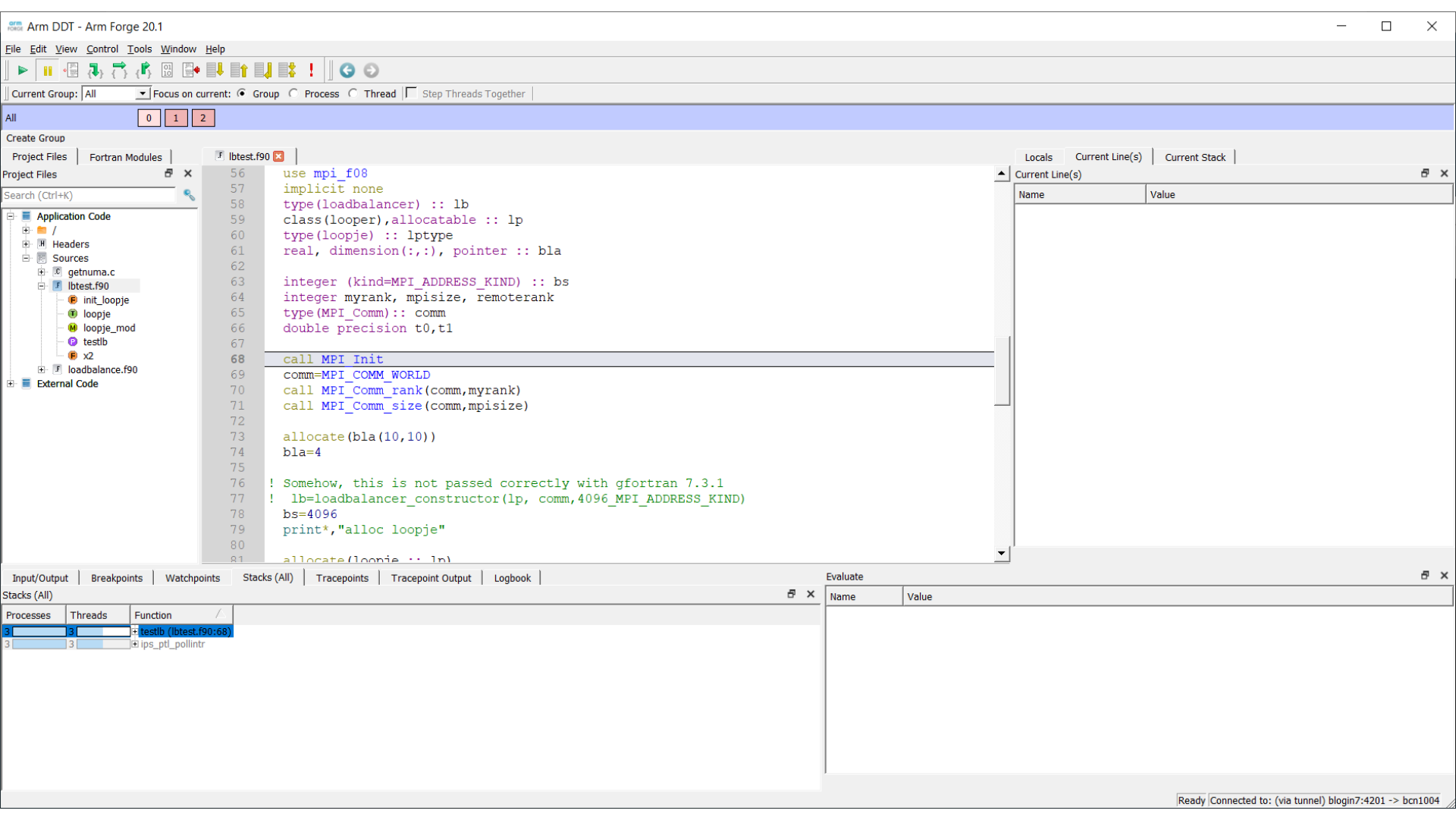
- ▶ Submit the job
  - Use the test queues, if possible: maximum 8 nodes, 1 hour walltime
- ▶ Remote client shows a popup to accept a Reverse Connect Request:



# Starting to debug an application

## Enable optional features







4

Interactive debugging  
session

5

More information

# Getting started, User guide, Tutorials, Video library

---

<https://developer.arm.com/tools-and-software/server-and-hpc/debug-and-profile/arm-forge/arm-ddt>

E.g.:

„Visualizing data with DDT“

„Memory debugging“

„Debugging CUDA and OpenACC“

# Thank you for your attention!

[john.donners@atos.net](mailto:john.donners@atos.net)

# Questions?

Atos, the Atos logo, Atos Syntel, Unify, and Worldline are registered trademarks of the Atos group. December 2019. © 2019 Atos. Confidential information owned by Atos, to be used by the recipient only. This document, or any part of it, may not be reproduced, copied, circulated and/or distributed nor quoted without prior written approval from Atos.

The Atos logo is displayed in white, bold, sans-serif capital letters. The letter 'o' is stylized with a white dot in the center, resembling a target or a bullseye. The logo is positioned in the bottom right corner of the slide.

4

Interactive debugging  
session

- 
- ▶ Explain screen:
    - main source,
    - buttons,
    - files,
    - stack/line vars,
    - lower left: tabs for parallel stack trace, breakpoints, output, ..
    - Lower right: evaluate, useful for long expressions
  - ▶ Step
  - ▶ Set myrank to 99
    - This can be useful to check what happens for a different value.
  - ▶ Step
  - ▶ Explain sparkline for variable myrank.

- 
- ▶ Step into loadbalancer constructor
  - ▶ Run until `numa_shared_`,
  - ▶ Only select 1 process and „step into“, show C code
  - ▶ Show stacks, with different paths for processes
    - This is very useful if your application hangs. It quickly shows processes in a different code path
  - ▶ Analyse arrays
  - ▶ Evaluate expressions
  - ▶ Set breakpoints
  - ▶ View the MPI message queue