Middlewares para Sistemas de Alto Rendimiento

José M. Peña

Contenidos

- Middlewares:
- Ejemplo lenguajes/entornos de programación:
  - Lenguaje de programación paralela:
    - OpenMP
- Ejemplos de servicios HPC:
  - Sistemas de ficheros para cluster
    - Lustre
  - Memoria compartida distribuida
    - TreadMarks

Middleware

- **Middleware**: En el concepto de “software intermediario”, da soporte al desarrollo de aplicaciones distribuidas:
  - Abstracción: Haciendo de interfaz entre las tareas/procesos y el hardware o sistema operativo.
  - Servicios: Proporciona servicios y funcionalidades para el desarrollo de aplicaciones.

Middleware de un Sistema Distribuido

- Objetivos:
  - Interoperabilidad.
  - Facilidad de desarrollo.
- Ejemplos de Servicios:
  - Descubrimiento.
  - Seguridad.
- Mecanismos:
  - Recubrimiento del Software
  - Abstracciones.
  - Deslocalización de servicios.

Middleware de un Sistema Cluster

- Objetivos:
  - Rendimiento.
  - Disponibilidad.
- Ejemplos de Servicios:
  - Acceso a datos.
  - Checkpointing.
- Mecanismos:
  - Acceso directo al HW
  - Reparto de carga.
  - *Caching*.

Middleware

Niveles de Middleware

- Nivel de implementación:
  - Acceso a E/S compartida (e.g., disco).
  - Migración de procesos/*checkpointing*.
  - Espacio unificado (procesos, usuarios, …)
- Nivel de programación:
  - Lenguajes y bibliotecas específicos (HPF, MPI, **OpenMP**)
  - Servicios:
    - **Sistemas de ficheros de cluster**
    - **Memoria compartida distribuida (DSM)**
- Nivel de gestión:
  - Gestión de trabajos.
  - Administración del sistema.

MIDDLEWARE DE SISTEMAS DE ALTO RENDIMIENTO

Lenguajes y entornos de programación

- Lenguajes de programación:
  - Lenguajes paralelos

OpenMP

- Estandarizado.
- Adecuado para:
  - Máquinas SMP
  - Memoria compartida
  - Paralelismo a nivel de tarea/datos
- Características:
  - Basado en directrices de pre-procesador (en fase de compilado).
  - Fácil de utilizar.
  - Flexibilidad limitada.
  - Dependencia del compilador.

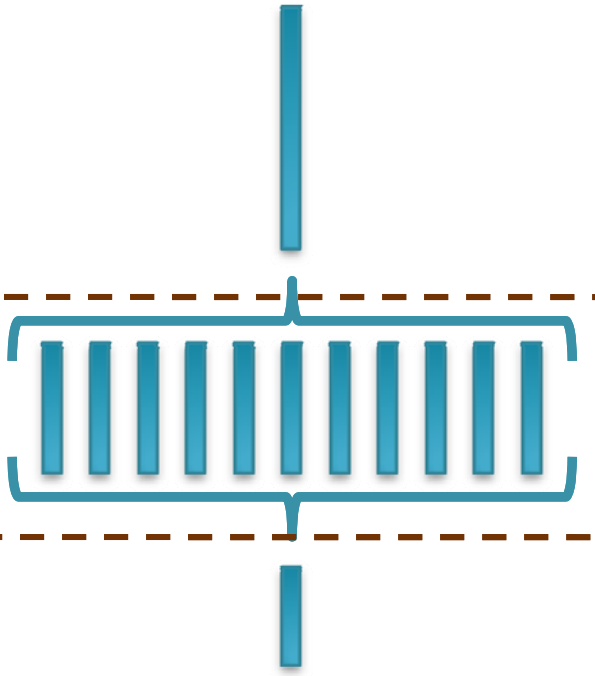
OpenMP

- Estructuras de control paralelas

```
int main(int argc, char **argv)
{
    const int MAX = 250000;
    int i, m[MAX];

    #pragma omp parallel for
    for (i = 0; i < MAX; i++)
        m[i] = i*i*1.23;

    return 0;
}
```



OpenMP

- Mecanismos de sincronización:
  - Barreras de sincronización:
    - **`#pragma omp barrier`**
  - Operaciones atómicas:
    - **`#pragma omp atomic`**
  - Secciones críticas:
    - **`#pragma omp critical`**

OpenMP

- Funciones auxiliares:
  - Barreras de sincronización:
    - **`omp_get_thread_num()`**
    - **`omp_get_num_threads()`**
  - Información de control:
    - **`omp_get_nested()`**
    - **`omp_in_parallel()`**
  - Control de cerrojos:
    - **`omp_*_lock()`**

MIDDLEWARE DE SISTEMAS DE ALTO RENDIMIENTO

Servicios a nivel de programación

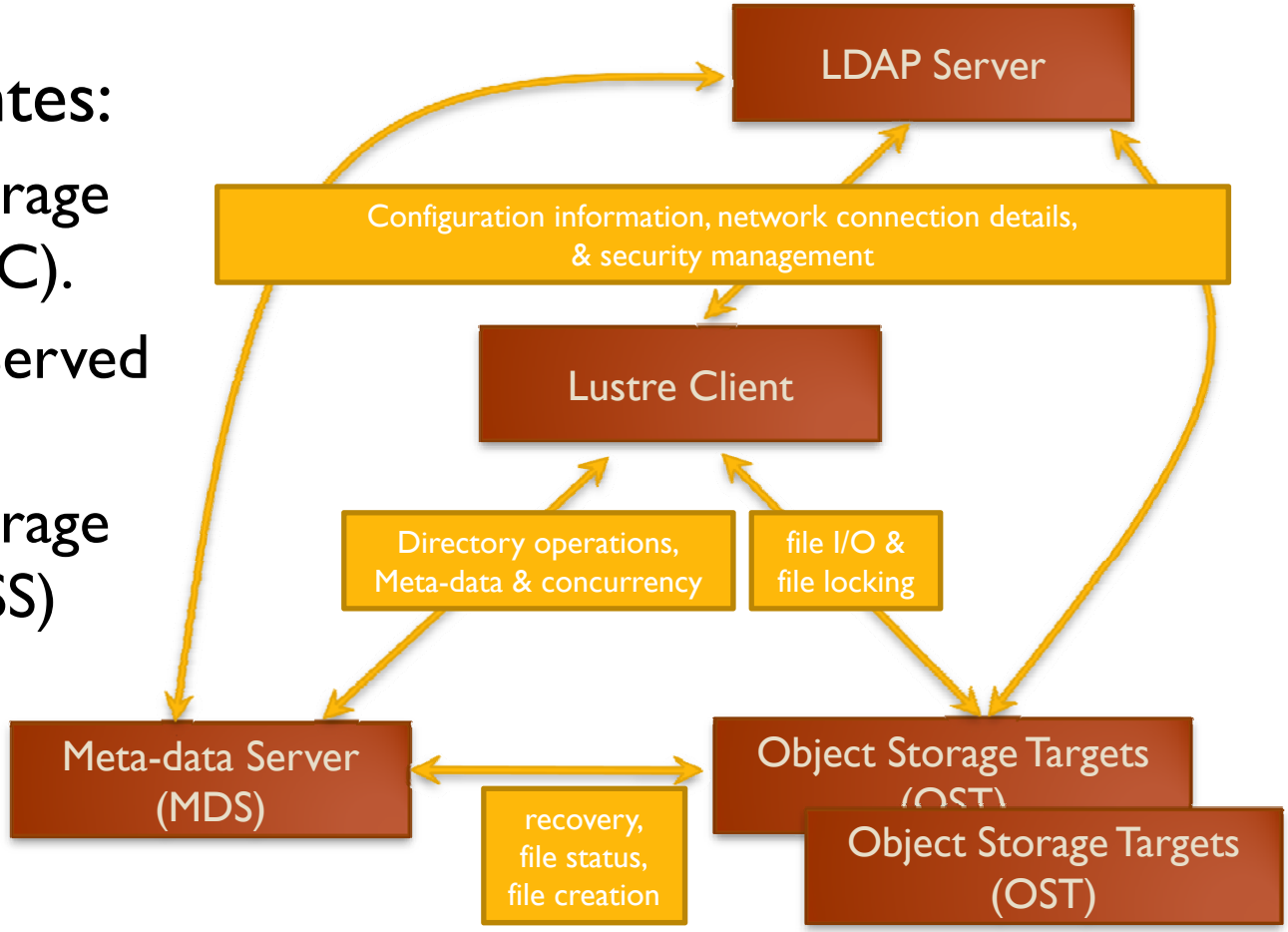
- Servicios:
  - Sistemas de ficheros de cluster.
  - Sistemas de memoria compartida distribuida.

Lustre

- Desarrollado por Sun Microsystems
  - Características generales:
    - Compatible POSIX
    - Con estado
    - Basado en objetos
    - Tolerante a fallos
  - Componentes:
    - Object Storage Client (OSC).
    - Metadata Served (MDS)
    - Object Storage Server (OSS)

Lustre

- Componentes:
  - Object Storage Client (OSC).
  - Metadata Served (MDS)
  - Object Storage Server (OSS)



Lustre

- Mejoras de rendimiento:
  - Soporte nativo sobre redes de altas prestaciones (e.g., InfiniBand), en lugar de pasar por la pila TCP/IP

DSM

- Motivación:
  - Migración de código de sistemas multiprocesadores/vectoriales.
  - Modelo de programación más intuitivo.
- Implementaciones:
  - Basadas en compiladores.
  - Basadas en memoria virtual:
    - TreadMarks

DSM: Gestión de Copias

- Número de copias:
  - Única: Mal rendimiento.
  - Múltiples de lectura.
  - Múltiples de lectura/escritura.

- Localización:
  - Broadcast.
  - Gestor de páginas DSM
  - Gestión distribuidas: Múltiples gestores.

DSM: TreadMarks

- Biblioteca de DSM a nivel de usuario:
  - Bibliotecas propias de gestión.
  - Identificación selectiva de variables compartidas (páginas de memoria virtual).
  - Indicación explicita de sincronización.
  - Consistencia perezosa.
  - Múltiples lectores/escritores.
  - Utiliza copias de trabajo y originales modificados.
  - Combinación en base a diferencias.

DSM: TreadMarks

- Uso:
  - Reserva de memoria (**`Tmk_malloc`**).
  - Compartición explícita de memoria (**`Tmk_distribute`**).
  - Sincronización entre todas las copias (**`Tmk_barrier`**, **`Tmk_lock_...`**).
  - Consistencia al adquirir cerrojos:
    - Invalidando datos compartidos.
    - Alternativas:
      - Recuperándolos bajo demanda (*invalidate*).
      - Mandando cambios al adquirir el cerrojo (*update*).