**ADIOS2**

ADIOS2 (Adaptable Input Output System) este o biblioteca open source dezvoltata pentru a facilita gestionarea eficienta a datelor in aplicatii stiintifice complexe, care ruleaza pe supercomputere sau pe platforme de calcul in paralel. In esenta, ADIOS2 permite utilizatorilor sa stocheze si sa acceseze date intr-un mod rapid si eficient, esential pentru simularile de mari dimensiuni sau procesele de calcul care genereaza cantitati mari de informatie. Acest sistem a fost creat pentru a reduce complexitatea gestionarii datelor si pentru a imbunatati performanta aplicatiilor care utilizeaza aceste date.

De asemenea, ADIOS2 este proiectat sa functioneze pe platforme de calcul distribuit, in care mai multe calculatoare lucreaza simultan pentru a rezolva o problema comuna. In astfel de medii, ADIOS2 permite mai multor procese de calcul sa scrie si sa citeasca datele in paralel, fara a interfera unele cu altele. Acesta poate gestiona fluxuri de date provenite de la mai multe noduri de calcul, permitand astfel o scalabilitate ridicata. Prin utilizarea tehnologiilor precum MPI (Message Passing Interface), ADIOS2 se asigura ca aplicatiile pot lucra in mod eficient chiar si pe supercomputere cu mii de noduri.

Un alt avantaj important al ADIOS2 este optimizarea procesului de citire si scriere a datelor. In loc ca datele sa fie salvate imediat ce sunt generate, ADIOS2 le poate colecta intr-un buffer temporar, ceea ce permite scrierea lor intr-un singur pas, reducand semnificativ costurile de I/O. Mai mult decat atat, ADIOS2 utilizeaza tehnici de comprimare a datelor, care nu doar ca reduc spatiul necesar pentru stocare, dar si imbunatatesc performanta transferului de date intre diferite sisteme.

ADIOS2 este, de asemenea, usor de integrat in aplicatiile existente. Acesta ofera un API simplu si bine definit, care poate fi utilizat in limbaje de programare populare, cum ar fi C++. Astfel, cercetatorii si dezvoltatorii nu trebuie sa invete noi tehnologii complexe pentru a utiliza ADIOS2, ci pot sa il integreze rapid in fluxurile lor de lucru.

ADIOS2 permite generarea fisierelor .bp printr-un proces eficient si flexibil, conceput pentru a stoca datele simulate sau experimentale intr-un format structurat si optimizat pentru citire paralela. Aceste fisiere pot include atat variabile, cat si atribute asociate, care descriu datele intr-un mod detaliat si clar. Pentru a face aceste fisiere compatibile cu ParaView, ADIOS2 permite utilizatorilor sa adauge un atribut specific cu cheia "vtk.xml" si o valoare care specifica tipul de date, cum ar fi "ImageData" sau "UnstructuredGrid", astfel incat ParaView sa poata interpreta corect structura acestora. Odata generate, aceste fisiere pot fi importate in ParaView, unde utilizatorii pot aplica filtre, animatii sau alte tehnici avansate de analiza pentru a explora datele stocate, facand astfel legatura intre simulare si vizualizarea rezultatelor.

Generarea datelor

Primul pas in lucrul cu biblioteca ADIOS2 este declararea clasei adios: adios2::ADIOS adios; Aceasta creeaza un obiect de tip ADIOS, care serveste drept punct central pentru toate operatiunile de intrare-iesire in cadrul aplicatiei. Prin acest obiect, utilizatorul poate configura modul in care datele vor fi scrise, citite sau gestionate.

Urmatorul pas este crearea unui obiect de tip IO: adios2::IO io = adios.DeclareIO("IO"); Acest obiect este un container logic care grupeaza toate variabilele si setarile asociate unei operatiuni specifice de intrare-iesire. Prin aceasta declaratie, utilizatorul poate configura parametrii necesari si poate organiza mai bine fluxul de date.

Dupa aceea se seteaza modul in care datele sunt transferate intre aplicatie si mediul de stocare: io.SetEngine("BP"); Pentru a genera date care pot fi vizualizate in ParaView este necesara utilizarea motorului BP (Binary Pack).

ADIOS2 permite stocarea atributelor in interiorul obiectului IO. Atributele sunt perechi de tipul (cheie, valoare), unde cheia este identificatorul unic al atributului, iar valoarea este informatia stocata de atribut. Spre exemplu, daca vrem sa stocam acceleratia gravitationala: io.DefineAttribute<float>("g", 9.81);

Pentru ca ParaView sa stie cum sa interpreteze datele, este necesara salvarea unui atribut cu cheia "vtk.xml" si o valoare care descrie tipul de structura si formatul datelor stocate. De exemplu:

io.DefineAttribute<std::string>("vtk.xml", R"(

<?xml version="1.0"?>

<VTKFile type="ImageData" version="0.1" byte\_order="LittleEndian">

<ImageData WholeExtent=")" + extent + R"(" Origin="0 0 0" Spacing="1 1 1">

<Piece Extent=")" + extent + R"(">

<CellData Scalars="U">

<DataArray Name="U" />

<DataArray Name="V" />

<DataArray Name="TIME">

step

</DataArray>

</CellData>

</Piece>

</ImageData>

</VTKFile>)"

);

O variabila este un obiect care contine o colectie de date, fie scalare, fie dimensionale. Pentru a defini o variabila scalara trebuie precizat doar numele variabilei: adios2::Variable<int> varStep = io.DefineVariable<int>("step");, dar pentru definirea unei variabile dimensionale este necesara si precizarea dimensiunilor, startului si countului. Spre exemplu, adios2::Variable<int> varScalars = io.DefineVariable<int>("scalars", {3, 4, 5}, {0, 0, 0}, {3, 4, 5}); defineste o variabila tridimensionala cu dimensiunile 3x4x5.

Pentru a scrie datale in fisierul .bp este necesara declararea unui writer: adios2::Engine writer = io.Open("output.bp", adios2::Mode::Write); Pe langa dimensiunile spatiale, simularile contin si o dimensiune temporala. Aceasta poate fi controlata utilizand functiile writer.BeginStep(); si writer.EndStep();, intre care sunt actualizate valorile variabilelor pentru pasul respectiv: writer.Put(scalars, vecScalars.data());

Comanda "bpls -lav output.bp" afiseaza o lista cu atributele si variabilele din fisier si tipurile lor de date. Fiecare atribut este insotit de valoarea lui, iar fiecare variabila este insotita atat de dimensiunile ei, cat si de valorea minima si cea maxima (ex. 10\*scalar = 0 / 9, 10\*{10, 10, 10} = 0 / 1).

Vizualizarea datelor

Pentru a vizualiza fisierul .bp este nevoie de o versiune a programului ParaView care ofera suport pentru ADIOS2. Dupa deschiderea fisierului se selecteaza readerul (ex. "ADIOS2VTXReader"), se da click pe "Apply" si se selecteaza campul de date care se doreste a fi vizualizat (ex. "scalars"), cat si modul de vizualizare (ex. "Surface"). Datele sunt prezentate sub forma unui color map (scalar sau logaritmic), cu un range customizabil. Meniul de informatii ii ofera utilizatorului un rezumat al datelor pentru pasul actual: numele fiecarui camp, insotit de tipul de date, cat si valoarea minima si cea maxima din campul respectiv. Utilizatorul poate selecta pasul pe care vrea sa il vizualizeze, sau poate sa vizualizeze toti pasii in succesiune, asemenea unei animatii.

In concluzie, ADIOS2 reprezinta o solutie eficienta si scalabila pentru gestionarea datelor in aplicatiile stiintifice de mari dimensiuni. Flexibilitatea, performanta ridicata si usurinta in utilizare il fac un instrument esential pentru cercetatori si dezvoltatori care lucreaza cu simulari de inalta performanta si care trebuie sa gestioneze volume mari de date. Prin integrarea tehnologiilor avansate de stocare si acces la date, ADIOS2 contribuie semnificativ la imbunatatirea performantei si eficientei simularilor stiintifice.

Bibliografie:

https://adios2.readthedocs.io

https://adios2.readthedocs.io/\_/downloads/en/v2.9.2/pdf/

https://adios2.readthedocs.io/en/v2.10.2/ecosystem/visualization.html

https://www.paraview.org/download/

https://docs.vtk.org

https://vtk.org

https://github.com/pnorbert/adiosvm

https://github.com/ornladios/ADIOS2