



SUPERCOMPUTING CONFERENCE

2008

A close-up photograph of water splashing, with white foam and blue water, set against a dark blue background with a yellow curved line on the right side.

Extending the OpenMP profiling API for OpenMP 3.0

Yuan Lin, Oleg Mazurov

Sun Microsystems, Inc.

An OpenMP Runtime API for Profiling

- Formal definition at
 - > http://www.compunity.org/futures/omp_collector_api.h
- White paper at
 - > <http://www.compunity.org/futures/omp-api.html>
- Last updated on Oct 31, 2007
- Requires another update to support
 - > tasks in OpenMP 3.0
 - > vendor specific extensions

Meta API

```
int __omp_collector_api( void *arg )
```

- Request layout

Size	Request #	Error code	Rsize	Mem
------	-----------	------------	-------	-----

- Multiple requests

Request 1	Request 2	...	Request n	0
-----------	-----------	-----	-----------	---

Meta API

- Dynamic binding (versionless)
- One call – many requests (consistency)
- Multi-value returns
- Variable size returns (strings, arrays, etc.)
- Full memory control on the consumer side
- Extensible

Profiling API

- Requests
 - > OMP_REQ_START
 - > OMP_REQ_REGISTER
 - > OMP_REQ_STATE
 - > ...
- Events
 - > OMP_EVENT_FORK
 - > OMP_EVENT_JOIN
 - > ...

New features

- Explicit request layouts

```
struct OMP_request_register {  
    int                size;    /* entry length */  
    OMP_COLLECTORAPI_REQUEST reqn; /* request number */  
    OMP_COLLECTORAPI_EC  errc;  /* error code */  
    int                rtsz;    /* return size */  
    OMP_COLLECTORAPI_EVENT event; /* event type */  
    void                *callback; /* callback address */  
};
```

New features (cont.)

- New requests
 - > OMP_REQ_TASK_IDN – get task ID at level N
 - > OMP_REQ_TASK_SRCN – get task source line info
 - > OMP_REQ_CREATED_TASK – get task ID of a newly created task
 - > ...
- New state
 - > THR_TSKWT_STATE – thread in #pragma omp taskwait

New features (cont.)

- New events
 - > OMP_EVENT_CREATE_TSK
 - > OMP_EVENT_BEGIN_TSK
 - > OMP_EVENT_SUSPEND_TSK
 - > OMP_EVENT_RESUME_TSK
 - > OMP_EVENT_END_TSK
 - > ...

New features (cont.)

- Extension mechanism

```
#define OMP_REQ_SUNEXTENSION \
    ((OMP_COLLECTORAPI_REQUEST)0x4A415641)
```

```
#define OMP_REQ_XREQ1 ((OMP_COLLECTORAPI_REQUEST)-1)
#define OMP_REQ_XREQ2 ((OMP_COLLECTORAPI_REQUEST)-2)
```

...

```
#define OMP_EVENT_XEVENT1 ((OMP_COLLECTORAPI_EVENT)-1)
#define OMP_EVENT_XEVENT2 ((OMP_COLLECTORAPI_EVENT)-2)
```

...

User Model

- User callstacks
 - > Implementation agnostic
- OpenMP thread tree
 - > Reflects parallel region nesting
- OpenMP task tree

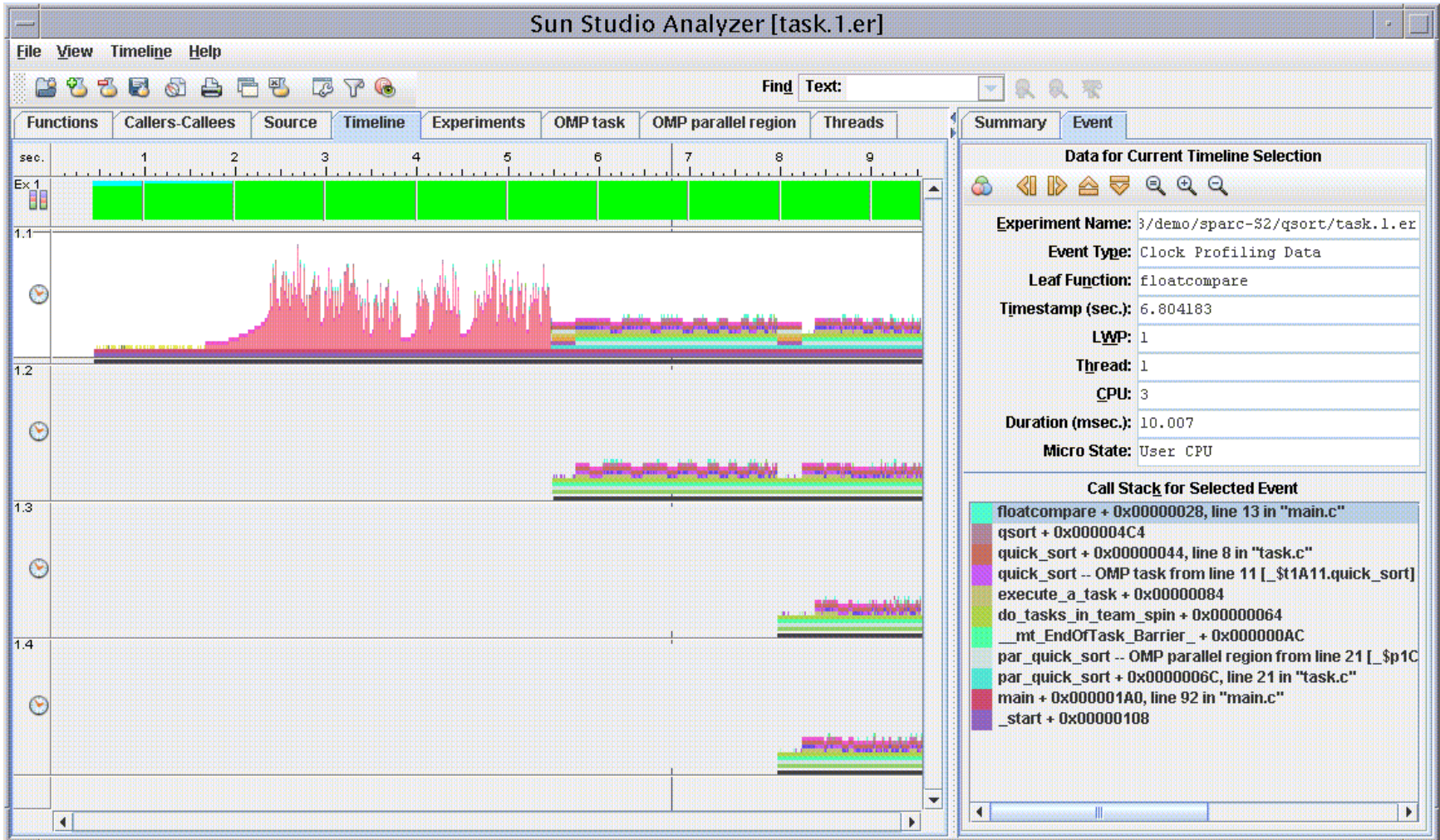
User Model (cont.)

- An arbitrary program event can be associated with
 - > OpenMP state
 - > User callstack
 - > A node in the thread tree with parallel region ID's and OpenMP thread ID's along the path
 - > A node in the task tree with task ID's along the path

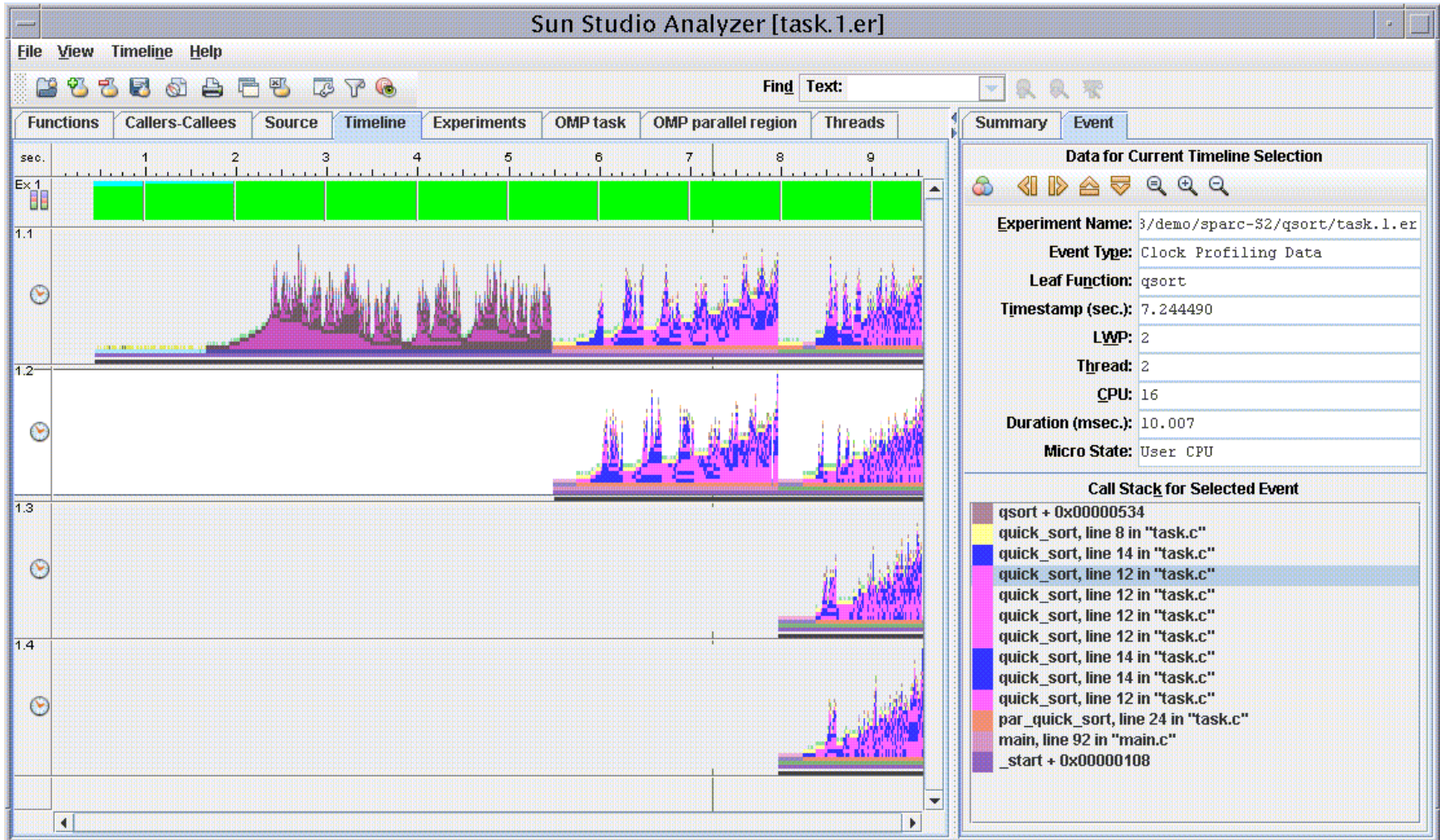
Example – quick sort

```
void quick_sort ( int lt, int rt, float *data ) {  
    if ( rt - lt < LOWLIMIT ) {  
        insertion_sort( lt, rt, data );  
    }  
    else {  
        int md = partition( lt, rt, data );  
        #pragma omp task  
        quick_sort( lt, md - 1, data );  
        #pragma omp task  
        quick_sort( md + 1, rt, data );  
    }  
}
```

Quick sort – machine view



Quick sort – user view



What's next?

- Demo at OpenMP booth and Sun booth (SunStudio station)
- A proposal to update the white paper
- Technical paper on user model implementation (tentative for IWOMP'2009)

THANK YOU

- **Oleg Mazurov**

oleg.mazurov@sun.com