

OpenFOAM su server Linux, lavoro quotidiano da Windows

Guida passo passo a FileZilla, Notepad++, PowerShell, log di calcolo e ParaView client/server.

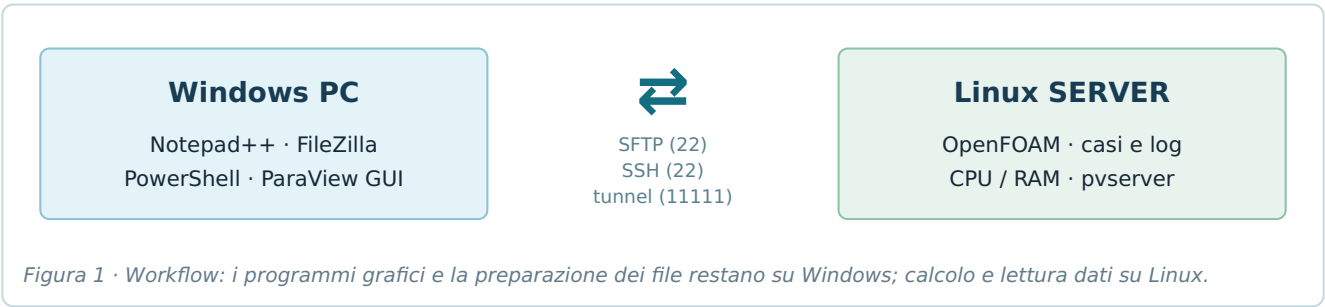
Windows 10/11 · Linux SSH/SFTP · OpenFOAM 14 · ParaView 5.13.2

Guida operativa indipendente dal server e dall'utente

Edizione: 16 ottobre 2026 · Esempi: OpenFOAM Foundation 14, ParaView 5.13.2, Windows 10/11, Linux con SSH/SFTP.

Obiettivo. Preparare e modificare i casi su Windows, trasferirli con FileZilla, eseguire soltanto i comandi necessari da PowerShell, scaricare e studiare i log su Windows e visualizzare i risultati con ParaView Windows collegato al pvserver Linux. Non occorre installare OpenFOAM su Windows per questo flusso.

1. Architettura: che cosa si fa e dove



Operazione	Windows	Server Linux
Modifica dei dizionari <code>system/</code> , <code>constant/</code> , <code>0/</code>	Notepad++ o VS Code	Conserva la copia eseguita
Trasferimento casi, script e log	FileZilla, protocollo SFTP	Servizio SSH, cartelle dei casi
Avvio mesh, controlli e simulazione	PowerShell → SSH	OpenFOAM, CPU/RAM e disco

Operazione	Windows	Server Linux
Monitoraggio del calcolo	PowerShell per comandi, FileZilla per copia log	<code>tail</code> , <code>grep</code> , log in scrittura
Analisi dei log	Notepad++, klogg , foglio di calcolo o Python	Facoltative estrazioni testuali
Visualizzazione CFD	ParaView GUI	<code>pvserver</code> , lettura dei risultati

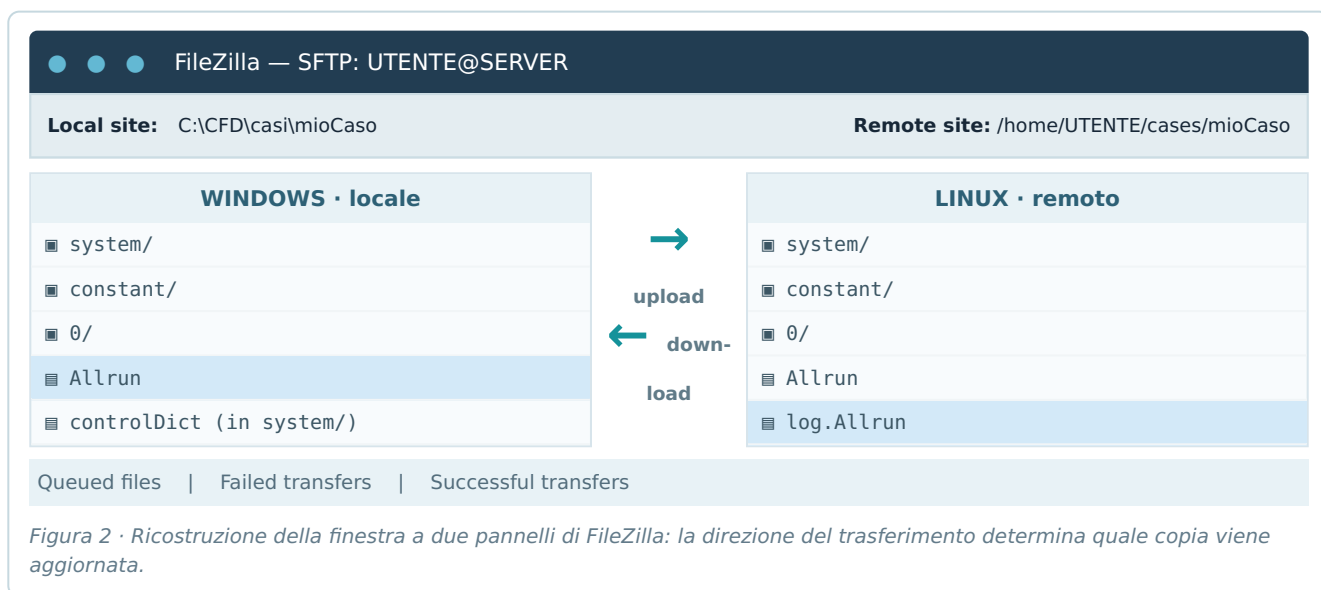
Tre nomi da sostituire negli esempi: `UTENTE` è l'account Linux, `SERVER` è il nome DNS/IP raggiungibile e `mioCaso` è la cartella del caso. Tutto ciò che segue presuppone l'accesso SSH autorizzato, normalmente sulla porta 22. Per una porta SSH diversa si usano `ssh -p PORTA` e la porta corrispondente in FileZilla.

2. Preparazione una tantum del PC Windows

1. Installare **FileZilla Client**, non FileZilla Server, da <https://filezilla-project.org/>.
2. Installare **Notepad++** da <https://notepad-plus-plus.org/>. In alternativa VS Code per navigazione di più file.
3. Installare **ParaView per Windows**, preferibilmente della **stessa versione** del `pvserver` Linux (esempio 5.13.2): <https://www.paraview.org/download/>. Evitare di mescolare client e server di versioni diverse.
4. Verificare il client OpenSSH: aprire PowerShell e digitare `ssh -V`. Se assente, attivare la funzionalità facoltativa **Client OpenSSH** di Windows. Non serve installare **Server OpenSSH** sul PC.
5. Facoltativo ma utile: **klogg**, visualizzatore di grandi file di log, da <https://github.com/variar/klogg>.

Crea, per esempio, `C:\CFD\casi\mioCaso` e `C:\CFD\log\mioCaso`. La prima è la copia di lavoro dei file di configurazione; la seconda serve a conservare le copie dei log scaricati. Scegli un solo punto di origine delle modifiche per evitare copie divergenti.

3. FileZilla: collegarsi in SFTP e trasferire i file



1. FileZilla → **File** → **Site Manager** (Gestore siti) → **New site**.
2. **Protocol:** `SFTP - SSH File Transfer Protocol` (**non** FTP in chiaro).

3. **Host:** SERVER . **Port:** 22 salvo diversa configurazione.
4. **Logon Type:** chiedere la password (**Interactive / Ask for password**, secondo versione) oppure **Key file** se è stata configurata un'autenticazione a chiave. **User:** UTENTE .
5. Premere **Connect**. Alla prima connessione confrontare l'impronta digitale della chiave host SSH con quella fornita dall'amministratore; non accettare automaticamente un'impronta sconosciuta.
6. Nel pannello **sinistro** aprire C:\CFD\casi\mioCaso , nel **destro** raggiungere /home/UTENTE/cases/mioCaso (o il percorso effettivo della propria area di lavoro).
7. Trascinare **da sinistra a destra** solo i file da pubblicare sul server; **da destra a sinistra** i log da analizzare. Controllare le schede **Failed transfers** e **Successful transfers**.

Attenzione alla direzione: il caricamento sovrascrive un file del server e il download può sovrascrivere la copia locale. Fare una copia dei dizionari prima di cambiamenti sostanziali; quando FileZilla segnala conflitti, verificare data e dimensione prima di sostituire. Un file che è ancora in scrittura può essere scaricato come fotografia temporanea, non come log definitivo.

Organizzazione consigliata del caso:

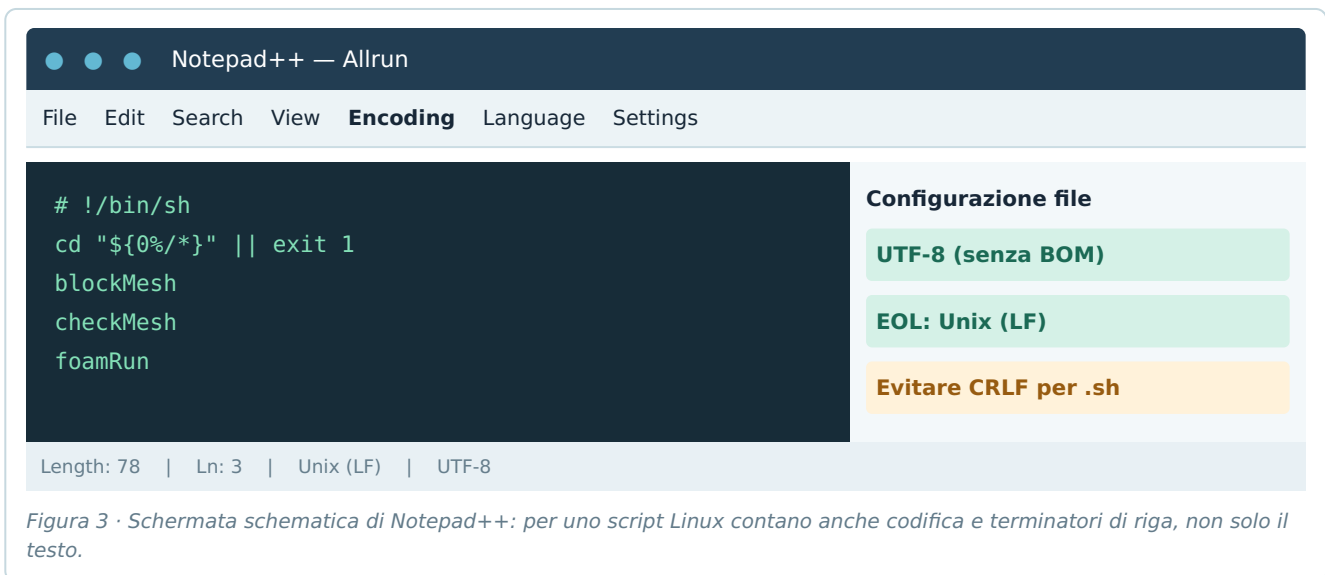
```

mioCaso/
├── 0/                # condizioni iniziali e al contorno
├── constant/         # proprietà e mesh
├── system/           # controlDict, fvSchemes, fvSolution...
├── Allrun             # eventuale script di esecuzione
└── log.*             # prodotti da utility e solver

```

Non trasferire automaticamente grandi directory dei tempi (0.1/ , 0.2/ , ...), processor*/ o l'intero post-Processing/ se l'obiettivo è lavorare in client-server: ParaView leggerà i dati direttamente sul server. FileZilla può impostare cartelle locali e remote predefinite nelle opzioni avanzate del sito.

4. Notepad++: scrivere dict e script Linux senza problemi



I file OpenFOAM sono generalmente **testo semplice**, anche se non hanno estensione. Notepad++ è pratico per controlDict, fvSchemes, fvSolution, blockMeshDict, file di campo in 0/ e script Allrun/All-clean.

- Impostare **Encoding** → **UTF-8** senza BOM; evitare UTF-16 e UTF-8 con BOM per script e file di configurazione.

- Impostare **Edit → EOL Conversion → Unix (LF)**, soprattutto per `Allrun` e script `.sh`. In **Settings → Preferences → New Document**, scegliere in anticipo formato Unix e codifica UTF-8.
- Usare **Language → C/C++** per una colorazione indicativa dei dizionari, oppure Shell per gli script; non è una validazione del formato OpenFOAM.
- Conservare **maiuscole/minuscole** dei nomi: Linux distingue `controlDict` da `controldict`. Non aggiungere `.txt` automaticamente a file come `fvSolution`.
- Non modificare i dizionari sul server mentre una simulazione li usa, se non si conoscono le conseguenze: preparare una revisione, trasferirla e **poi** rilanciare in modo controllato.

Esempio di modifica in `system/controlDict`:

```
application      foamRun;
startFrom        startTime;
startTime        0;
endTime          100;
deltaT           1;
writeControl      timeStep;
writeInterval     10;
```

Il contenuto è **solo illustrativo**: non sostituire in blocco il `controlDict` di un caso reale. In OpenFOAM 14 l'eseguibile `foamRun` usa un modulo di solver scelto in base alla configurazione del caso: per questo occorre leggere i dict esistenti.

Errore classico: `./Allrun: /bin/sh^M: bad interpreter`. Significa quasi sempre che lo script è stato salvato con terminazioni Windows CRLF; convertirlo in LF su Windows e ricaricarlo. In caso di dubbio verificare con `file Allrun` sul server.

5. PowerShell: entrare su Linux e orientarsi

PowerShell funge da **terminale di accesso**. Dopo il comando `ssh`, il prompt appartiene alla shell Linux del server: i comandi successivi sono Linux, **non PowerShell**.

```
# Da Windows PowerShell:
ssh UTENTE@SERVER
# Per SSH su porta diversa dalla 22:
ssh -p 2222 UTENTE@SERVER
```

```
# Una volta dentro Linux:
whoami          # account corrente
pwd             # directory corrente
ls -lh          # elenco leggibile, dimensioni
ls -la          # include file nascosti
cd ~/cases      # cartella dei casi
cd mioCaso      # entra nel caso
cd ..           # torna alla directory superiore
cd ~            # torna alla propria home
mkdir -p ~/cases/mioCaso # crea una cartella se assente
find . -maxdepth 2 -type f # elenco file fino a 2 livelli
exit            # chiude SSH, torna a PowerShell
```

Se `foamVersion` non viene trovato, la shell potrebbe non avere caricato l'ambiente OpenFOAM. Per un'installazione Foundation in `/opt/OpenFOAM-14`:

```
source /opt/OpenFOAM-14/etc/bashrc
foamVersion
command -v blockMesh
command -v foamRun
```

Il percorso è un **esempio**, non una garanzia: dipende dall'installazione. Alcuni server configurano già l'ambiente al login. Per le verifiche essenziali, usare `command -v` e `--help` dei programmi disponibili.

6. Avviare un caso senza imparare tutta la shell

Prima esecuzione e controllo

```
cd ~/cases/mioCaso
ls -l Allrun system/controlDict
sed -n '1,100p' Allrun          # leggere PRIMA lo script
blockMesh 2>&1 | tee log.blockMesh
checkMesh 2>&1 | tee log.checkMesh
```

Eseguire `blockMesh` **solo se** il caso usa `blockMeshDict` e la mesh va generata con questo strumento. I comandi del file `Allrun` potrebbero già generare la mesh: evitare duplicazioni; seguire l'ordine definito dal caso. `checkMesh` esamina la mesh esistente.

Avvio dello script batch `Allrun`

```
cd ~/cases/mioCaso
chmod u+x Allrun          # se manca il permesso di esecuzione
./Allrun 2>&1 | tee log.Allrun # mostra a video e salva su file
```

Non dare per scontato che ogni caso abbia `Allrun`: lo script è una convenzione, non una caratteristica obbligatoria. Alcuni casi prevedono un diverso script o l'avvio diretto di `foamRun`.

Non perdere il calcolo se la rete cade: `tmux` (consigliato)

```
tmux new -s cfd_mioCaso
cd ~/cases/mioCaso
./Allrun 2>&1 | tee log.Allrun
```

Per **lasciare il lavoro attivo**: premere `Ctrl+B`, rilasciare, poi `D` (*detach*). Per rientrare in un'altra sessione SSH:

```
tmux ls
tmux attach -t cfd_mioCaso
```

`Ctrl+C` **interrompe** il comando attivo: non va usato per scollegarsi. `exit` dalla shell ordinaria può chiudere la sessione: prima fare il detach di `tmux`. Se lo script non esiste, avviare l'eseguibile appropriato (`foamRun` in molti casi Foundation 14) soltanto dopo aver controllato la configurazione.

Opzione senza `tmux`, meno interattiva: `nohup ./Allrun > log.Allrun 2>&1 < /dev/null &`; si usa solo se `tmux` non è disponibile e se lo script è compatibile con un avvio non interattivo.

7. Controllare esecuzione e log dal terminale

```
cd ~/cases/mioCaso
ls -lh log*
tail -n 40 log.Allrun          # ultime 40 righe
tail -f log.Allrun             # segue le nuove righe (Ctrl+C esce da tail)
grep -n -E 'FOAM FATAL|Error|error|Courant|Solving for|Time =' log.Allrun
ps -u "$USER" -o pid,etime,cmd # processi dell'utente
```

L'elenco `ps` conferma che un processo esiste, **non** che la soluzione stia convergendo. Anche un messaggio `End` non sostituisce il controllo dei risultati. Su molti casi i log dei singoli comandi sono nominati `log.blockMesh`, `log.checkMesh` o `log.foamRun`: consultare ciò che è stato realmente creato.

Un log OpenFOAM tipico contiene progressione temporale (`Time =`), residuali dei campi (`Solving for ... Initial residual ... Final residual ...`), `Courant Number`, tempo di esecuzione e segnalazioni di errore. I valori e i criteri di accettazione dipendono dal modello: **residuali piccoli non dimostrano da soli** la correttezza fisica.

8. Scaricare i log con FileZilla e analizzarli su Windows

1. Da FileZilla aprire a destra la cartella **remota** del caso e a sinistra `C:\CFD\log\mioCaso`.
2. Individuare `log.Allrun`, `log.foamRun`, `log.checkMesh` o altri `log.*`; selezionare i file di interesse.
3. Trascinare **da destra verso sinistra** o usare **Download**. Verificare la coda di trasferimento.
4. Rinominare **solo la copia Windows**, se serve storicizzarla: `log.foamRun_2026-10-16_provaA.txt`.
5. Aprire file piccoli/medi in Notepad++; per log molto lunghi usare **klogg**: ricerca rapida, espressioni regolari, evidenziazione e navigazione dei risultati.

Ricerca rapida in klogg o Notepad++: `FOAM FATAL`, `Floating point exception`, `Courant Number`, `Time =`, `Initial residual`, `Final residual`, `ExecutionTime`, `End`.

Esempio di espressione regolare per cercare righe risolutive: `Solving for|Initial residual|FOAM FATAL|Courant Number`.

Per **grafici** dei residuali, esportare una tabella a due colonne (iterazione/tempo e residuale) tramite un piccolo script Python, oppure utilizzare i dati già prodotti dalle function object di OpenFOAM se disponibili. Non aprire un log grezzo direttamente in Excel aspettandosi colonne omogenee: è prima necessario analizzarne il formato.

Regola pratica: FileZilla non fornisce un vero `tail -f` remoto. Per l'andamento in tempo reale usare PowerShell+SSH; per analisi approfondita scaricare una **copia** del log. I download ripetuti di un file in crescita possono dare versioni parziali e richiedono di aggiornare manualmente la copia.

9. ParaView in modalità client/server: principio di funzionamento



La GUI **ParaView su Windows** controlla un processo **pvserver su Linux**. Il `pvserver` apre mesh e risultati nel filesystem Linux, esegue lettura e filtri e comunica il necessario al client; non serve scaricare sul PC tutti i campi. I menu **File → Open** dopo la connessione navigano quindi **sul server**, non sul disco `C:`. La documentazione ParaView avverte che `pvserver` non autentica i client e il protocollo non cifra da solo il traffico: usare un **tunnel SSH** invece di esporre la porta 11111.

Metodo raccomandato: un terminale SSH che porta con sé il tunnel

Passo A. Da PowerShell Windows, con porta 11111 libera sul PC:

```
ssh -L 11111:127.0.0.1:11111 UTENTE@SERVER
```

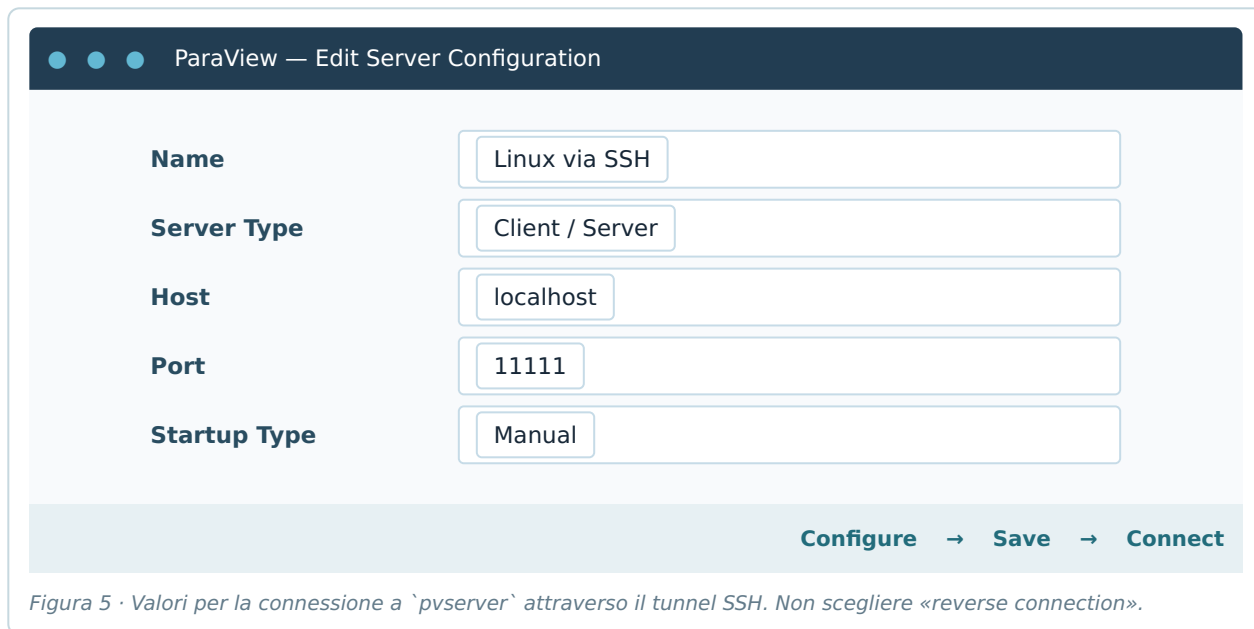
Questo comando apre una shell SSH e simultaneamente inoltra il `localhost:11111` di Windows verso il `127.0.0.1:11111` del server. **Tenere aperta** la finestra PowerShell per tutta la visualizzazione.

Passo B. Nella shell Linux appena aperta:

```
command -v pvserver
pvserver --server-port=11111 --bind-address=127.0.0.1
```

`--bind-address` è disponibile in ParaView 5.13.x e limita la porta al loopback del server. Se l'eseguibile non è nel `PATH`, trovare il percorso installato (per esempio `/opt/paraview-5.13.2/bin/pvserver`) e sostituirlo. Attendere un messaggio simile a `Waiting for client...`.

Passo C. Su Windows avviare ParaView GUI → File → Connect → Add Server:



Campo	Valore
Name	Linux via SSH
Server Type	Client / Server non reverse
Host	localhost
Port	11111
Startup Type	Manual

Salvare e premere **Connect**. Se la connessione funziona, il Pipeline Browser mostra una connessione di tipo `cs://localhost:11111` al posto del solo server locale `builtin`.

Aprire un caso OpenFOAM dal server

Nella shell Linux, nella directory del caso:

```
cd ~/cases/mioCaso
touch mioCaso.foam
```

`touch` crea un **file segnaposto vuoto**, non una copia dei risultati. Con ParaView collegato al server, scegliere **File → Open**, raggiungere la cartella Linux `/home/UTENTE/cases/mioCaso`, selezionare `mioCaso.foam`, usare il lettore **OpenFOAM** e premere **Apply**. Quindi abilitare campi, regioni o patch e scegliere il time-step desiderato. La disponibilità del reader dipende dai componenti installati in `pvserver`.

Se la simulazione è stata scomposta in `processor0/`, `processor1/` ecc., verificare nel lettore la modalità **Decomposed Case**, se supportata; in alternativa ricostruire con `reconstructPar` quando il caso lo permette. Non eseguire una ricostruzione onerosa solo per aprire il caso, senza prima verificare le opzioni del lettore.

Terminare la sessione

Da ParaView scegliere **Disconnect**; quindi nel terminale del server terminare `pvserver` con `Ctrl+C` se è ancora attivo e uscire da SSH con `exit`. La chiusura del tunnel rende impossibile la connessione GUI finché non viene riaperto.

Se non appare nulla o ci sono errori OpenGL: non è un problema da risolvere installando OpenFOAM su Windows. Verificare che `pvserver` abbia un backend grafico adeguato (X11, EGL o rendering software, secondo build e server), controllare gli errori nel terminale e chiedere all'amministratore quale modalità di rendering è configurata. Una sessione Linux priva di display non implica automaticamente che ogni build di `pvserver` possa renderizzare senza configurazione.

10. Mini-esercitazione completa: da Windows al risultato

1. In Notepad++ apri `C:\CFD\casi\mioCaso\system\controlDict`; fai una modifica concordata e salva in **UTF-8 senza BOM, LF**.
2. In FileZilla entra via SFTP, confronta i file e carica **solo** il `controlDict` modificato nella corretta cartella remota `system/`.
3. Da PowerShell: `ssh UTENTE@SERVER`, poi `cd ~/cases/mioCaso` e verifica con `pwd` e `ls system/controlDict`.
4. Leggi `Allrun` prima di avviarlo. Crea una sessione `tmux`, lancia la procedura e salva `log.Allrun`.
5. Controlla il log con `tail -f` (da un'altra sessione SSH) e verifica l'assenza di errori critici.
6. Con FileZilla scarica il log in `C:\CFD\log\mioCaso`; aprilo in klogg, cerca messaggi di errore e valori residui.
7. A calcolo terminato, usa una **nuova** sessione SSH con tunnel, avvia `pvserver`, collega ParaView Windows e apri il `.foam` direttamente dal filesystem Linux.
8. Salva sul PC immagini, tabelle e note di analisi. I risultati CFD restano sul server, salvo i file deliberatamente esportati.

11. Problemi frequenti e diagnosi

Sintomo	Verifica / correzione
<code>ssh: command not found</code> su Windows	Attivare il client OpenSSH di Windows; riaprire PowerShell.
<code>Permission denied</code> SSH/SFTP	Controllare utente, password/chiave, autorizzazioni e impronta host.

Sintomo	Verifica / correzione
FileZilla mostra directory sbagliata	Controllare la cartella remota assoluta e i permessi; non assumere che la root / sia la home.
foamRun / blockMesh: command not found	Caricare l'ambiente OpenFOAM o verificare installazione e PATH.
./Allrun: Permission denied	Controllare <code>ls -l Allrun</code> e, se appropriato, <code>chmod u+x Allrun</code> .
bad interpreter: /bin/sh^M	Convertire terminazioni CRLF in LF, ricaricare lo script.
SSH si chiude, simulazione si interrompe	Avviare in <code>tmux</code> e fare detach prima di chiudere.
ParaView non si connette	<code>pvsolver</code> attivo? tunnel aperto? stessa porta 11111? client/server compatibili?
Address already in use	Porta 11111 già occupata su Windows o server: liberarla o usare coppia di porte coerente.
Il file .foam non appare	<code>touch mioCaso.foam</code> nella directory giusta; controllare se si naviga lato server.
Campi mancanti / lettura fallita	Verificare formato del caso, mesh, decomposizione e reader disponibile in <code>pv-server</code> .
Schermata nera / errore OpenGL	Controllare le capacità di rendering e i messaggi di <code>pvsolver</code> ; non usare tentativi casuali su X11/GPU.

12. Scheda rapida stampabile

```

WINDOWS / POWERSHELL
ssh UTENTE@SERVER
ssh -L 11111:127.0.0.1:11111 UTENTE@SERVER # sessione tunnel ParaView

LINUX / CASE
pwd ; ls -lh ; cd ~/cases/mioCaso
foamVersion
sed -n '1,100p' Allrun
blockMesh ; checkMesh # SOLO se richiesto dal caso
tmux new -s cfd_mioCaso
./Allrun 2>&1 | tee log.Allrun
# Detach: Ctrl+B, poi D
# Riprendi: tmux attach -t cfd_mioCaso
tail -n 40 log.Allrun
tail -f log.Allrun

LINUX / PARAVIEW (NELLA SESSIONE SSH CON TUNNEL)
pvsolver --server-port=11111 --bind-address=127.0.0.1
cd ~/cases/mioCaso ; touch mioCaso.foam

WINDOWS / PARAVIEW
File > Connect > Add Server > Client/Server
Host localhost ; Port 11111 ; Startup Manual
File > Open > /home/UTENTE/cases/mioCaso/mioCaso.foam

```

13. Fonti ufficiali e approfondimenti

- OpenFOAM Foundation, **OpenFOAM 14** e guida introduttiva: <https://openfoam.org/version/14/> e <https://openfoam.org/download/14-ubuntu/> .
- ParaView, **Remote and parallel visualization**, manuale **5.13.2**: <https://docs.paraview.org/en/v5.13.2/ReferenceManual/parallelDataVisualization.html> .
- ParaView, **Loading Data**, manuale **5.13.2**: <https://docs.paraview.org/en/v5.13.2/UsersGuide/dataIngestion.html> .
- ParaView, note di rilascio 5.13 (`--bind-address`): https://www.paraview.org/paraview-docs/v5.13.2/cxx/md__builds_gitlab-kitware-sciviz-ci_Documentation_release_ParaView-5_813_80.html .
- FileZilla Wiki, **Site Manager** e autenticazione SFTP: https://wiki.filezilla-project.org/Site_Manager .
- Notepad++ User Manual, **New Document / encoding / line endings**: <https://npp-user-manual.org/docs/preferences/> .
- Microsoft Learn, **OpenSSH Windows**: https://learn.microsoft.com/it-it/windows-server/administration/openssh/openssh_install_firstuse .
- klogg, **log viewer** e documentazione: <https://github.com/variar/klogg> .

Schermate originali di riferimento (consultabili online): FileZilla SFTP, menu Unix LF in Notepad++, configurazione server ParaView (guida Kitware). Le immagini mostrano versioni/interfacce che possono differire da quelle installate.

Nota sulle figure: le schermate in questa guida sono **ricostruzioni schematiche** dei controlli principali, non fotografie dello specifico PC. Le interfacce possono cambiare con la versione o la lingua del programma; i riferimenti ufficiali sopra includono anche schermate originali dei rispettivi programmi.